8-13-2020 11:15 AM

Terramechanics and Machine Learning for the Characterization of Terrain

Bryan W. Southwell, *The University of Western Ontario*

Supervisor: McIsaac Kenneth, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Engineering Science degree in Electrical and Computer Engineering

Recommended Citation

Southwell, Bryan W., "Terramechanics and Machine Learning for the Characterization of Terrain" (2020). *Electronic Thesis and Dissertation Repository*. 7245.
https://ir.lib.uwo.ca/etd/7245

Abstract

An instrumented rover wheel can collect vast amounts of data about a planetary surface. Planetary surfaces are changed by complex geological processes which can be better understood with an abundance of surface data and the use of terramechanics. Identifying terrain parameters such as cohesion and angle of friction hold importance for both the rover driver and the planetary scientist. Knowledge of terrain characteristics can warn of unsafe terrain and flag potential interesting scientific sites. The instrumented wheel in this research uses a pressure pad to sense load and sinkage, a string potentiometer to measure slip, and records motor current draw. This thesis demonstrates the utilization of the instrumented wheel's data to estimate cohesion, angle of friction and grain size and demonstrates a machine learning solution for classifying terrain types with the same data. Mars simulants available at NASA-JPL were used for the collection of the data. Two machine learning classifiers were explored: Random Forest and Support Vector Machine. Binary and multi-class classification were both demonstrated and it is proposed that the classification model can identify terrain types based on the instrumented wheel data. The Random Forest model performed best in all classification types.

Keywords: Terramechanics, Planetary Science, Machine Learning, Rover Wheels, Random Forest, Support Vector Machine

Lay Summary

Terramechanics is the study of vehicle-terrain interaction. Terrain characteristics change depending on geologic processes such as erosion that may have occurred in a region. A change in terrain characteristics can be observed via terramechanics models which define the interaction of vehicles in the terrain. Through the collection of data on the interaction of a planetary rover with the surface terrain, terramechanics models can be used to estimate terramechanics parameters which define terrain characteristics. The identification of these parameters is important for both the rover driver and planetary scientist as they reveal if the terrain is unsafe for driving, previously unencountered and/or of scientific interest, and reveal the history of the planetary region where the rover finds itself. Cohesion, Angle of friction, and grain size are important parameters that can give insight into the safety for driving or history of the region. This thesis demonstrates the capability of a ”smart” sensing rover wheel to collect surface terrain data and the use of this data in estimating terrain parameters, and the identification of terrain types using machine learning methods. Two classification algorithms were explored: Random Forest and Support Vector Machine. It is proposed that both binary and multi-class classification models can differentiate between Mars simulants used to collect data for this study using pressure pad, wheel slip, and motor current data.

Acknowledgements

I would first like to thank my supervisor Dr. Kenneth McIsaac for his support throughout the research process and for encouraging me to explore all the areas of this research I felt had potential. I would like to thank Dr. Matt Cross for his abundantly helpful knowledge and guidance and for always making time to discuss my research direction. I want to thank my research group for their friendship, support and the many group outings and nacho nights which made this academic adventure all the better. I would also like to thank my supervisors and colleagues at NASA-JPL: Jack Lightholder, Yuliya Marchetti, Lukas Mandrake, Darwin Mick, Teresa McBryan, Paul Horton, Sawyer Myers, James Montgomery, Kevin Shannon and Alec Helbling for their friendship, advice, and for making every day of my JPL internship so wonderfully memorable. Finally, to my family, I thank them for their unending support and love which made stressful times so much easier and the good times even better.

Contents

List of Figures

List of Tables

List of Acronyms

Acronyms	Definition
SVM	Support Vector Machine
MMS	Mars Mojave Simulant
GRC	Glenn Research Center
WED	Wedron
MER	Mars Exploration Rover
JPL	Jet Propulsion Laboratory
NASA	National Aeronautics and Space Administration
RBF	Radial Basis Function
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron

Chapter 1

Introduction

1.1 Motivation

The broad motivation for this thesis work is to demonstrate that data collected by an instrumented rover wheel can be used to provide planetary scientists with information about terrain that they cannot otherwise obtain, and in addition, that classification of terrain using this data may have use in autonomous rover driving. An instrumented rover wheel passively collecting scientific data as the rover drives and other instruments perform science offers a unique avenue for furthering scientific understanding of planetary surfaces. In the context of the data available for this thesis research, the motivation is to present a method to use the wheel data from the *Barefoot Rover* project to estimate terramechanics terrain parameters, classify terrain using machine learning methods, and investigate a possible path to determine grain size. To the planetary scientist, terramechanics terrain parameters and grain size are valuable pieces of information for understanding past geologic processes in a specific region. For autonomous rover developments, the machine learning research will be most relevant. Terrain classifications can serve as a crucial piece of information for "go" or "no-go" decisions in an autonomous control system. The instrument data of interest is; the pressure pad data for estimating the load/contact area/sinkage of the wheel, the string potentiometer for slip ratio data, the motor current for estimation of the shear stress at wheel-terrain interface, which come together to allow the estimation of the terramechanics terrain parameters cohesion and angle of friction. Development of the *Barefoot Rover* project began at NASA-JPL as a way of enabling Mars rovers to incorporate in-situ terrain data into driving and scientific decisions. The project fuses terramechanics, robotics, and machine learning to prove the feasibility of an instrumented tacticle-sensing wheel as an instrument for further scientific research and for increased driving safety.

The estimation of terrain parameters, the classification of terrain and insight into grain size can aid in the following:

- The development of more accurate and representative Mars simulants for terrestrial use.
- The identification of unsafe terrain while a rover is driving on a celestial body.
- The identification of previously unseen terrain characteristics and scientifically interesting areas.
- The development of increasingly autonomous rover driving systems
- Ground truth against previous estimations and orbital data.

The intent of this work is to use regressor tools to estimate terramechanics parameters for MMS 2mm and GRC-01 Mars simulants, use a multi-class classifier to classify up to four Mars simulants, and investigate correlation between sinkage exponent - an existing terramechanics parameter - and the grain size of individual Mars simulants. This work uses data collected by the novel *Barefoot Rover* wheel and opens the door to the utilization of such technology for in-situ terrain study and enhanced rover safety.

The inspiration behind this research came from the development of the *Barefoot Rover* technology and how that technology could be used for terramechanics study and machine learning use cases. The classification of terrain allows for the back-inference of terramechanics parameters of a known simulant. This work is a demonstration of what can be achieved with such instrumented wheels and makes recommendations for deployment on the surface of celestial bodies like Mars. The *Barefoot Rover* wheel is unique compared to classical terrain measurement systems like the Bevameter for its ability to directly measure, as it rolls over terrain, wheel sinkage, wheel entrance and exit angles, and normal pressure exerted on the wheel, which in current state-of-the-art wheels must be visually or telemetry estimated with error. A Bevameter traditionally collects sinkage and pressure data, however, it is a stationary instrument and the vehicle must stop, take a measurement, move, take another measurement, and so on. The *Barefoot Rover* wheel can collect the same data while rolling and supporting a vehicle chassis, allowing a rover to perform other science with it's instrument suite. The Bevameter, in addition, is not practical for deployment on another planet due to it's size and heavy mass. The rover would also need to stop and measure repeatedly, with visual estimation of the sinkage, which does not fit into the context of supporting autonomous rover driving capabilities with continuous feedback about the vehicle-terrain interaction. Using the *Barefoot Rover* technology allows for passive and direct measurement of terramechanics characteristics as the wheel rolls.

R. Sullivan [1] and R. E. Arvidson [2, 3] carried out numerous studies using orbital data and data from the Viking landers and Mars Exploration Rovers (MER) to estimate the surface cohesion and angle of friction of certain areas of Mars. These studies showed reasonable estimates could be determined from this approach but they required significant computational power, required some visual imagery verification and could not be done in near real-time such as lightweight machine learning applications.

Ding [4] developed and verified terramechanics models for rover wheels equipped with grousers/lugs. He showed that it was possible to estimate terramechanics parameters reasonably well using least squares regression and data from a rover testbed. Sinkage and slip were implicitly measured using visual means.

The question then posed to be answered with this thesis work was, *“If individual terrains exhibit their own terramechanical characteristics, is it possible to classify them based on these characteristics, and is it possible to estimate cohesion, angle of friction, and grain size from raw instrumented rover/instrumented wheel data?”*

The demonstration of terrain classification and estimation of terramechanics parameters is especially useful for rover drivers and planetary scientists. Automated detection of unsafe, anomalous, or novel terrain can be used to make decisions on the safety of the terrain the rover is driving into or whether to stop and take further scientific analysis.

1.2 Tasks and Thesis Contribution

This research explores the use of instrumented rover wheel data to identify terrain types and their terramechanics characteristics. Specifically, the pressure pad, string potentiometer and motor current data can be used to develop a machine learning classification model of known Mars simulants that were driven upon when the data was collected. This classification can be done using engineered features and the Random Forest and Support Vector Machine machine learning algorithms. Additionally, this same data can be used to estimate terrain cohesion and angle of friction within known ground truth ranges, and be used to estimate sinkage exponent and a correlation between mean sinkage exponent values for various Mars simulants and their respective grain sizes.

The technology and data being used to conduct this research is a potentially new method of utilizing rovers, and specifically their wheels, to study a terrain's characteristics and to implement machine learning for terrain type detection.

1.3 Thesis Outline

The remainder of this thesis is organized as follows: Chapter 2 contains background about the instrumented wheel and terrain parameter estimations and a literature review of both terramechanics and machine learning theory relevant to this thesis. Chapter 3 discusses the investigation of the instrumented wheels ability to detect changes in terrain and describes the method and results for the estimation of cohesion and angle of friction from the instrumented wheel data. Chapter 4 details the methodology behind the machine learning classification of Mars simulants and presents the results of the classification. Chapter 5 discusses the estimation of terrain grain size using mean sinkage exponent values. Chapter 6 contains the discussion, conclusion and possible future work.

Chapter 2

Background and Literature Review

This chapter reviews background project information, classical terramechanics theory and its use in analyzing terrain parameters of Mars regolith, and background machine learning theory.

2.1 The Instrumented Wheel

The *Barefoot Rover* wheel is a solid-cut wheel designed and built to be mechanically similar to the *Curiosit*y rover wheels. Its development is funded by NASA-JPL and the work is being done by the Machine Learning and Instrument Autonomy group at JPL.

The final wheel test rig configuration is shown in Figure 2.1. The wheel itself has a radius of 0.27 m and has a mass of approximately 30 kg. 43 grousers are mounted around the wheel. Each grouser is 0.445 m long and extends 0.11 m off the pressure pad. The wheel is equipped with four externally wrapped pressure pad quadrants, which are overlapped and attached to the wheel to cover the entire outer surface area. The grousers are mounted overtop the pressure pad and fastened to the rim of the wheel. The wheel structure is mounted on the test rig as is shown in Figure 2.1. It is important to note that the wheel is powered and drives the rig; the rig does not drive the wheel. The rig's purpose is to remain on the rails laid on the ground and thus guide the wheel down the trough of Mars simulant. It also houses the electrical equipment required for data collection. A string potentiometer is mounted between the starting end of the trough and the rig to measure slip ratio throughout each data collection run. On the wheel is an IMU sensor for the recording of wheel angular location.

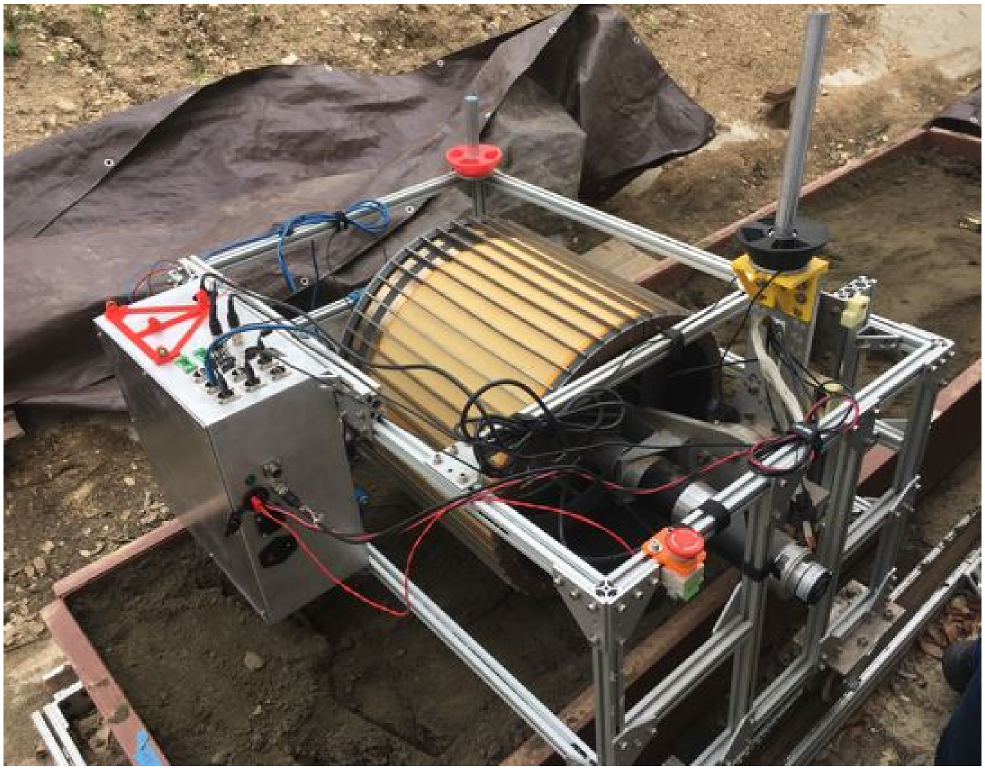


Figure 2.1: The *Barefoot Rover* project; wheel, rig, and accompanying instruments.

The wheel is powered by a Midwest Motion DC motor. It's rated for operation at a motor current of 5.7 amperes and has a gear ratio of 195.26:1 with a gear efficiency of 70%. The motor can operate in a "fast" or "slow" configuration, at 301 rpm or 113 rpm, respectively, and a corresponding wheel rpm of 1.54 and 0.58, respectively. All test runs have been done with the motor in the "fast" configuration. Forward velocity will of course change as a result of wheel slip or terrain patterns, such as small smooth dunes, but is nominally 0.043 m/s. The motor is controlled by an ESCON 50/5 Servo controller which controls the motor speed through integrated potentiometers. In addition, the ESCON 50/5 outputs motor current draw to the Arduino mounted inside the electronics box on the rig.

The electronics box on the rig holds the components necessary for collecting and storing the data obtained during each data collection test run. The main components in the electronics box are:

- NUC (Next Unit of Computing) computer for running test run python scripts and storing data

- Two DC to DC converters to step 48 V down to 12 V
- Arduino to take in ESCON outputs, IMU data
- National Instruments Multifunction I/O device (Black) for measuring strain gauge resistance in F/T sensor
- National Instruments Multifunction I/O device (White) to process data for the NUC

There are seven troughs, each filled with a type of Mars simulant. The seven simulants are:

- Mars Mojave 2mm Simulant (MMS 2MM)
- Mars Mojave Intermediate Simulant (MMS Intr)
- Mars Mojave Coarse Simulant (MMS Coarse)
- Minus-30 (Mins 30)
- GRC-01
- Wedron 730 (WED 730)
- Best 110 (BST 110)

These Mars simulants are dry low-cohesive sands representative of Martian terrain. These simulants were created to mimic the chemical composition and grain sizes of Mars terrain encountered in past missions to the red planet. The wheel is tested on all seven troughs. A single data collection run is carried out by powering the wheel to roll down the length of the trough at a constant 1.54 rpm. The wheel is not run in reverse after the initial forward roll because the data collected would not be representative of an undisturbed planetary surface. The goal is to assess whether the collected data can be used to infer changes in terrain which is being done by training machine learning classifier models on the collected data and subjecting them to new collected data that the models have not seen.

2.2 Background to Terrain Parameter Estimation

Bekker - Wong [5, 6] terramechanics theory are used to estimate the cohesion c and angle of friction ϕ of a terrain. Shear stress at the wheel-terrain interface is a crucial piece of information to determine cohesion and angle of friction. Ding et al. [4] proposed three methods for solving for unknown terrain parameters based on in-situ rover data. He developed a simplified integral model by linearizing the normal and shearing stresses in order to decouple the involved functions. He solved for terrain parameters using a linear least-squares estimator, hypothetical data and assumed typical values for terrain shear deformation modulus. The estimated values of terrain parameters were close to the measured soil parameters. The experimental data used by Ding was obtained from a single-wheel testbed and from a four-wheeled rover. Using Ding's method requires knowledge of the resistance moment which was not measured by the *Barefoot Rover* wheel.

Sullivan et al. [1] employed a method to estimate cohesion and angle of friction that relied on motor current data from MER *Spirit* and *Opportunity*. He determined shear stress from motor current and derived equations based on ratios of electromechanical work to estimate terrain parameters. These equations were derived from a simplified shear stress model which does not consider wheel slip and contact area. Both these values were not known and could only be roughly approximated using images. Using electromechanical work was a more accurate method to use at the time.

However, the *Barefoot Rover* wheel is equipped with a pressure pad to measure contact pressure and a string potentiometer is mounted externally to measure wheel slip. Therefore, Sullivan's method of computing shear stress from motor current can be used and input into the shear stress model which takes into account changing wheel slip and contact area.

2.3 Terramechanics Theory

Terramechanics is the study of vehicle-terrain interaction, with specific focus on wheeled or tracked vehicles. The study of terramechanics is used for either engineering vehicles to improve performance in a given terrain or understanding how a terrain will react under a specific vehicle loading. Well known vehicle-terrain interaction models have been developed by Bekker and Wong [5, 6]. The models assign parameters to specific soils which characterize their response to loading. Both these models aimed to improve vehicle performance. Ding [4] developed a terramechanics model specific to a lugged rover wheel with the intention of estimating terrain parameters based upon knowledge of wheel performance.

Bekker's work provided the foundation for further studies done by Wong and Ding. Both Bekker and Wong used a bevameter - a tool to apply a known pressure to terrain and measure output sinkage of a plate of a particular dimension - to analyze pressure-sinkage curves. Bekker derived his terramechanics theory based on these curves. Bekker and Wong studied a variety of terrains such as sand, sandy loam, clay, mud, grassy fields, and snow. It was noted by Bekker that variability in the measurement response was seen to increase in terrains containing organic material or water due to the non-homogenous nature of the soil itself. Snow and terrain containing organic material have greater compressibility. Sands, or mineral terrains, are homogeneous terrain that exhibit low variability in pressure-sinkage measurements. Mars regolith is a loose mineral terrain of varying grain sizes.

Bekker and Wong studied a variety of mobility platforms including rigid wheels, chain tracks, rigid tracks, pneumatic tires, and screw propulsion. Ding focused primarily on the lugged rigid wheel most similar to a common rover wheel. Lugs, or grousers, are the edged protrusions on a wheel outer surface to aid in traction and thrust generation. Rigid wheels have been used on all past and current Mars rovers. A rigid wheel was used in the *Barefoot Rover* project and testing was carried out in Mars simulants. Therefore, only rigid wheel interaction in Mars simulant terrains will be discussed in this thesis.

2.3.1 Modelling Terrain

Terrain can be treated as either a plastic or elastic material. Most classical terramechanics material treats terrain as an elastic material until a maximum load or shear stress is met; like solid mechanics theory.

The Mohr-Coulomb criterion is applied to define the shear stress in relation to the normal stress at the wheel-terrain interface:

$$\tau = [c + \sigma \tan \phi] \tag{2.1}$$

Terrain is modelled as an elastic material until the maximum shear stress has been reached. At this point, the terrain is in plastic equilibrium. If the load is increased further, it then enters a plastic state. The shear curve of this increasing load represents an ideal plastic material and is then fitted with the equation:

$$\tau = [c + \sigma \tan \phi][1 - \exp \frac{-j}{K}] \tag{2.2}$$

Where:

- c is terrain cohesion. Cohesion is the shear strength component of terrain that is due to electrostatic forces between grains and is influenced by hydration, compaction and soil salt content [7]. Cohesion is one of the parameters to be estimated.

- ϕ is the angle of friction of terrain. This terrain parameter is the shear strength component of terrain that is due to grain roughness, size, and the resulting friction generated through grain interaction. Angle of friction is influenced by grain size and roughness [8]. Angle of friction is one of the parameters to be estimated.

- The shear stress τ is a known value determined by the motor torque, which is determined by motor current draw.

- The normal stress, or pressure σ is a known value determined by the load sensed by the pressure pad and the total wheel contact area.

- The slip deformation j is a function of slip ratio and is a known value for each data collection run.

- The shear deformation modulus K is the third parameter to be estimated. It is of less interest to planetary science than cohesion and angle of friction, and is to be estimated because it is a wheel-terrain interaction value that represents a curve fit term for equation 2.2. Its estimated values can be used as a third validation to reasonable numbers, and K is specific to the wheel-terrain interaction.

- The wheel radius r is a known value.

- The wheel contact arc θ is measured by the pressure pad by the number of pixels sensing ground along the wheel arc.

The shear stress τ can be estimated from the motor torque output, and governs the wheel's instantaneous forward thrust. Shear stress is a function of cohesion and angle of friction and thus they also influence forward thrust. A decrease in either parameter will decrease maximum thrust. If a torque greater than the shear strength of a terrain is applied, the wheel will slip.

Additional methods of terrain modelling, like the finite element method, are available. This method requires the user to have prior information about the terrain and necessitates the use of large computational resources. Mars regolith is practically unknown terrain, and would require erroneous assumptions to be made for use of this method. A Mars rover has limited computational resources which are not sufficient to perform the finite element method in-situ and in near-real-time, as is the intention of this research for the *Barefoot Rover* project. This additional method of modelling is impractical for deployment on a Mars rover.

2.4 Rigid Wheel Terramechanics

Bekker-Wong [5,6] classical terramechanics theory has been the basis for studying the terramechanical properties of planetary terrain. Therefore, it is important to first understand this foundation. Both Bekker and Wong developed wheel-terrain interaction models for rigid wheels that are driven, towed, stationary, and pushed. *Spirit*, *Curiosity*, and Mars 2020 (*Perseverance*) wheels are rigid [9]. The *Barefoot Rover* project uses a single rigid powered wheel mounted on a test rig. Thus, this thesis is limited to the discussion of powered and rigid wheel-terrain interaction. An additional requirement is to consider the grousers, or lugs, on the rover wheel. These are common features on planetary rovers and are used to increase forward thrust.

Wheel Vertical Loading

For a slow-moving rover, a quasi-static state can be assumed to estimate the load on the wheel given that dynamic forces are negligible under speeds of 10 cm/s [10]. On the Barefoot Rover wheel, the pressure pad wrapped around its outer surface allows for explicit measurement of the vertical loading experienced by the wheel. The vertical load force balance is as follows [10]:

$$W = rb\int_{\theta_2}^{\theta_1} \sigma(\theta)\cos\theta d\theta + \int_{\theta_2}^{\theta_1} \tau(\theta)\sin\theta d\theta \tag{2.3}$$

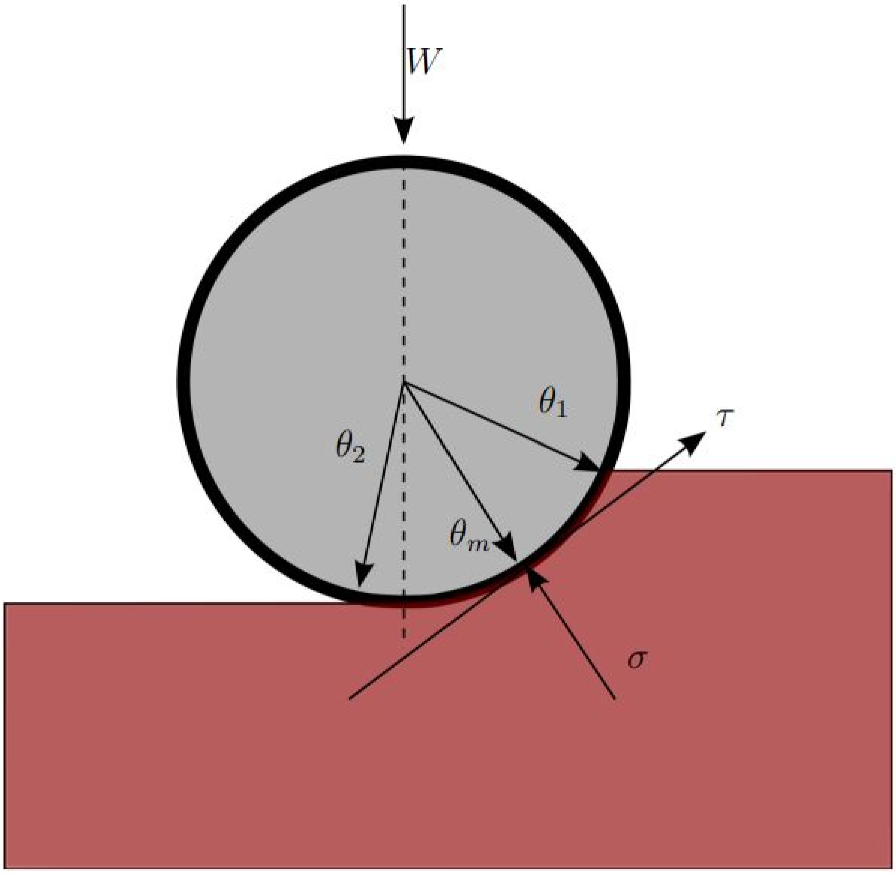


Figure 2.2: Free body diagram of the vertical forces acting on the rigid wheel. These forces are the weight on the wheel W, the the normal stress σ and the shear stress τ, which act over the contact area represented by angle θ [11].

The terrain-wheel interaction consists of a continuous radial normal stress and tangential shearing stress [4]. These are divided into forward and rear sections. The rear part occurs from θ_2 to θ_m and the forward part occurs from θ_m to θ_1, shown in Figure 2.2. Where these two parts meet is where maximum normal and shearing stress occurs, and is shown by angle θ_m.

Shibly [10] states that the stress distributions are nearly linear and thus a linear distribution can be observed. This distribution can then be approximated by linear functions:

$$\sigma_1(\theta) = \frac{\theta_1 - \theta}{\theta_1 - \theta_m}\sigma_m \tag{2.4}$$

$$\sigma_1(\theta) = \frac{\theta}{\theta_m}\sigma_m \tag{2.5}$$

$$\tau_1(\theta) = \frac{\theta_1 - \theta}{\theta_1 - \theta_m}\tau_m \tag{2.6}$$

$$\tau_1(\theta) = \frac{\theta}{\theta_m}\tau_m \tag{2.7}$$

Shibly found an average difference of 9.34% between the linearized and original nonlinear stress distribution which is considered sufficient to accurately represent the nonlinear equations. θ_2, the wheel exit angle, is assumed to be zero [10]. 2.3 can now be written as:

$$W = rb\int_{\theta_2}^{\theta_1} \sigma_1(\theta)\cos\theta + \tau_1(\theta)\sin\theta d\theta + \int_{\theta_2}^{\theta_1} \sigma_2(\theta)\cos\theta + \tau_2(\theta)\sin\theta d\theta \tag{2.8}$$

The term *rb* represents the contact area of the wheel and terrain.

Wheel Sinkage and Loading

Figure 2.3 shows the relationsip between the wheel entrance angle, θ_1 (θ_c), wheel sinkage z, and wheel radius r.

The sinkage of a wheel is defined as:

$$z = r\cos\theta_1 - r \tag{2.9}$$

Where θ_1 is the contact arc of wheel and terrain. If θ_2 is not assumed to be zero as is the case here, the contact arc is $\theta_c = \theta_1$ - θ_2, where θ_2 is negative from the vertical. Inversely, the

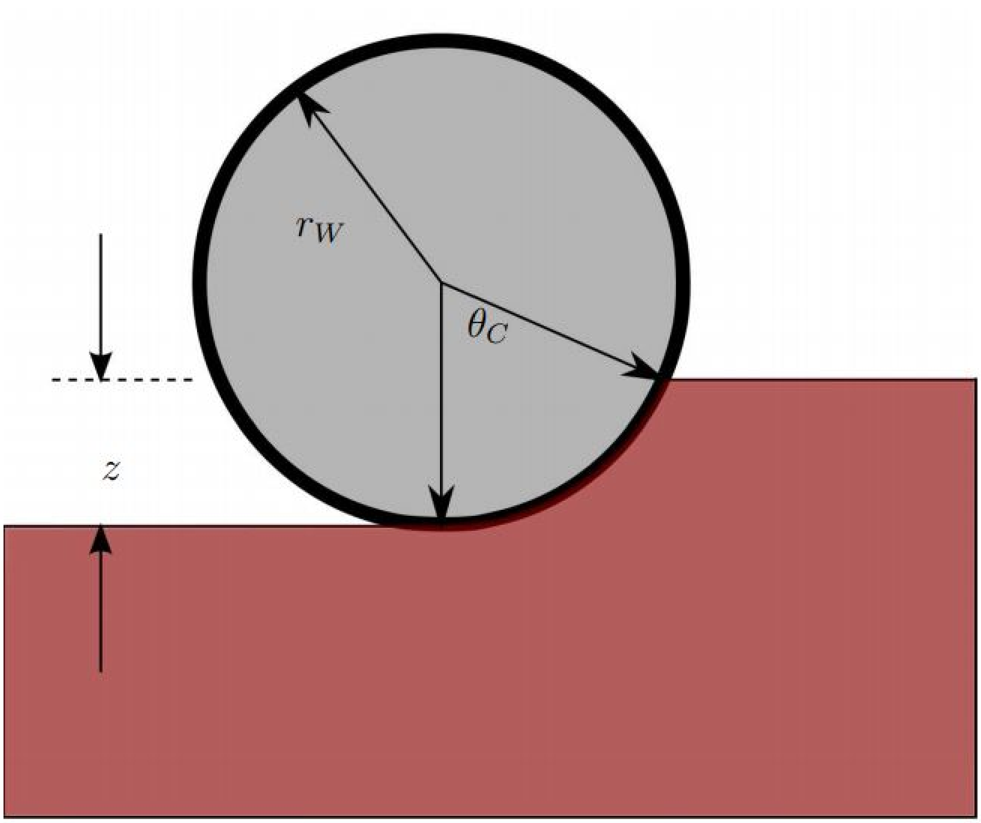


Figure 2.3: Geometric relationship between wheel radius r_w, sinkage z and contact arc angle θ_c, or given the previous assumptions, θ_1 [11]

contact arc can be determined by sinkage:

$$\theta_c = \arccos[\frac{(r - z)}{r}] \tag{2.10}$$

Bekker [5] used a bevameter on terrains such as snow, sand and mud to characterize a normal pressure and sinkage relationship. These relationships were established with circular or rectangular plates that were hydraulically pressed into the terrain with a known pressure. Their sinkage was then recorded with increasing pressure. These plates, according to Bekker, behave equivalently (purely in terms of applied load and sinkage) to wheels of corresponding dimensions. This relationship, shown in Equation 2.11 also reveals information about the terrain, such as its compressibility and uniformity:

$$p = [\frac{k_c}{b} + k_\phi]z^n \tag{2.11}$$

Where k_c, k_ϕ and n are terrain parameters estimated by applying a curve of best fit to collected pressure p and sinkage z data. The sinkage exponent n offers an indication of the compressibility of the terrain, as it is indicates the amount of sinkage that occurs given an applied pressure. Different terrains will exhibit different sinkage exponents based on how much a wheel can sink in the terrain. Nonuniform terrain will show high variability in the resulting pressure-sinkage plots.

However, deploying a bevameter onto another planetary surface as part of a rover instrument suite is not feasible given its mass and power requirements. Especially in the context of auntonomous rover developments, a bevameter is even more infeasible. The rover would be required to stop for each measurement even if the bevameter were automated, whereas, an instrumented rover wheel like that in the *Barefoot Rover* project can supply the same information as a bevameter passively and while other science is being conducted. Additionally, any estimation of the sinkage of the bevameter would be empirical, whereas the *Barefoot Rover* wheel can explicitly measure sinkage without need for manual or visual estimation.

Sinkage is a critical variable for estimating the cohesion and angle of friction of terrain. In Section 3.4, contact area and contact arc, both functions of sinkage, are critical variables needed for the terrain parameter estimation method. Cross [11] states that wheel sinkage has been difficult to measure autonomously in the past. He mentions several studies that have been conducted to try to present a robust method to measure sinkage autonomously. While these methods produced promising results, none are as accurate as an explicit measurement directly from the wheel itself.

Wheel Torque

The free body diagram of force around the axis of rotation of the wheel is shown in Figure 2.4. This is the relationship between motor/wheel torque T and wheel-terrain interface shear stresses.

The torque output by the wheel's motor is needed to overcome the shear stress experienced at the wheel-terrain interface . The force balance equation about the wheel axis is as follows:

$$T = r^2 b \int_{\theta_2}^{\theta_1} \tau(\theta) d\theta \tag{2.12}$$

Where b is the wheel width in contact with ground. Taking into account the assumption that θ_2 is zero and the previous linearization assumptions verified by Shibly, Equation 2.13 can be expanded to:

$$T = r^2 b \int_{\theta_m}^{\theta_1} \tau_1(\theta) d\theta + \int_{0}^{\theta_m} \tau_2(\theta) d\theta \tag{2.13}$$

According to Ding [4], the motor torque can be estimated from motor current data. For every data collection run using the *Barefoot Rover* wheel, motor current is recorded, and this data is used to compute motor torque for estimating terrain parameters.

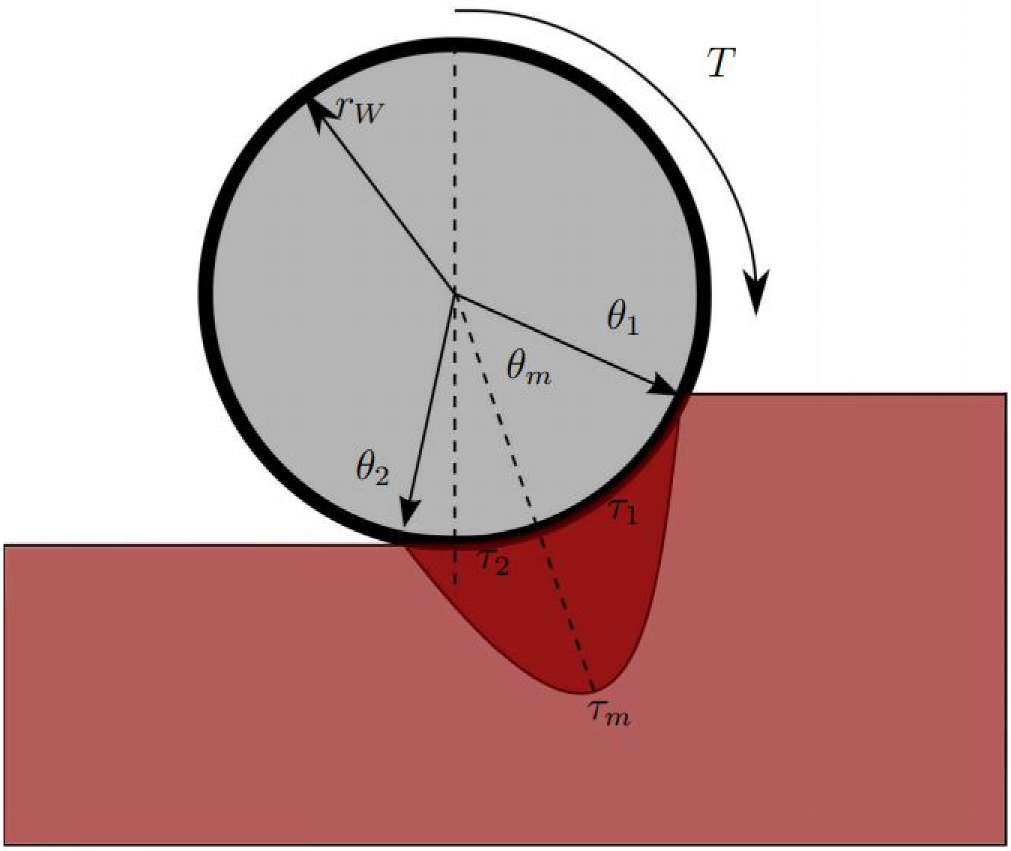


Figure 2.4: Relationship between shear stresses at wheel-terrain interface and wheel torque T [11].

Wheel Slip

Wheel slip refers to the difference in actual forward velocity versus intended wheel forward velocity, or, the difference in actual wheel forward displacement versus intended forward displacement. This effect is seen by the observer as “skidding”, or, the wheel spinning faster than its forward motion.

Slip velocity V_j, according to Higa [12], is the difference between the forward velocity r_w and the tangential component of the travelling speed of a wheel V_t, where $V_t = rw$, and is represented as follows:

$$V_j = rw(1 - (1 - i)\cos\theta) \tag{2.14}$$

Where the wheel slip ratio i is expressed as:

$$i = 1 - \frac{V}{rw} \tag{2.15}$$

Where V is the nominal forward velocity. Figure 2.5 shows the relationship between slip velocity V_j, wheel angular velocity w and forward velocity V.

For the *Barefoot Rover* project, wheel slip is measured by an external string potentiometer mounted on the rig. A string potentiometer measures linear extension by producing an electrical signal that is proportional to the cable’s extension. As the wheel rolls down the trough, and thus the rig with it, the string potentiometer records the distance travelled. This distance is compared to expected distance given the wheel angular velocity and time of the data collection run. It is possible, however, to see the result of slip in the pressure pad data and work is being done to estimate slip purely from the pressure pad data. Figure 2.6a and 2.6b show this variation observed in the data.

The slip deformation, j, is a function of slip ratio and is represented by the following equation [11]:

$$j = r[\theta_1 - \theta_2(1 - i)(\sin\theta_1 - \sin\theta_2] \tag{2.16}$$

Given the previous assumption that θ_2 is equal to zero, equation 2.18 can be expressed as:

$$j = r[\theta_1 - (1 - i)(\sin\theta_1)] \tag{2.17}$$

$$j = r[\theta_1 + i\sin\theta_1 - \sin\theta_1] \tag{2.18}$$

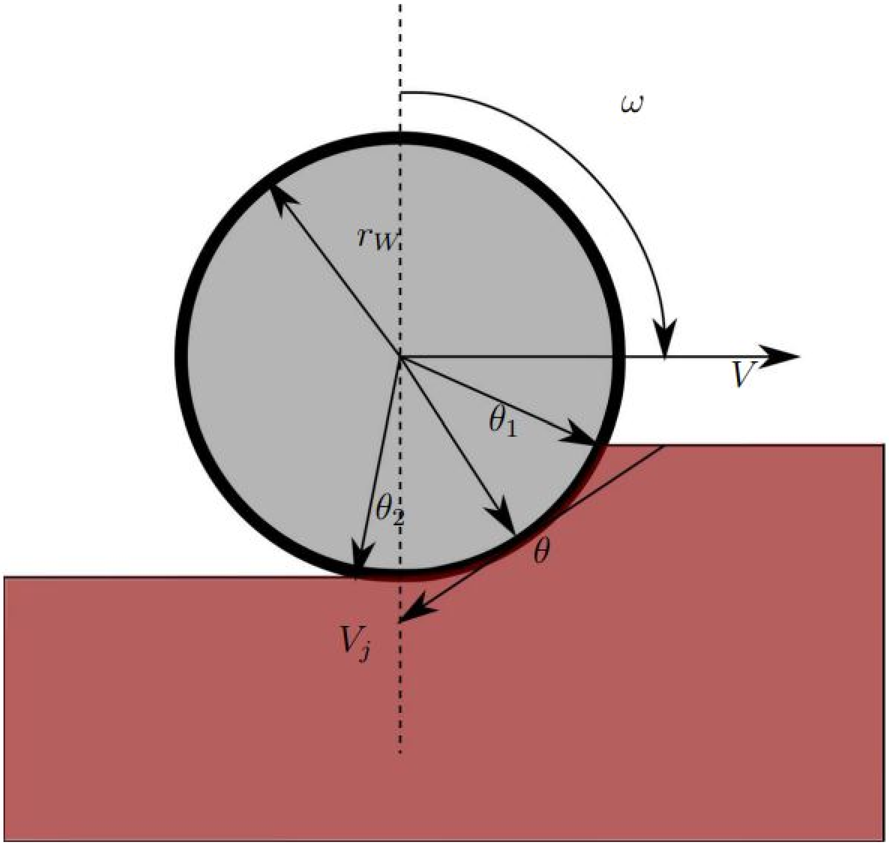
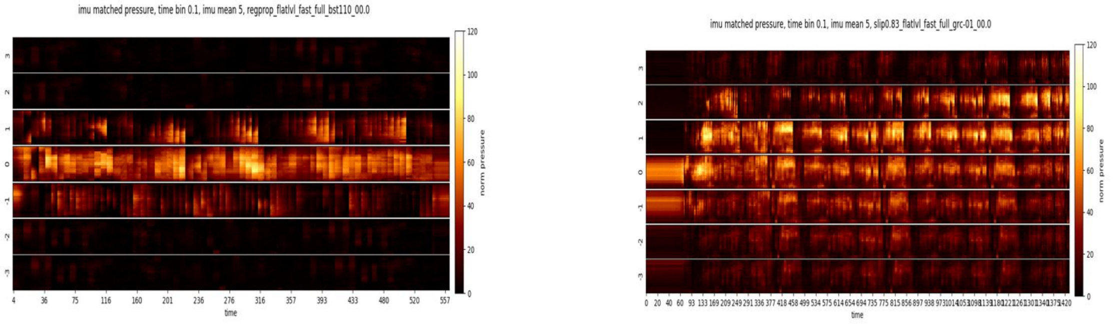


Figure 2.5: Slip velocity of a rigid wheel [11].



(a) Low wheel slip pressure pad data.

(b) High wheel slip pressure pad data.

Figure 2.6: Two conditions of wheel slip shown through a heatmap of the pressure pad data.

The shear stress along the wheel-terrain interface then becomes:

$$\tau = [c + \sigma \tan \phi][1 - \exp(\frac{-r}{K}\theta_1 + i \sin \theta_1 - \sin \theta_1)] \tag{2.19}$$

Cross [11] discussed methods proposed to detect the amount of slip a wheel is experiencing in-situ. These were using encoder data to compare variations in longitudinal velocity between two wheels, using encoder data in combination with z-axis gyro readings to detect any rotation about the z-axis compared to commanded rotation, and using motor current data to infer slip based on the difference between expected current draw. These methods, however, do not produce an explicit value of wheel slip, but only a flag to indicate that slip is occurring. While this is useful, it is necessary to obtain an accurate slip ratio for the estimation of terrain parameters.

As mentioned, the *Barefoot Rover* test setup utilizes a string potentiometer to measure slip. Though accurate, using a string potentiometer anchored to "zero" location is not feasible for a Mars rover to determine the slippage of its wheels. This is where another use of the pressure pad data may come to fruition. For actual deployment on Mars, it will be necessary to develop an alternative method of determining wheel slip and the pressure pad data offers a potential source of information to accomplish that.

With the addition of directly measuring contact pressure, the shear stress distribution at the wheel contact surface is now a function of known parameters and the three unknowns c, ϕ, and K, of which are to be estimated.

2.5 Machine Learning Classification

Machine learning is the process by which a computer is fed data in a way that allows the computer to learn the ability to process and perform the desired activity without the need to be explicitly programmed to do so or fed with similar or extra data. It is the study of the algorithms and statistical models that allow computers to perform these tasks without explicit instruction.

In traditional programming, a computer is fed data and a program is then run to determine an output using the provided data set. In machine learning, data and it's already known output is provided to the computer which are then both used to create a program, or model, which can provide the output when presented with new data. The model relies on patterns in the data to identify a solution.

In classification, a model outputs a decision as to what class the given example data belongs to. Binary classification and multi-classification refer to classification types which involve two classes in the data or more than two classes, respectively. Both types of classification will be discussed in this thesis.

2.5.1 Features

Features are the basis on which the machine learning model is trained. The data set is constructed as an m × n matrix, where m is the number of examples in the dataset, and n is the number of features. If the data is a time series consisting of four features, each time step is another example where information about the features has been collected. The labelled class of each example is a feature in addition to the three others. For example, the machine learning model for terrain class prediction discussed in this thesis uses the following four features for training:

- The wheel contact pressure σ
- The wheel slip i
- The motor torque T
- The terrain type class

If a wheel data collection run contains 300 time steps, or time bins of data, then the feature set is a matrix of dimensions 300 × 4. If a particular terrain type had 10 data collection runs performed on it, then the feature set describing that terrain is of the dimension 3000 × 4.

Feature engineering may be good practice when not enough information is presented in the baseline features one chooses to use, or if the data is an m × 1 time series where extracting features is necessary. It is obvious when there is not enough information contained in the existing features because the algorithm will not learn sufficiently to produce a model that can make predictions with reasonable accuracy. Statistical features are common approaches to feature engineering. For example, the mean or variance of a time series could be used as a feature. These features are engineered from existing knowledge and may provide further insight for the construction of a more robust model.

2.5.2 Training, Testing, and Validation

The steps to creating a robust machine learning model may be summarized by the words; training, testing, and validation. Of course, before that, one must ensure the data being used is representative of the problem and offers unbiased insight into say, different terrain classes. And one must ensure the features being used or that have been engineered offer enough information for the model to be trained.

The three keywords that are the title of this subsubsection indicate the portion of data being used from the overall data set. For training, approximately 70-80% of the data is used. For testing, 20-30% is used. The validation set may not always be necessary but is good practice. This set of data is 10-15% of the data set and is used after training and testing to validate the robustness of the model. However, if cross-fold validation to determine optimal hyperparameters is done, the validation set is not necessary, because cross-fold validation trains the model on many different arrangements of training data from the overall data set and iterates over the hyperparameters in use to find the most optimal training scenario. Cross-fold validation ensures the model generalizes well to unseen data and helps to avoid over-fitting.

The test set is data which the model has not seen in training. The labelled classes are not included in this set, so prediction is done on one less feature than training is performed on. Testing is done to evaluate the predictions made, the selected error metric(s), and identify whether insufficient features may be in use or if overfitting is occurring. Overfitting is the term to describe whether the model can predict well only on the data that it has been presented with and is not robust to new data.

2.5.3 Learning Methods

There are three types of learning in Machine Learning. These are supervised, unsupervised, and semi-supervised learning. Unsupervised and semi-supervised learning methods are sometimes more complex as these are used for datasets without labelled data (and in the case of semi-supervised, for both labelled and unlabelled data). A common use of unsupervised learning is clustering. The algorithm will assign the data into a number of clusters that it has identified to build the model. New data is continuously binned into a class. Anomaly detection is commonly done through unsupervised clustering.

Supervised learning is carried out when the data classes are already known. For each example in a feature set, the labelled class is included in the training set. The model is then trained via prior knowledge to make predictions on new and unseen data.

Since the data collected by the *Barefoot Rover* data has labels (the class of simulant the data is collected from), unsupervised or semi-supervised learning methods are unnecessary and will not be discussed further.

2.5.4 Decision Tree Classification

A decision tree at its core is a series of if statements, or, yes or no checks. A series of questions is asked until the solution is determined. Decision trees take a top-down approach. Meaning, the roots, or the root node, is at the top of the decision tree and the terminal nodes, or, the leaf nodes, are at the bottom. The root node represents the population of input data of which is then divided into two predictor spaces that continue down the tree until leaf nodes are reached. Decision trees are said to be 'greedy' because they look for the best split at the current node and do not consider future splits that may lead to a better tree/model [13].

Decision trees are solely used as supervised learning algorithms. They have the advantageous attribute that they can be used for both regression and classification problems. In the case of this thesis, the classification decision trees are used. The decision tree is trained on prior knowledge and creates a set of decision rules (the model) whereby new data can be applied to these rules and a prediction can be made [14]. Throughout this process, the features fed to the tree cannot change.

Figure 2.7 shows a simple decision tree example. Within a decision tree, each internal decision node corresponds to a feature, and each leaf node to a labelled class [14]. The pseudocode for the decision tree algorithm is as follows:

- Place the best feature of the dataset at the root node.
- Split the training set into subsets, with each subset containing data with the same value for a feature.
- Repeat the previous steps on each subset until leaf nodes are found for every tree branch.

The 'best feature' refers to the feature that has the most influence on the classification decision. The two most common criterion for determining the best feature is information gain

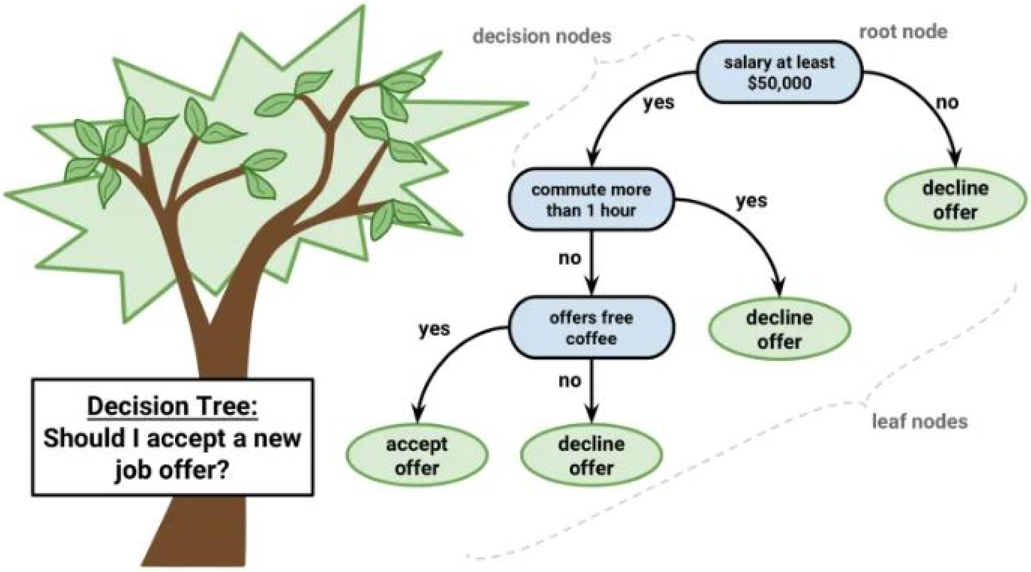


Figure 2.7: An example of a basic decision tree [14].

or the gini index.

Information gain estimates the information contained within each feature. It uses information theory to measure the entropy of a random variable, *x*. The entropy of this random variable, or feature in this case, is defined as:

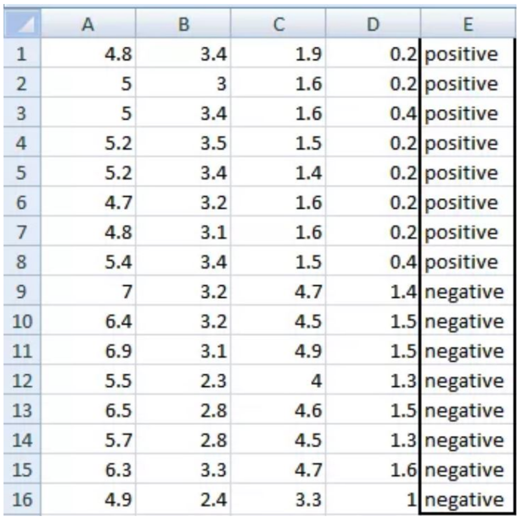
$$H(X) = -\sum_{x \epsilon X} p(x) \log p(x) \tag{2.20}$$

Where *H* is entropy and *p(x)* is the probability of the class. The entropy of a feature allows for its information gain to be calculated. An example of building a decision tree from [14] will be used to present this concept.

Shown in Figure 2.8, features A, B, C, D are the predictors and E column class labels are the target variable. The data contained in features A, B, C, D are continuous and must be discretized for construction of the decision tree.

Random values were chosen to categorize each feature and are shown in Table 2.1 below.

The two steps to calculate the information gain for each feature are to calculate the entropy of the target and the entropy for each feature. By subtracting a feature's entropy from the target entropy, information gain is found. In this particular example, it is a binary classification with equal class representation and thus the target entropy can be estimated as one. Calculating the information gain for feature A is done as follows:



	A	B	C	D	E
1	4.8	3.4	1.9	0.2	positive
2	5	3	1.6	0.2	positive
3	5	3.4	1.6	0.4	positive
4	5.2	3.5	1.5	0.2	positive
5	5.2	3.4	1.4	0.2	positive
6	4.7	3.2	1.6	0.2	positive
7	4.8	3.1	1.6	0.2	positive
8	5.4	3.4	1.5	0.4	positive
9	7	3.2	4.7	1.4	negative
10	6.4	3.2	4.5	1.5	negative
11	6.9	3.1	4.9	1.5	negative
12	5.5	2.3	4	1.3	negative
13	6.5	2.8	4.6	1.5	negative
14	5.7	2.8	4.5	1.3	negative
15	6.3	3.3	4.7	1.6	negative
16	4.9	2.4	3.3	1	negative

Figure 2.8: Data example from [14].

A	B	C	D
≥ 5	≥ 3	≥ 4.2	≥ 1.4
< 5	< 3.0	< 4.2	< 1.4

Table 2.1: Random values for feature categorization.

- For Feature A ≥ 5 and class == positive: $\frac{5}{12}$
- For Feature A ≥ 5 and class == negative: $\frac{7}{12}$
- $Entropy(5,7) = -1[(\frac{5}{12})\log_2(\frac{5}{12}) + (\frac{7}{12})\log_2(\frac{7}{12})] = 0.9799$
- For Feature A < 5 and class == positive: $\frac{3}{4}$
- For Feature A < 5 and class == negative: $\frac{1}{4}$
- $Entropy(3,1) = -1[(\frac{3}{4})\log_2(\frac{3}{4}) + (\frac{1}{4})\log_2(\frac{1}{4})] = 0.8113$
- $Entropy(Target, A) = P(\geq 5)E(5,7) + P(< 5)E(3,1) = (\frac{12}{16})(0.9799) + (\frac{4}{16})(0.8113) = 0.9377$

The information gain for feature A is then found by subtracting the entropy of feature A from the entropy of the target which produces an information gain of 0.06225. Doing these same steps for features B, C, and D, it is found the B has the highest information gain of 0.7071 and hence is places at the root node.

The gini index evaluates how often a randomly chosen element would be incorrectly identified [14]. The lower the gini index, the better. The same data in Figure 2.8 is used and the data is yet again categorized into the same random values shown in Table 2.1.

The gini index calculation for feature A is as follows:

- For Feature A $\geq$ 5 and class == positive: $\frac{5}{12}$
- For Feature A $\geq$ 5 and class == negative: $\frac{7}{12}$
- $gini(5,7) = 1 - [(\frac{5}{12})^2 + (\frac{7}{12})^2] = 0.486$
- For Feature A < 5 and class == positive: $\frac{3}{4}$
- For Feature A < 5 and class == negative: $\frac{1}{4}$
- $gini(3,1) = 1 - [(\frac{3}{4})^2 + (\frac{1}{4})^2] = 0.0.375$
- $gini(Target, A) = (\frac{12}{16})(0.486) + (\frac{4}{16})(0.375) = 0.4582$

Following these same steps to calculate the gini index for features B, C, and D results in feature C yielding the smallets gini index of 0.2 and is thus considered the best feature and placed at the root node.

For non-binary classification, the calculation of information gain or gini index is increased in complexity but still follows the same steps. The categorization (discretization) process will require additional cases for categorization thus increasing the number of terms within the entropy or gini index calculations. All decision trees, however, are built by making splits at features, with the best feature at the root node which is dependent on the criterion selected.

Decision trees are prone to overfitting, which in the case of the decision tree, often means that a deep tree with many branches has been constructed based on nuances within the data that its been presented with. Overfitting reduces accuracy of prediction on unseen data. Overfitting can be reduced by pruning, which is a method particular to decision trees.

Pruning is the identification of leafs that are producing losses in the model and cutting up to previous splits until a positive gain is found [13].

The advantages of a decision tree are that it is easy to explain and visualize. The logic used by a decision tree is the same approach by which humans make decisions. Additionally, there are less hyperparameters to tune compared to other algorithms.

The disadvantages of a decision tree are that it is likely to overfit and steps to mitigate overfitting must be taken. Decision trees produce a lower prediction accuracy for a given dataset when compared to other machine learning algorithms. Additionally, calculations for the construction of the decision tree can become complicated when there are many class labels. The Random Forest is a type of decision tree algorithm that reduces overfitting and increases accuracy.

2.5.5 Random Forest

The decision tree model suffers from limitations mentioned in the previous subsection, which leads to poor results and overfitting. The Random Forest model improves upon these limitations. The Random Forest is a model built of several decision trees. An umbrella algorithm which incorporates multiple algorithms is known as an ensemble method. The result of each internal decision tree is averaged, or in the case of classification, the majority class prediction is taken as the predicted class. The combination of many trees reduces overfitting and creates a more robust model to new data.

Each tree with a decision tree is trained on different subsets of the training set and is thus biased to that subset. However, the combination of these individual trees makes for a robust classifier. Within the trees and each node, a random number of features is selected from all the features. The feature that allows for the best split according to the information gain or gini index criterion is used for the split. This feature selection is repeated at the next node. Figure 2.9 shows an example of Random Forest architecture.

There are a few major hyperparameters of the Random Forest model to tune. These are:

- The number of trees in the forest.
- The criterion, gini or information gain.
- The maximum depth of the trees. It is the number of splits that the tree is allowed to make.

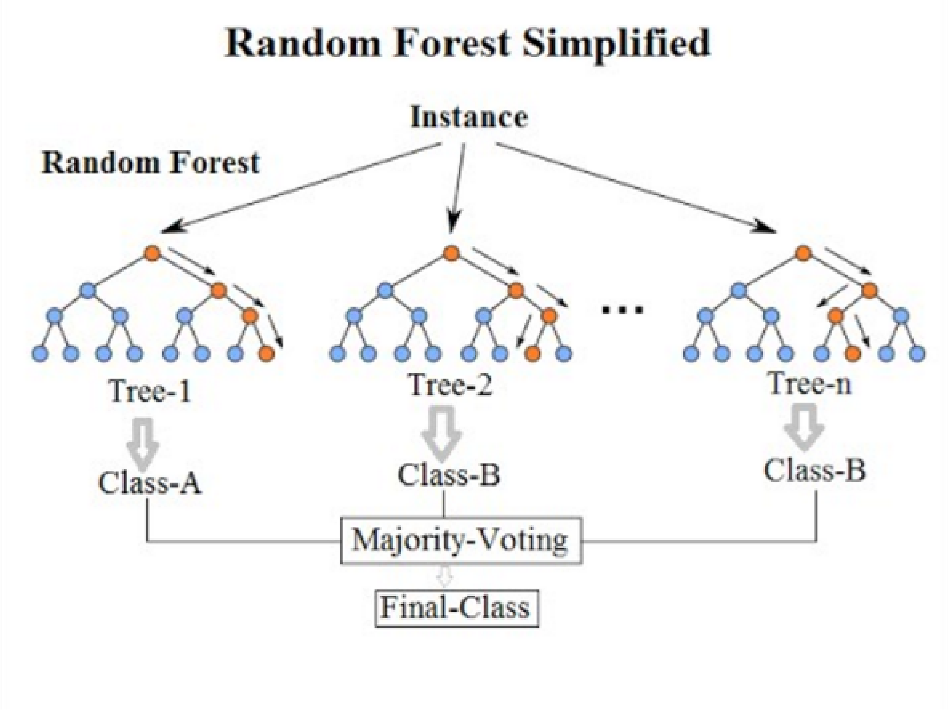


Figure 2.9: Random Forest architecture [15].

In general, the Random Forest was chosen over other classification algorithms for the following reasons:

- It is easy to visualize the solution and explain it as a result of the decision tree created by the algorithm. The solution is not a black box.
- It is simple to identify the most important feature(s) and gain insight into the data.
- The Random Forest is a low computationally complex algorithm [13].

2.5.6 Support Vector Machine

The Support Vector Machine (SVM) identifies classes by creating a hyperplane that divides the data into its respective classes. Figure 2.10 shows a number of possible hyperlane solutions between two classes, the blue circles and red squares. The x and y axis represent the two features being considered. The green lines are the possible solutions.

The SVM's purpose is to identify the best solution out of all the green lines. It finds this solution by maximimizing the the margin between the hyperlane and the closest points of each

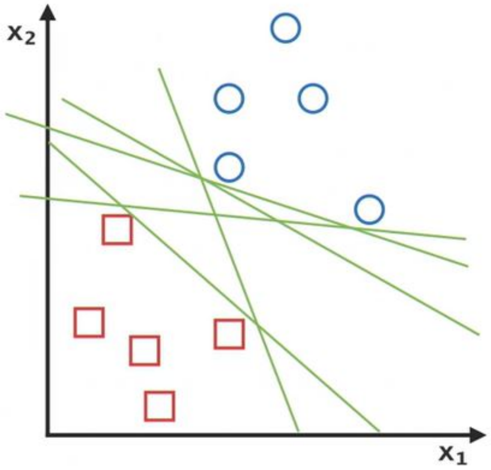


Figure 2.10: Support Vector Machine example [16].

class. Figure 2.11 shows an example of this maximization.

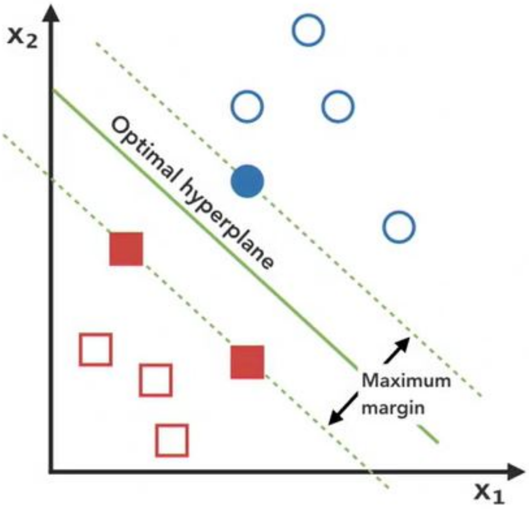


Figure 2.11: Support Vector Machine example showing maximum margin [16].

For a linear SVM to function properly the data must be linearly separable, meaning a line must be able to be drawn between the classes. Data is not always linearly separable, but the kernel method transforms data that is not linearly separable into a higher dimension where it can be separated [16]. Figure 2.12 shows this transformation. The data is projected from a 2-D space to a 3-D space and separated by a plane rather than a line.

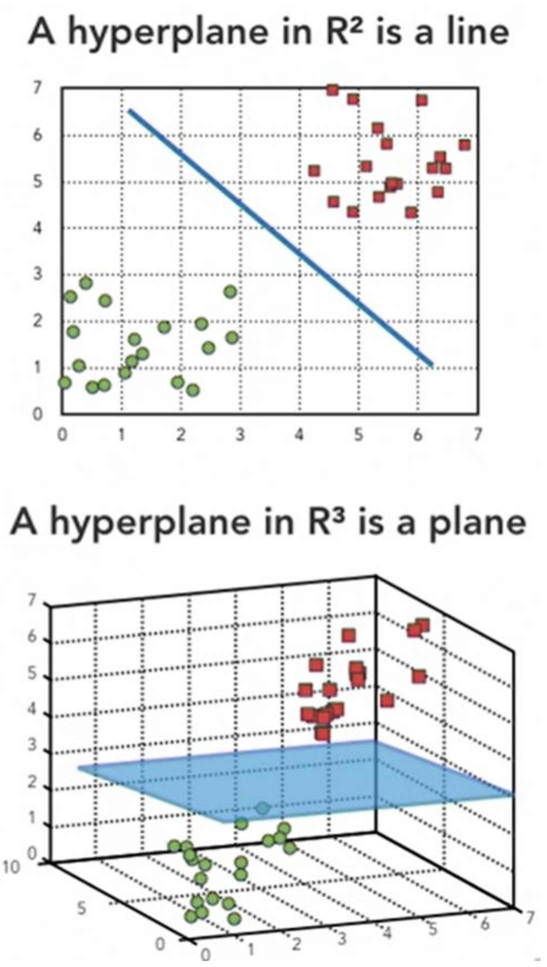


Figure 2.12: Support Vector Machine 2-D to 3-D space transformation [16].

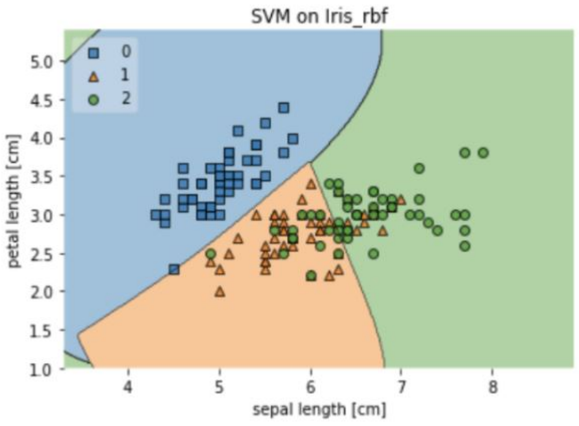


Figure 2.13: Non-linear SVM example results [17].

For multi-class classification, usage of an SVM becomes more difficult. One can construct a multi-class classifier by combining multiple binary classifiers [18]. These are "one-against-all” and “one-against-one” methods [18]. However, it is more computationally complex to use this type of combination [18]. In addition to “one-against-all” and “one-against-one” methods, a non-linear approach is necessary in this case. Figure 2.13 shows an example of non-linear results using the Radial Basis Function (RBF), a non-linear kernel method for SVM.

2.5.7 Logistic Regression

Logistic regression is a multivariate method for modeling the relationship between independent variables and dependent variables, or in this case, between features and the corresponding classes [19]. Because the sigmoid function is used, the logistic regression algorithm is able to model more complex data, or, data that is non-linear. Linear regression fails when data is non-linear. Figure 2.14 shows an example of the sigmoid function.

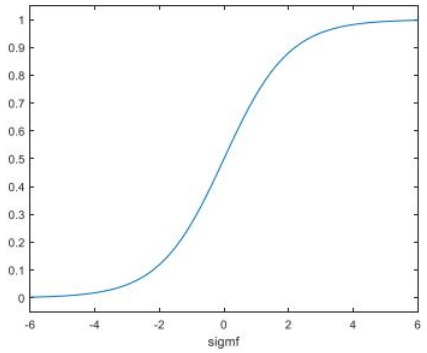


Figure 2.14: Sigmoid function [20].

The sigmoid function is represented by the following equation:

$$y = \frac{1}{1 + \exp -(mx + b)} \tag{2.21}$$

x and y are the x and y coordinates of each point, m is the slope of the line, and b is the y-intercept. Linear regression attempts to fit all the data to a line which, as shown in Figure 2.15, results in predictions outside of y = 1 and y = 0, which is impossible. The data at y = 1 and y = o represent the two classes (in binary classification), respectively.

When using logistic regression, it is assumed that it can handle the non-linear relationships between features and classes because a non-linear log transformation of the linear regression is used [19].

Logistic regression can be used for multi-class classification. it is a modified version which uses the softmax function rather than the sigmoid function [21]. The softmax equation is represented by the following:

$$\text{softmax}(x)_i = \frac{\exp x_i}{\sum_{j=1}^{n} \exp x_j} \tag{2.22}$$

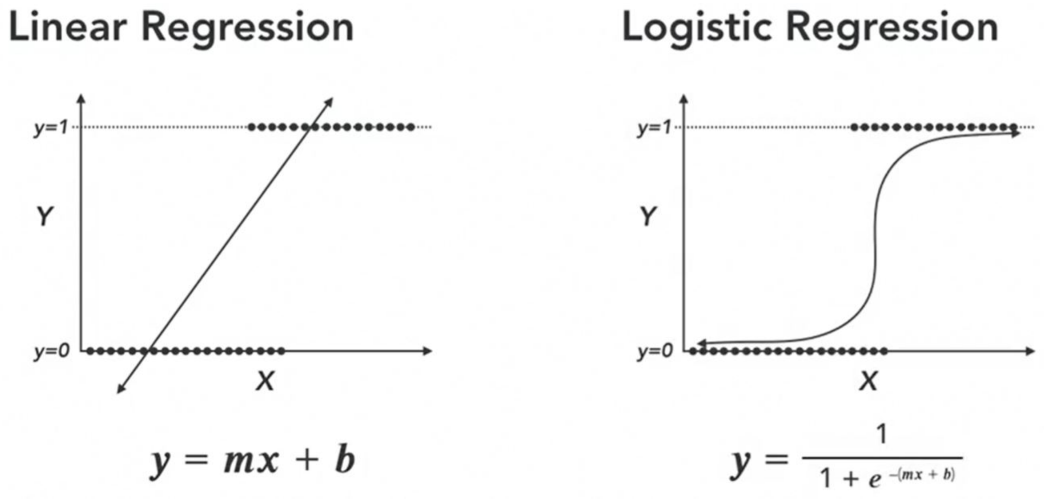


Figure 2.15: Linear vs. Logistic Regression [16].

.

Binary logistic regression can also be used similarily to binary SVM classification with "one-vs-all" and "one-vs-one" methods [21]. "One-vs-all" and "one-vs-one" do not require use of the softmax function.

There are drawbacks to the softmax function. Small differences in the input data are taken out of proportion and biases the classifier towards a particular class [21]."one-vs-all" and "one-vs-one" methods perform better but are more computationally complex. The solution to logistic regression classifier is less physically interpretable than a Random Forest decision tree, which given the problem at hand of classifying terrain with collected data from a wheel rolling on the terrain, physical interpretation is of utmost importance.

When Logistic regression is compared to an SVM, it can be expected that SVM will perform marginally better [22]. Logistic regression is useful for simpler problems but for more complex datasets that may not be linearly separable, SVMs are recommended [22]. Therefore, the logistic regression algorithm was not selected for comparison against the Random Forest. SVM is the selected comparison model.

2.5.8 Neural Networks

The neural network is a viable type of machine learning method to pursue for this research. The drawbacks to the neural network compared to the Random Forest are that a neural network is a black box method and not easily interpretable. Given the physical nature of the problem at hand, an interpretable solution where one can understand what the algorithm is picking up on from the data is necessary. In the case of the neural network, there are more hyperparameters to tune which themselves include a number of options, such as the number of hidden layer, the number of neurons in those layers, the training algorithm, the activation function, and the intial weights [23]. In the context of a rover computer, which is less computationally powerful than a cellphone, computational complexity is an important issue. Neural networks can often require large amounts of computational resources to run.

A neural network is a powerful tool but not always necessary, especially when considering classification using tabular data. Neural networks are best used for images, audio, and text data [23]. With these points in mind, the neural network will be used if the Random Forest or SVM models do not perform sufficiently well.

2.5.9 Recurrent Neural Networks

RNN's are a subset under the banner of neural networks of which the LSTM is the most well known and most powerful model in the subset. In the case of this research, RNN's and the LSTM are not applicable because they take into account temporal changes in data, or, time dependent data [24]. The data in this case is plotted through time, but it is not time dependent. Terrain does not change with respect to time it changes with respect to physical conditions. As such, RNN's were not considered further.

2.5.10 Multi-Layer Perceptron

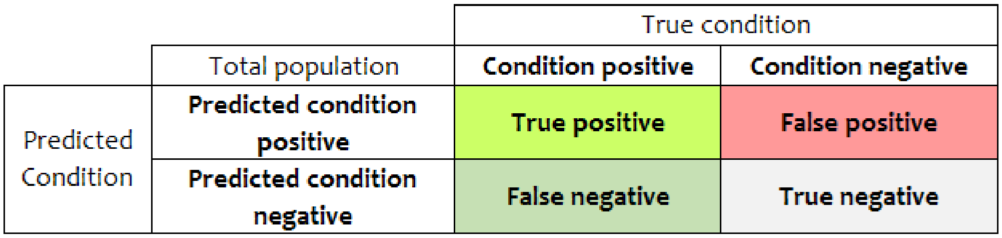
Multi-Layer Perceptrons were most widely used for computer vision, though, they are now considered obsolete when compared to new methods like Convolutional Neural Networks [25]. MLP's are time invariant and thus can be given time independent data such as the data used in this thesis. The MLP would therefore classify the terrain at each time step in the data without considering the previous time step. The MLP is an algorithm to consider if the Random Forest or SVM models do not perform sufficiently well.

2.5.11 Performance Metrics

In order to express the performance of a machine learning model, it is necessary to select a metric to do so. A common metric used for classification problems is the confusion matrix.

Confusion Matrix

In binary classification, the confusion matrix categorizes the predictions into four types: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) [26]. True Positive and True Negative indicate a correct classification of either the positive (1) or negative (0) class. False Positive and False Negative indicate an incorrect prediction of class [26]. The confusion matrix displays how many correct and incorrect predictions the model made. Figure 2.16 shows a confusion matrix example.



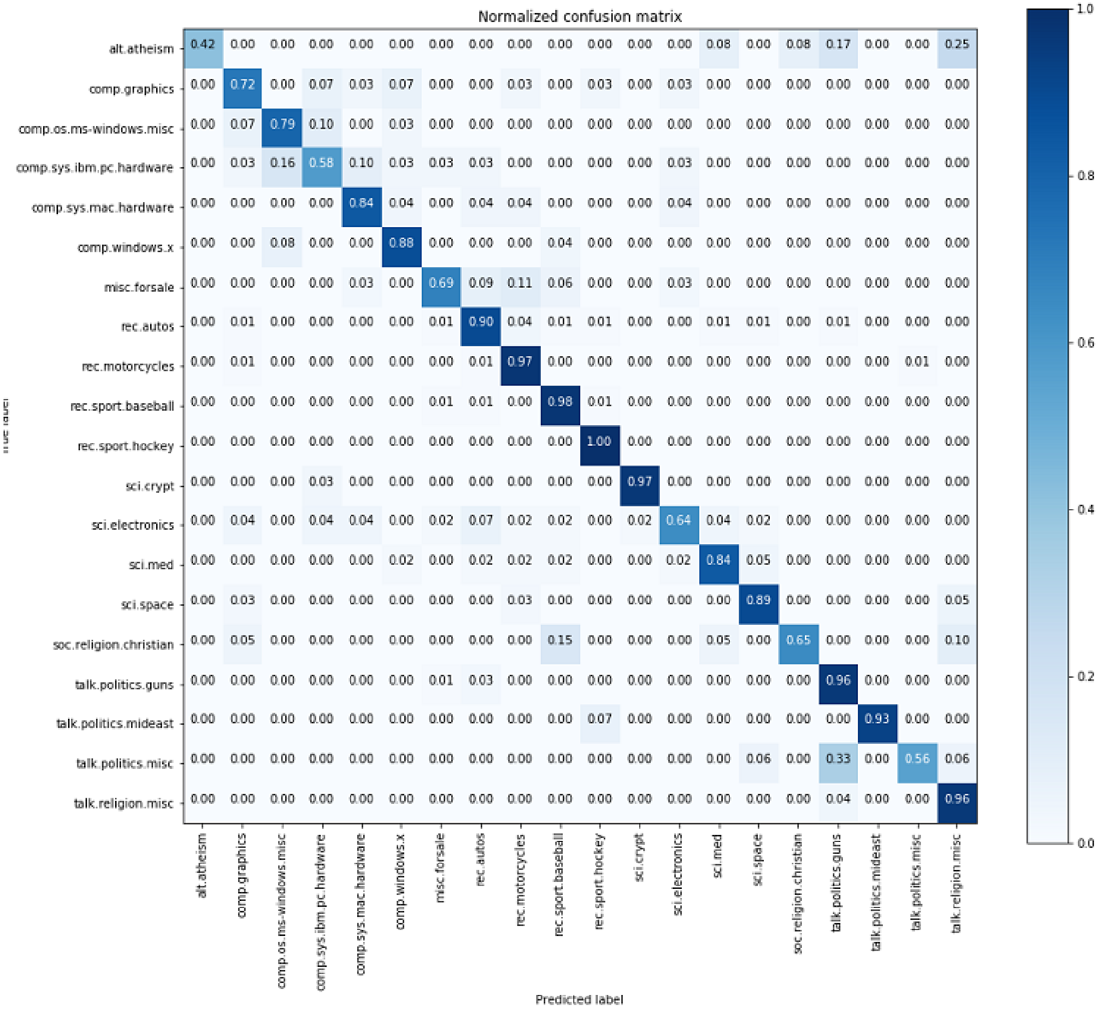
		True condition	
	Total population	**Condition positive**	**Condition negative**
Predicted Condition	**Predicted condition positive**	**True positive**	**False positive**
	Predicted condition negative	**False negative**	**True negative**

Figure 2.16: Confusion Matrix [27].

For multi-class classification, the confusion matrix is expanded to include all classes. An example of a multi-class confusion matrix is shown in Figure 2.17. The numbers in each grid space indicate the number of predictions made corresponding to the actual class and predicted class. The downward diagonal is where the actual and predicted classes match, and thus ideally should contain the highest values.

The accuracy of a classification model is calculated based upon the TP, TN, FP, and FN values. The equation is shown below:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.23}$$



	alt.atheism	comp.graphics	comp.os.ms-windows.misc	comp.sys.ibm.pc.hardware	comp.sys.mac.hardware	comp.windows.x	misc.forsale	rec.autos	rec.motorcycles	rec.sport.baseball	rec.sport.hockey	sci.crypt	sci.electronics	sci.med	sci.space	soc.religion.christian	talk.politics.guns	talk.politics.mideast	talk.politics.misc	talk.religion.misc
alt.atheism	0.42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.08	0.17	0.00	0.00	0.25
comp.graphics	0.00	0.72	0.00	0.07	0.03	0.07	0.00	0.00	0.03	0.00	0.03	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00
comp.os.ms-windows.misc	0.00	0.07	0.79	0.10	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
comp.sys.ibm.pc.hardware	0.00	0.03	0.16	0.58	0.10	0.03	0.03	0.03	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00
comp.sys.mac.hardware	0.00	0.00	0.00	0.00	0.84	0.04	0.00	0.04	0.04	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00
comp.windows.x	0.00	0.00	0.08	0.00	0.00	0.88	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
misc.forsale	0.00	0.00	0.00	0.00	0.03	0.00	0.69	0.09	0.11	0.06	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00
rec.autos	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.90	0.04	0.01	0.01	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.00
rec.motorcycles	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00
rec.sport.baseball	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.98	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
rec.sport.hockey	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
sci.crypt	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
sci.electronics	0.00	0.04	0.00	0.04	0.04	0.00	0.02	0.07	0.02	0.02	0.00	0.02	0.64	0.04	0.02	0.00	0.00	0.00	0.00	0.00
sci.med	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.02	0.02	0.02	0.00	0.00	0.02	0.84	0.05	0.00	0.00	0.00	0.00	0.00
sci.space	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.89	0.00	0.00	0.00	0.00	0.05
soc.religion.christian	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.15	0.00	0.00	0.00	0.05	0.00	0.65	0.00	0.00	0.00	0.10
talk.politics.guns	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.00	0.00	0.00
talk.politics.mideast	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.93	0.00	0.00
talk.politics.misc	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.33	0.00	0.56	0.06
talk.religion.misc	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.96

Figure 2.17: Confusion Matrix [28].

Precision is a metric which can be calculated from the confusion matrix. Precision defines - when the classifier predicts the positive class - how often it does this correctly. In the case of multi-class classification, precision defines how often the classifier is correct when it has predicted a specific class. The equation is as follows [26]:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.24}$$

Recall is yet another metric which can be determined from the confusion matrix. Recall defines when the class is actually positive, how often the classifier predicts this correctly [26]. In the case of multi-class classification, Recall is how often a specific class is predicted correctly. The equation is shown below:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.25}$$

F-1 score is the harmonic mean of precision and recall. It takes both precision and recall into account. A model with a higher F-1 score is performing better than those with lower F-1 scores [26]. The equation is as follows:

$$\text{F-1} = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.26}$$

Chapter 3

Detection of Terrain Changes and Terramechanics Parameter Estimation

The previous chapters presented the background information needed to develop a method to use the data collected by the wheel to estimate terramechanics parameters and classify terrain using the Random Forest model. The estimation of grain size also builds upon the previous terramchanics information, though, some more information is needed and will be covered in Chapter 5. This chapter discusses work that was done in verifying that changes in terrain could be detected by the wheel instruments. Two test run types are presented that were used for this verification.

3.1 Methodology

With the intention of estimating terramechanics terrain parameters using the *Barefoot Rover* wheel, it was first necessary to confirm that the instruments on the wheel and the test setup itself could detect changes or transitions in terrain. If it could not, then attempting to obtain terramechanics parameters would hold no value. The purpose of a wheel of this type is two-fold; to carry and move a load (a planetary rover chassis) and to collect data about the planetary terrain surface that can be meaningfully interpreted by on-board post-processing. If the instruments on the wheel could not detect transitions and changes in terrain then the second portion of that purpose could not be accomplished.

Two test runs were designed to verify the wheel could detect changes in terrain. These test runs were Free Slip and Duricrust test runs. Duricrust will be discussed in its respective section.

3.2 Free Slip Test Runs

Prior to carrying out the Free Slip test runs, all previous test runs that induced wheel slip accomplished this by driving two aluminum stakes into the terrain to specific depths and mounting this to the rig. A restraining force was thus induced on the rig which caused the wheel to slip as the motor attempted to overcome this force. This method was adequate for testing the accuracy of the string potentiometer and observing major differences in pressure pad data, but is not representative of the actual conditions of a planetary rover. An external resistive force will likely never be experienced by a rover apart from encountering a rock. Wheel slip will be a result of changing terrain, cemented to loose sand, as an example. This change will also be smaller in magnitude and require more sensitive instruments than detection of high slip where the wheel is already experiencing a substantial slip ratio (> 0.6). Free Slip runs aimed to verify the detection of these natural transitions to a higher fidelity.

To set up this compaction transition, the first half of a MMS 2mm trough was highly compacted and the second half was made to be loose and fluffy. This corresponds to a terrain transition from higher cohesion to lower cohesion. Figure 3.1 shows the test setup for a Free Slip run prior to running the data collection run. Note that the transition is not visible to the naked eye, of which optical navigation systems rely on.

The point of interest for data analysis in this case was at the location of material transition. The two most important pieces of data were the slip ratio and wheel-terrain contact area. Figure 3.2 shows the slip ratio data for all four Free Slip data collection runs. The vertical black bar indicates the time at which the wheel passed over the location of material transition.

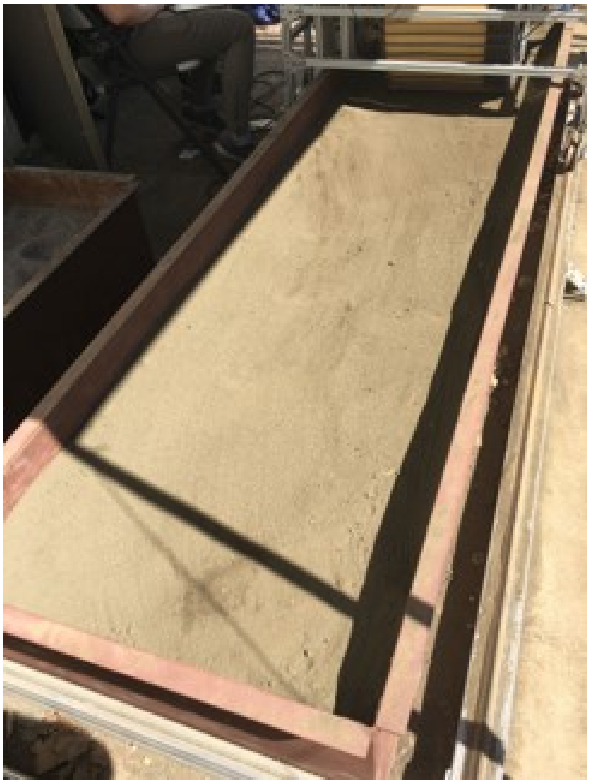


Figure 3.1: Free Slip test run setup.

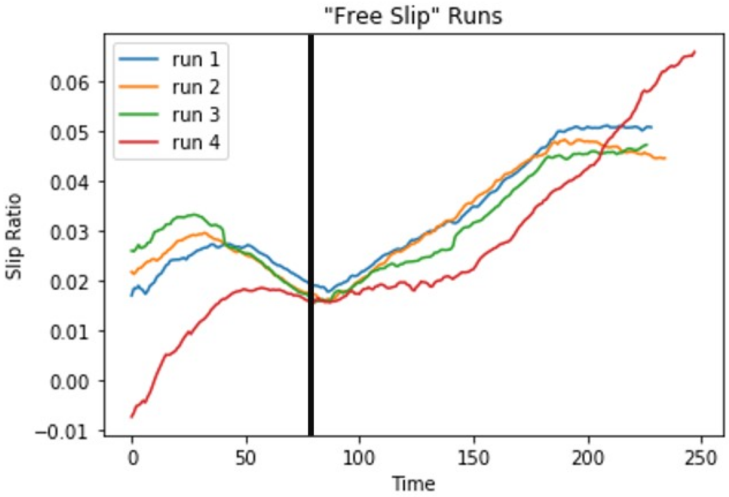


Figure 3.2: Free Slip test run slip ratio data for all four test runs of this type.

Figure 3.2 shows that at the location of material transition, slip is detected to increase, as was expected. The data is plotted over time indicating the length of time the wheel rolled over the simulant. An increase in slip in approximately the first 80 seconds of the test run is observed which is attributed to the wheel starting from rest, initially experiencing some slip, and then settling into steady state. Figure 3.3 shows a normal MMS 2mm flat no-slip run as a means of comparing data trends and verifying that the trend observed in the Free Slip data is not one that occurs naturally in other data.

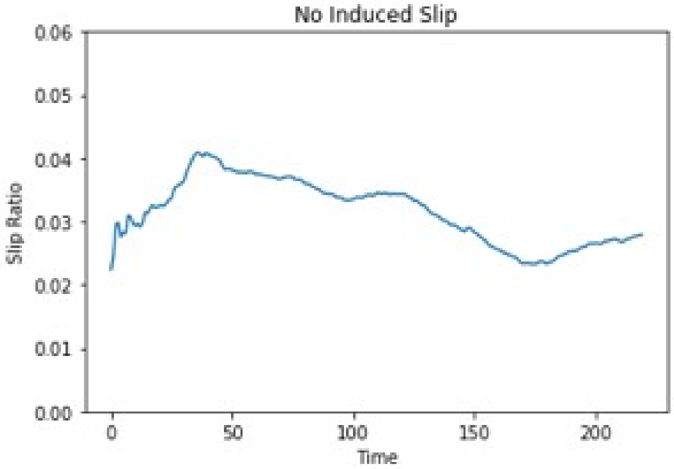


Figure 3.3: Comparison of Slip ratio for a normal flat run

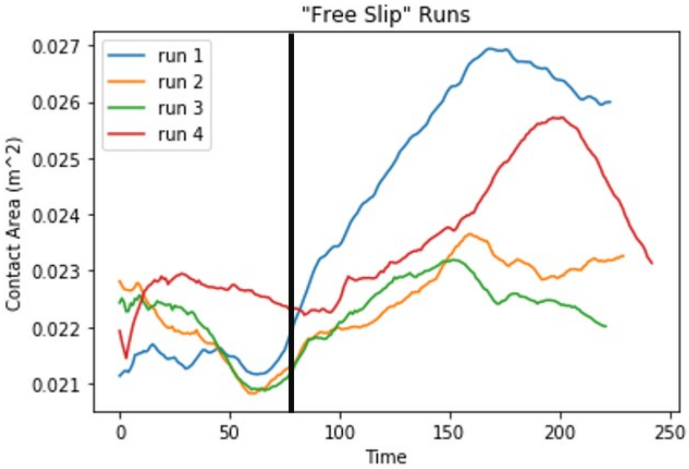


Figure 3.4: Free Slip contact area data.

The contact area results are shown in Figure 3.4. One can observe an increase in contact area seen by the pressure pad, as was expected. The point of inflection is noted to be slightly off from the time at which the wheel crossed over the terrain transition, which is attributed to the noise levels in the pressure pad data.

The Free Slip test runs showed that the instruments on the wheel and the setup could detect transitions in terrain that resulted in low magnitude changes in slip and contact area.

3.3 Duricrust Detection

Duricrust is a cemented or highly cohesive terrain layer either on the surface or near to the surface. This cemented layer is primarily induced by water soluble minerals that remain after evaporation, one of which is often salt. Duricrust can be dangerous to planetary rovers, as was observed in the case of MER *Spirit* which broke through a layer of duricrust and became immobilized in the low cohesive sand below, effectively ending the mission [29]. It's detection is important for improving the safety and traverse capabilities of a rover [30]. Because of this danger, a duricrust data collection run was designed for the *Barefoot Rover* wheel to examine the wheel's ability to detect this type of terrain. A trough of MMS 2mm sand was hydrated and left to dry over one week. A hard layer of crusted terrain formed over top loose 2mm sand. Figure 3.5 shows the trough as it was before carrying out the data collection run. After the test run, the wheel grouser marks were barely visible on the duricrust because of its level of cementation.

Only one duricrust test run was performed because of the time needed for getting the sand in the proper condition and the limited availability of Mars simulant. The results of this singular data collection run, however, were promising. As is shown in Figure 3.6 and in Figure 3.7, there is a decrease in slip ratio and wheel-terrain contact area versus normal non-duricrust MMS 2mm data. The reduction in slip and contact area is the expected result and one which indicates that the instruments on the wheel and such a setup can detect duricrusted terrain. Lower slip was expected because duricrust is a hard cemented layer of sand, and therefore the wheel rolls on it as if it were rolling on pavement. The same applies to sinkage, the wheel simply cannot sink as much on duricrust because of its cemented nature.

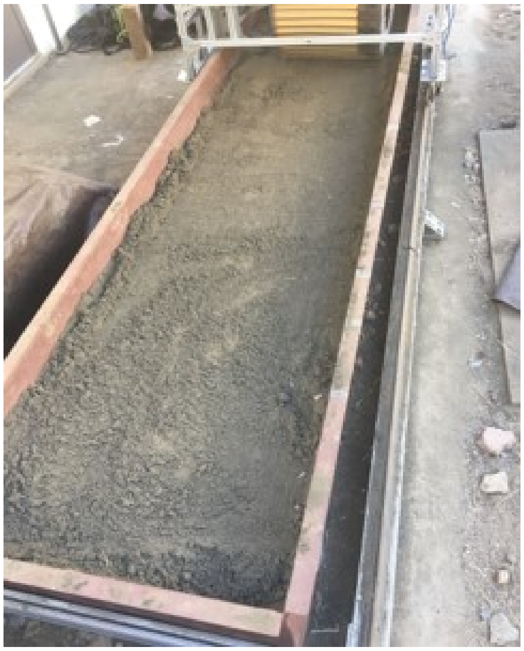


Figure 3.5: Duricrusted MMS 2mm sand.

Both the Free Slip and Duricrust test runs confirmed that the instruments on the wheel and using a rover wheel in such a manner is possible and able to detect transitions in terrain to a level that will be useful for rover driving and for the scientific community. Data like this accompanied by terrain parameter estimations can be used to flag areas of interest for further scientific study or to avoid driving further into the forthcoming terrain. Indications of cementation levels (which indicate certain weathering or geologic processes) can help geologists understand formation processes on a planetary surface [31].

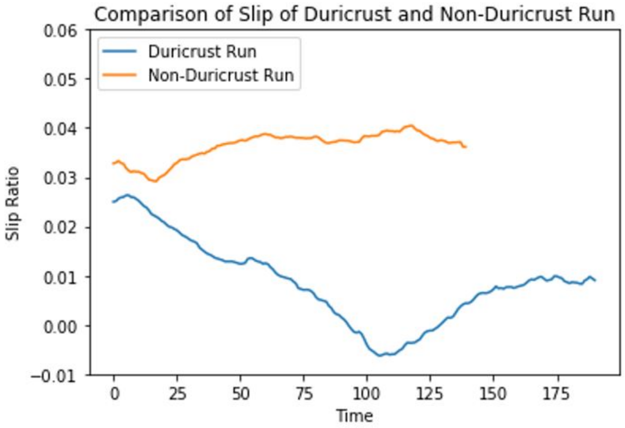


Figure 3.6: Slip ratio data from the duricrust test run.

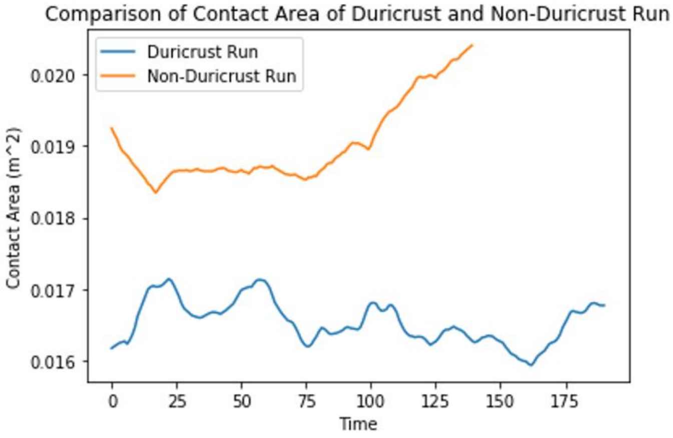


Figure 3.7: Contact area data from the duricrust test run.

It is important to note that for the detection of duricrust, differences in slip and contact area are not enough alone. The wheel could also be detecting the existence of bedrock underneath the sandy surface. The existence of bedrock can be checked by using cameras on the rover to examine if bedrock has been revealed in the rover wheel tracks. Duricrust detection will require additional instruments for verification and the instrumented wheel data can serve as the preliminary flag that the rover may be driving into duricrust.

3.4 Parameter Estimation Method and Results

The previous sections presented the verification that the instruments on the wheel and that the *Barefoot Rover* test setup could be used to detect changes in terrain. This sections discuss the development of the process to which the terramechanics parameters can be estimated from collected data and the results of this process. The first section describes the method for calibrating the pressure pad to physical load values and the second section describes the optimization method for estimating terrain parameters. The second section presents the results.

3.4.1 Methodology

Pressure Pad Calibration

As noted previously, the pressure pad mounted around the outer surface of the wheel shown in Figure 3.8 does not produce physical load readings unless otherwise calibrated. Thus, calibration was necessary to input wheel contact pressure into the classical terramechanics models for parameter estimation. The pressure pad is a grid of 96 × 20 pixels, or taxels, which sense applied load and output a data value. Each quadrant of the pressure pas is 24 × 20 taxels in size and each taxel has an individual calibration curve from raw output to physical value. It was not known whether a linear or non-linear calibration curve would be found for each taxel, or whether all taxels would follow a linear or non-linear calibration. The reasoning behind this concern was that the pressure pad had been used for quite some time and had likely experienced degradation. Some taxels may be now insensitive or completely unresponsive. Additionally, the pressure pad layers may become permanently deformed after prolonged use, resulting in various base readings for each taxel. Thus, each taxel will have its own calibration curve.

The procedure to calibrate the pressure pad was developed to produce the calibration curves for the individual taxel. There were two iterations of this procedure which are discussed below. Figure 3.9 shows the wheel resting on the plank used for calibration. Both procedural iterations used this plank.

Iteration 1: The wheel was rolled down the plank while collecting pressure pad data with increasing weights mounted internally. These were done in order of 10, 20, and 30 lb incre-

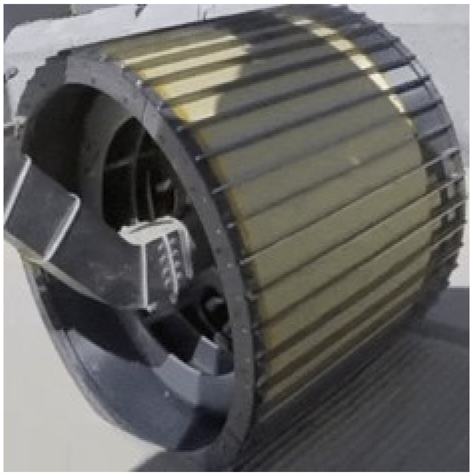


Figure 3.8: The *Barefoot Rover* wheel with pressure pad wrapped around the wheel and under grousers.

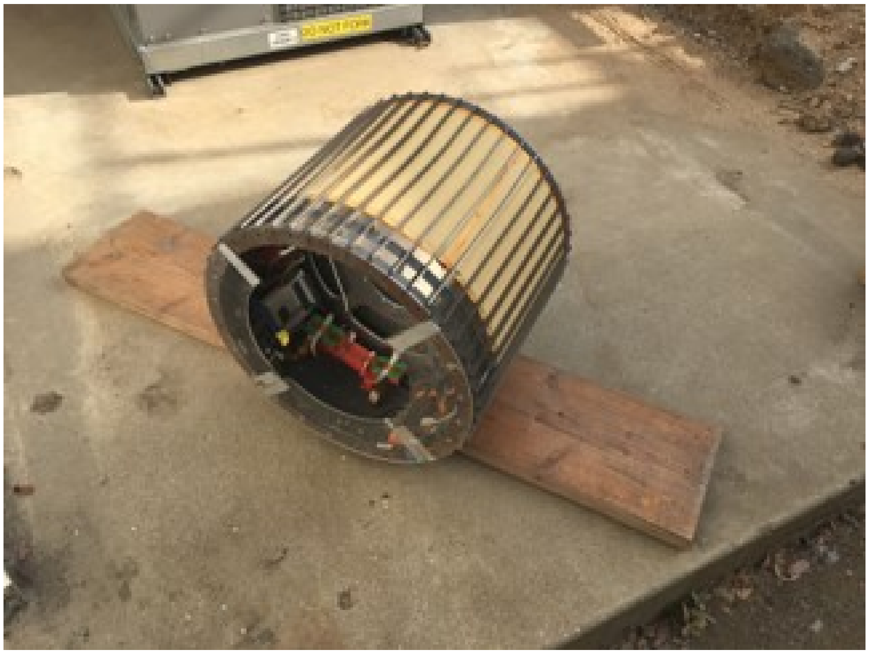


Figure 3.9: The *Barefoot Rover* wheel resting on calibration plank.

ments. The raw data output and known total loading was compared to form the calibration curves. To start, it was assumed that each taxel would show a linear calibration curve. Upon analyzing the slope coefficients of the linear trends an interesting anomaly was observed shown in Figure 3.10. There are noticeable horizontal regions where the slope coefficient was near or at zero, which indicated an incorrect fit. It was determined the horizontal regions of poor fit were due to the width of the plank, which was less than the width of the pressure pad, and so created anomalous readings at the edge of the plank. The second iteration of the calibration runs aimed to fix this.

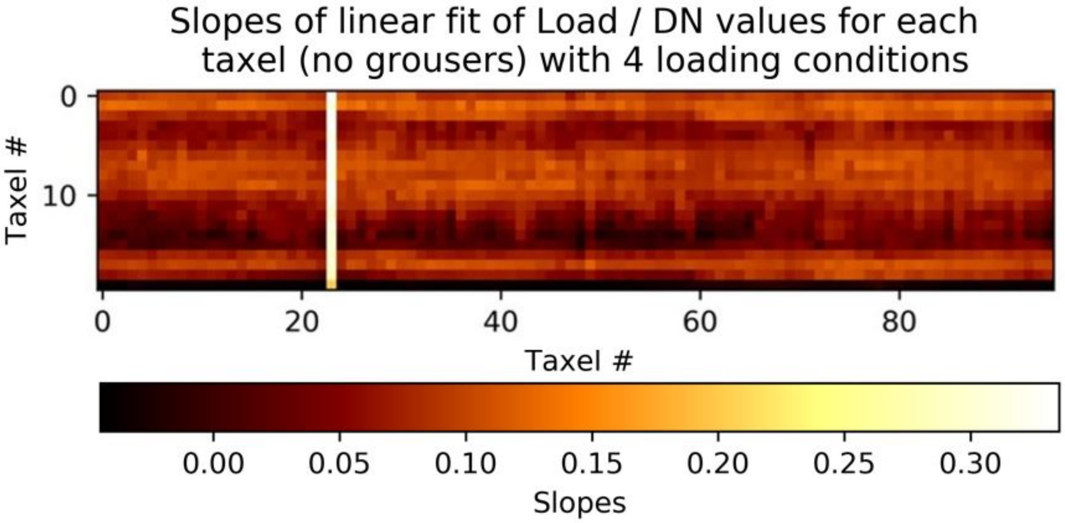


Figure 3.10: Slope coefficients for the individual pressure pad taxels.

Iteration 2: The wheel was set up so that one edge of the pad aligned with an edge of the plank. The wheel was then rolled, while collecting pressure pad data, down the plank. Upon reaching the end of the plank the wheel was reset to the starting angle, offset so that the opposite edge was aligned with an edge of the plank, whereupon the wheel was then rolled back up the plank while collecting pressure pad data. In this way, the anomalous edge zones were blanketed by good data for all the pressure pad taxels. The resulting slope coefficients are shown in Figure 3.11, where one can see that the poor data regions no longer exist.

It was then necessary to examine the R2 (or R squared) score for each taxel linear fit. This score indicates how well the collected output data to increasing mounted load follows a linear trend, as shown in Figure 3.12

Many of the pressure pad taxels follow a linear calibration curve well. There are four noticeable vertically oriented regions of poor fitting. These are where the four pressure pads overlap to form the single large pad, which created insensitive regions. These are not consid-

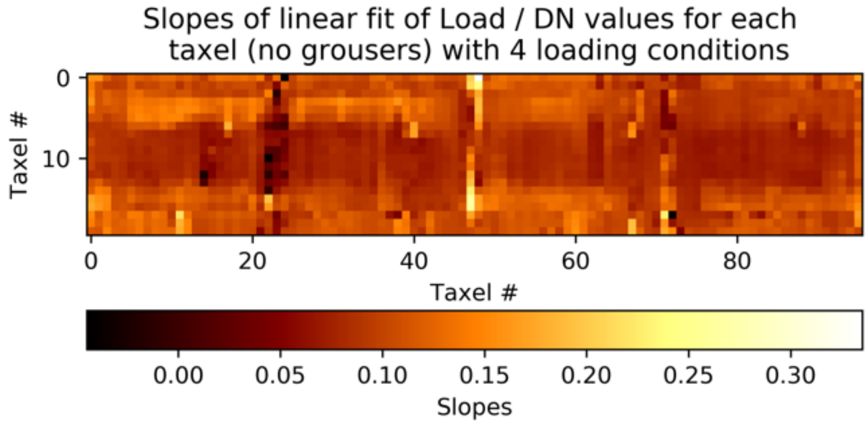


Figure 3.11: Slope coefficients for the individual pressure pad taxels.

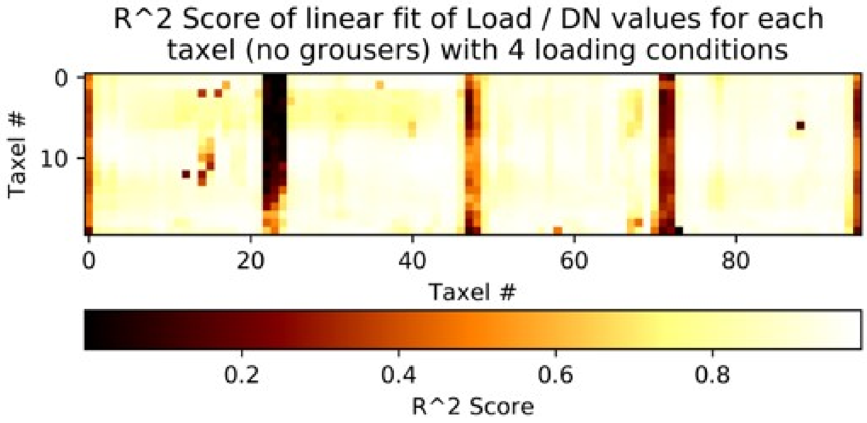


Figure 3.12: R2 Score for every pressure pad taxel linear calibration fit.

ered in any analysis of the data and any time bins where wheel contact occurs in these known regions are removed in post-processing. There are then singular or small groups of taxels which do not follow a linear calibration fit. These taxels are considered to be insensitive or poorly performing taxels and are removed by post-processing before data analysis. What remains for further analysis is approximately 87% of the pressure pad taxels and their output data for each data collection run.

The calibration was expected to produce physical pressure values which would indicate a total load on the pressure pad close to that of the wheel itself, which weighs 30 kg, or has a mass of 294.3 N. This was verified by using the pressure pad data from a flat no-slip MMS 2mm run and applying the calibration to each taxel and summing their output. Load felt by the wheel was estimated for each time bin of the data collection run and so an instantaneous, not bulk, load is produced. The resulting loads are shown Figure 3.13. There is a maximum error of approximately a 10% compared to the expected 294.3 N load. The rise at the beginning of the load curve is attributed to the wheel starting its roll and having not yet reached steady state. The fall may be attributed to the rig taking off some load from the wheel, though this was never shown to happen. The loading estimation was repeated for several more data collection runs with similar results. Total instantaneous pressure is determined by dividing the load by contact area as seen by the pressure pad in the same time bin.

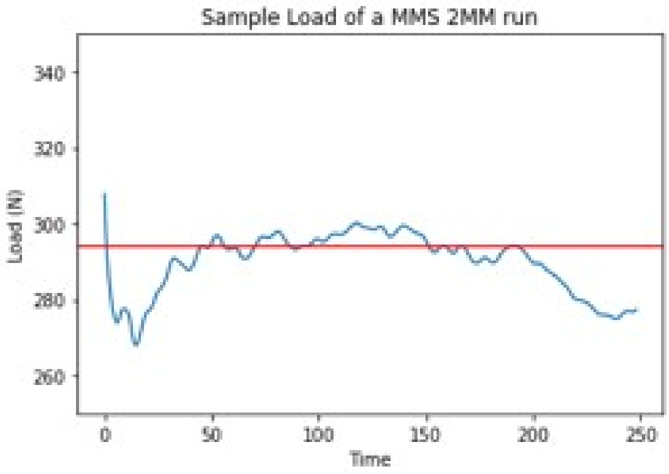


Figure 3.13: The instantaneous load felt by the pressure pad at each time bin during a data collection run. The red line shows expected wheel loading.

Like most instruments, the pressure pad degrades over time. Taxels on the pad either begin to malfunction or become entirely insensitive. The fabric and foil-like layers of the pad can also become permanently deformed after long periods of use, resulting in incorrect base value data readings. All of this means that the pressure pad should be calibrated once again after each data collection campaign. The data discussed in this thesis was collected during one data collection campaign, or, approximately four months of use.

Terrain Parameter Optimization Method

With the development of contact pressure estimation completed, all required inputs for estimating terrain parameters using the wheel-terrain interface shear stress distribution equation could be obtained from the wheel data. The equation is again shown below:

$$\tau = [c + \sigma \tan \phi][1 - \exp \frac{-j}{K}] \tag{3.1}$$

Cohesion c, angle of friction ϕ, and shear modulus K are to be estimated. The known variables are then shear stress τ, contact pressure σ, and slip deformation j. The shear stress is obtained using the following method as described by Sullivan et al. [1]:

$$\tau = \frac{F_s}{A} \tag{3.2}$$

Where A is the wheel contact are and is a known value and Fs is the shear force and is related to motor torque:

$$\text{torque} = rF_s \tag{3.3}$$

Where r is the radius of the wheel and is a known value. Motor torque can be determined by collected motor current data and motor efficiency parameters:

$$\text{torque} = I\eta\kappa\zeta \tag{3.4}$$

Where I is motor current, η is transmission efficiency, κ is torque constant, and ζ is the gear ratio. All of these values are known for the motor powering the *Barefoot Rover* wheel. Following the method outlined in equations 3.2 - 3.4, shear stress is determined. Contact pressure or, sigma, is known from collected pressure pad data. Slip deformation j is determined through the following:

$$j = r(\theta + i \sin \theta - \sin \theta) \tag{3.5}$$

Where θ is the contact arc and is a known value measured by the pressure pad and i is the slip ratio and is a known value measured by the string potentiometer.

To then estimate terramechanics terrain parameters, focus is again turned to equation 3.1. There are three unknowns and three known values. The shear deformation modulus K is estimated purely because it is not known, however, it holds little physical meaning for terrain and is not discussed further.

To estimate three unknowns simultaneously with one equation, the built-in sklearn curve-fit optimization function was used in Python. The optimization is done over moving windows of the data for an individual data collection run. The best performance window was determined to be between two data points, driven by the noisy nature of the wheel data. The optimization bounds are set to be outside of known ground truth cohesion and angle of friction values for the particular simulant terrain. In this way, the bounds are not "driving" the solution, the data is. It was necessary to apply a post-processing running mean smoothing function to the data prior to the parameter estimation.

Curve-Fit Function

Shear stress is determined at each time step in the data using equations 3.2 - 3.4. If a data collection run on a simulant type is 300 time steps, then shear stress is a one dimensional time series of 300 data points. Knowing that shear stress can also be represented by equation 3.1, this time series can be plotted and represents a function where information about the three terrain parameters c, ϕ and K for that terrain is captured. This information is supported by the known values τ, σ and j in equation 3.1. Fitting a curve to the plotted data allows for the calculation of the terrain parameters that best fit the plotted time series, which can be achieved with the curve-fit function [32]. Because there are three variables to identify, multivariate interpolation is needed, and the least squares method was selected due to the non-linear nature of the function.

Curve-fit reads in a function, in this case defined as equation 3.1, fits a curve to the provided y-values and returns the best fit values for the provided function [32]. The curve-fit function in this case does not fit the entire time series because the raw data contained too much noise for curve-fit to estimate a proper fit. To deal with the high noise, the data is fed to curve-fit in windows of three points, which then produces "apparent" rather than bulk terrain parameter estimates.

3.4.2 Terrain Parameter Estimation

The previous section discussed the work done, and method development in order to proceed to estimating terramechanics terrain parameters based upon data collected by the *Barefoot Rover* wheel and its accompanying instruments. This section presents the results of terramechanics terrain parameter estimation for normal flat data collection runs on MMS 2mm and GRC-01 simulants, Free Slip on MMS 2mm simulant, and the duricrusted MMS 2mm simulant.

3.4.3 Nominal flat MMS 2mm and GRC-01

The two simulants discussed in this section were chosen for their greatest dissimilarity in ground truth cohesion and angle of friction values; a result of their individual characteristics of soil makeup and grain size.

It is important to note that these parameter estimations are “apparent” cohesion and angle of friction values. The resulting cohesion and angle of friction values at each time step are the product of the instantaneous wheel-terrain interaction at that window. These apparent terrain parameters are then compared to the bulk ground truth values, with the expectation that the apparent values throughout a data collection run will plot within the bulk ground truth range bands with little error. Apparent values, rather than bulk estimates, contain greater resolution in terms of information about changes occurring in the terrain as the rover drives. This greater resolution is useful for both rover drives and planetary geologists, and for the development of autonomous rover systems. Figure 3.14 shows the apparent cohesion values of seven MMS 2mm flat data collection runs. The data from seven test runs is simply appended together to form one larger time series. No averaging or other processing takes place. The horizontal red bands indicate the ground truth bands for cohesion of the MMS 2mm simulant [33].

From Figure 3.14, it is shown that approximately 90% of the time, the optimization estimation method produces values of cohesion within the known ground truth range. The 10% of time where the results fall outside of this band only do so with approximately 8% error. This small error indicates that the data collected and the method used for analysis is sound and results in useful terrain parameter values.

It is expected that the trend observed in cohesion results should be similar in angle of fric-

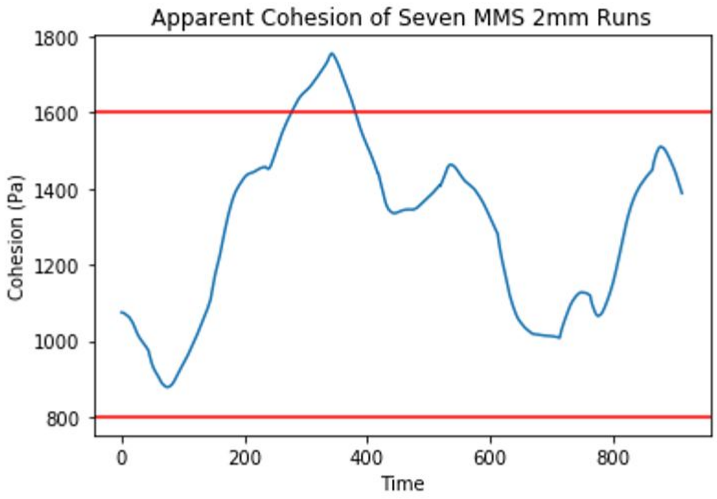


Figure 3.14: The resulting cohesion values of seven flat MMS 2mm data collection runs.

tion results. There should be this correlation because, as discussed previously, both cohesion and angle of friction are shear strength components of terrain. When the shear strength of a terrain is reduced by a decrease in cementation or smoothing of grains both components of shear strength will decrease. Figure 3.15 shows the angle of friction estimations for the same seven runs shown in Figure 3.14. The expected correlation of trend is observed.

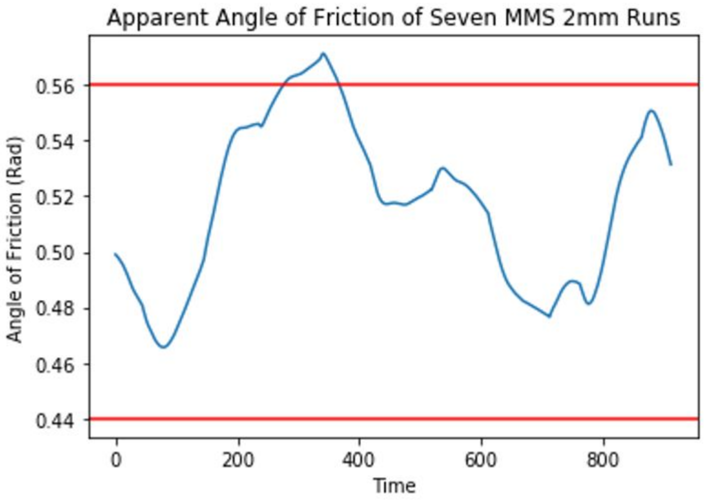


Figure 3.15: The resulting angle of friction values of seven flat MMS 2mm data collection runs.

The same cohesion and angle of friction estimates were made from data collected on the GRC-01 simulant. The data collection runs were of the same flat type and no changes were made to the wheel and rig setup. Figure 3.16 and Figure 3.17 show the resulting terrain parameter estimates. The horizontal red lines again indicate the known ground truth range for each terrain parameter for GRC-01 [34]. The same correlated trend between the two parameters is observed.

The results shown in this section showcase how the wheel data and estimation method produces reasonable cohesion and angle of friction estimates that are, 90% of the time, within known ground truth ranges. It also highlights that the terrain parameter estimate values change from simulant to simulant as a result of the collected data, indicating that the instrumented wheel is able to detect different terrains. The subsequent sections present terrain parameter estimation results on the previously discussed Free Slip and Duricrust data collection runs.

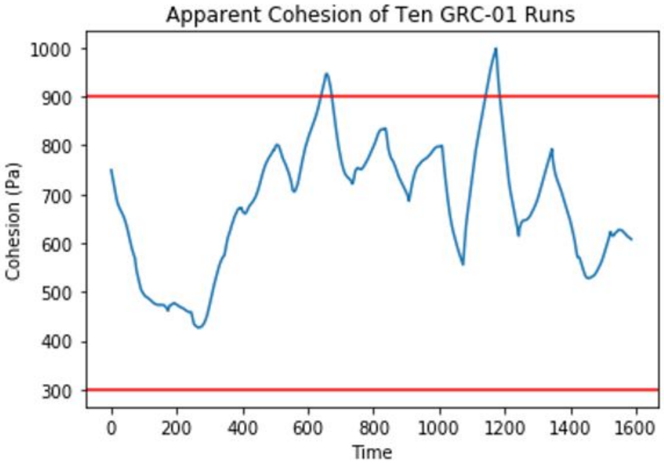


Figure 3.16: The resulting cohesion values of ten flat GRC-01 data collection runs.

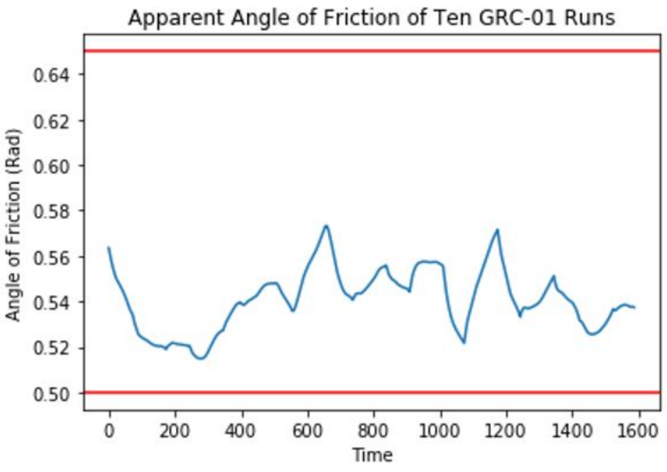


Figure 3.17: The resulting angle of friction values of ten flat GRC-01 data collection runs.

3.4.4 Free Slip

When the Free Slip data was analyzed to confirm that the wheel and accompanying instruments could detect slight transitions in terrain, the natural next step was to obtain terrain parameter estimations from the same data. With decreasing compaction, one would expect, and terramechanics models dictate, that cohesion and angle of friction would decrease as well as a result of the decreasing shear strength of the terrain. Figure 3.18 shows the cohesion and angle of friction estimates for a Free Slip Run, with similar results observed for the others.

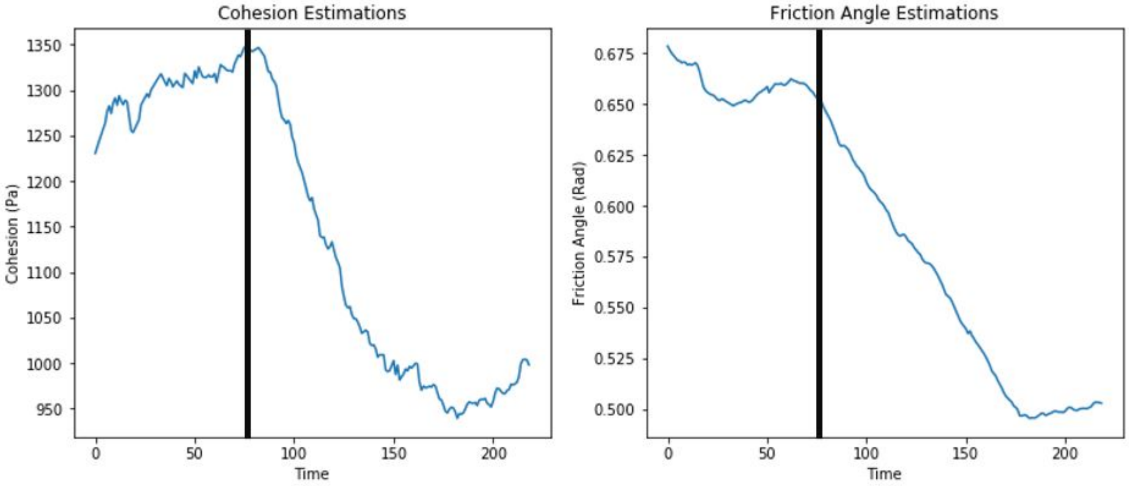


Figure 3.18: The resulting terrain parameter estimates for a Free Slip data collection run.

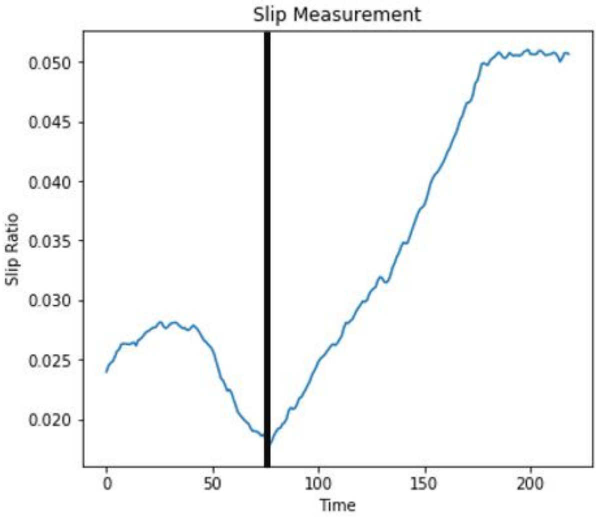


Figure 3.19: The slip ratio data for the data collection run used for Figure 3.18.

The black vertical line indicates the time at which the wheel rolled through the transition region to less compacted terrain. For reference, the slip ratio data collected on the same data collection run in Figure 3.18 is shown in Figure 3.19.

What Figure 3.18 shows is that the terrain parameter estimates reasonably represent a compaction transition within the same terrain and within an expected value range. Not only are these terrain estimates made using slip ratio and contact area data, but also are made with shear stress from motor current data and calibrated pressure pad data.

3.4.5 Duricrust

The terrain parameter estimation for the duricrust data was contrary to that of the other parameter estimations done. It was expected, or rather hoped, that the estimation method would produce (as a result of the collected data) terrain parameter values *outside and greater* than the known ground truth ranges for the MMS 2mm simulant (the same simulant was used for the duricrust). Duricrusted terrain is highly cemented and therefore should have a greater cohesion and angle of friction than nominal MMS 2mm simulant. Ground truth values for duricrust do not exist and as a result, it was not known what values in particular to expect, but it was expected that the terrain parameter estimations should be greater than the nominal ranges and indicate a change in terrain. Figure 3.20 and Figure 3.21 show the resulting terrain parameter estimations. The horizontal red lines indicate the known ground truth values for the particular parameter of normal non-duricrusted MMS 2mm simulant.

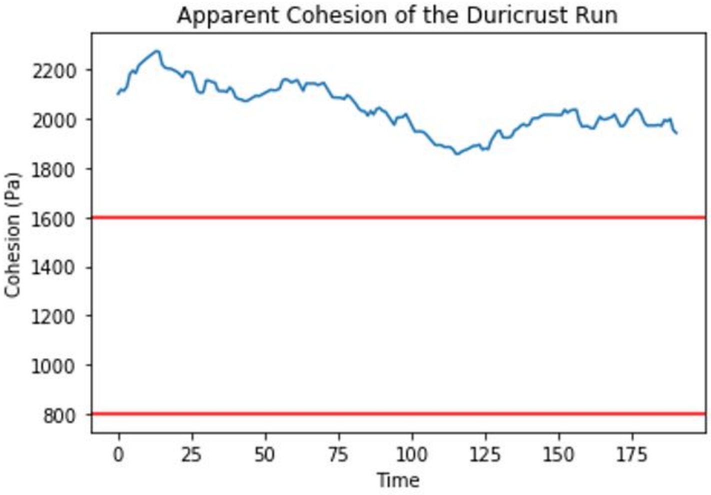


Figure 3.20: The cohesion estimates for the duricrust data.

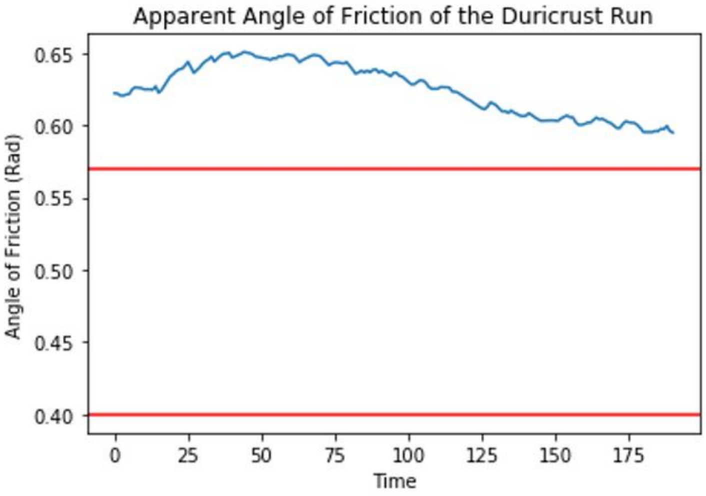


Figure 3.21: The angle of friction estimates for the duricrust data.

As Figure 3.20 and Figure 3.21 show, the resulting terrain parameter values are indeed outside and greater than what one would expect for a normal MMS 2mm simulant. This result indicates that this type of instrumented wheel can be used for the detection of anomalous terrains, or terrains which exhibit characteristics outside of expectation. Not only does this technology and methodology have applications for the unsupervised in-situ machine learning clustering of terrain encountered on Mars, but also in rover safety and as a feedback loop in autonomous rover driving systems for the flagging or warning of unsafe or previously unencountered terrain types. Geologists and planetary scientists will find great use in apparent cohesion and angle of friction knowledge for terrain on Mars and other planetary bodies because this increases the resolution at which information about the terrain is known and can help them understand formation processes.

Some further uses for estimating cohesion and angle of friction values on Mars and other planetary bodies include; the development of more representative Mars simulants using the terrain parameter estimations as a baseline for the creation of these new simulants. All wheel data will play a role in this aspect so that simulants here on Earth may interact with tested rover wheels more alike how they will interact with each other on the target celestial body. Additionally, the collected data and terrain parameter estimates can be used as a ground truth for the estimation of terrain makeup using orbital thermal inertia data [2]. For deployment and any uses in machine learning, it is recommended that unsupervised methods be pursued, as these are most suited to categorizing unseen and new information, such as terrain previously undisturbed for thousands of years. Additionally, for deployment, a method to estimate wheel slip from pressure pad data will need to be developed. Using an external string potentiometer is not feasible. As discussed previously, there are distinct differences between low and high slip pressure pad data which may be useful for the prediction of slip ratio using a machine learning regressor.

Chapter 4

Terrain Classification

This chapter outlines the methodology and results of the classification of terrain types using data collected by the *Barefoot Rover* wheel. The following will be discussed:

- The data, pre-processing and feature selection
- Cross-fold validation and hyperparameter tuning
- Results

4.1 Data, Pre-Processing and Feature Selection

The data used to create the feature set for classification are time series of sensor data from individual data collection runs. For a given terrain, these time series are combined to form a larger time series that includes every example of data on that terrain that has been collected (in the same data collection run configuration, no slip and flat-level terrain). For a given sensor, the data for each terrain is appended to form a time series of different terrain data for this sensor. For use in the Machine Learning model, this must be combined with the relevant sensor data and engineered features and a corresponding class label for each example. Figure 4.1 shows an example of data for a single feature, before pre-processing.

One can see in Figure 4.1 that outliers exist in the data which represent most likely an anomaly in the function of the sensor. With reference to Equation 2.14, in order for slip ratio to be negative, $\frac{V}{rw}$ must be greater than 1. This is only possible if V is greater than rw.

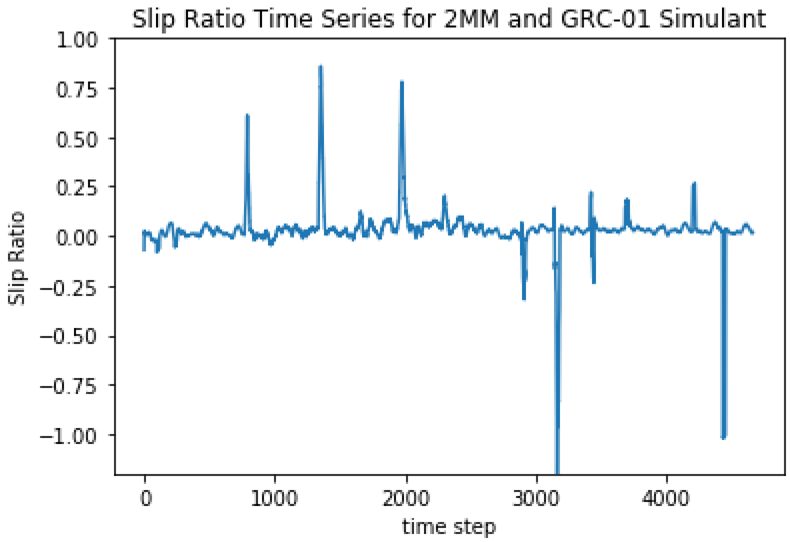


Figure 4.1: Time series of slip ratio data from two terrains: 2MM and GRC-01 simulant. This is an example for a single feature (pre-filtering).

It is impossible for forward velocity V to be greater than the constant value of rw unless an external pushing force in the positive direction of forward velocity is applied. This force was not applied during any data collection run, and therefore, the negative slip values represent an inconsistency in the sensor readings. The positive outliers are equally as anomalous given that they suggest the wheel, in approximately five time steps of 9.2 milliseconds each, experiences an increase in slip ratio from approximately 0.05 to above 0.5. Meaning, that in approximately 0.046 seconds, the wheel forward velocity decreased by about 70% from approximately 0.04 m/s to 0.0129 m/s and then back to 0.04 m/s in another 0.046 seconds. Any such occurrence would have been noticed during testing, and is unrealistic given the inertia of a 30 kg wheel, and as such these values are anomalies of the sensor and should also be filtered out. Figure 4.2 shows the result of the filtering.

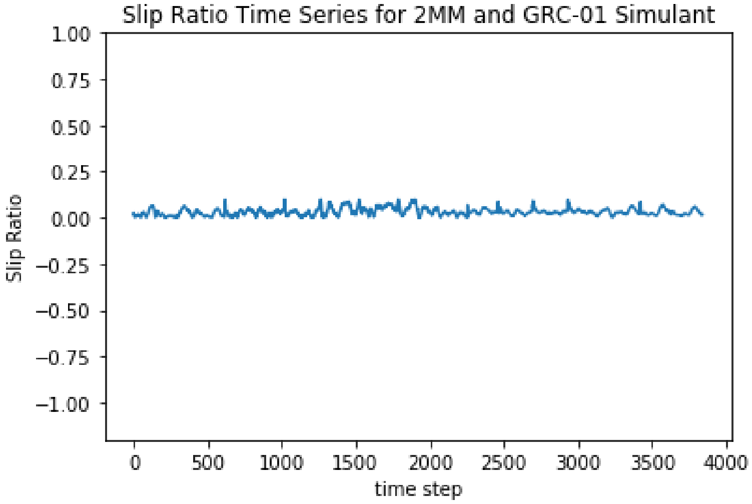


Figure 4.2: Time series of slip ratio data from two terrains: 2MM and GRC-01 simulant. The time series is example data for input into the machine learning algorithm for training and testing (post-filtering).

The three sensors/engineered features are as follows:

- The pressure pad data is used as a feature. The pad data is calibrated to physical pressure values as discussed in Section 3.4. The contact pressure feature is thus engineered from raw data.

- The potentiometer sensor data is used as a feature. This sensor outputs actual distance travelled, which is compared to the distance the wheel believes it has travelled based on run time and angular velocity. Slip ratio is then calculated, and is used as an engineered feature.

- The motor current data is used to engineer the motor torque feature using the method outlined in Section 3.4.

These three features were selected given their important in classical terramechanics models. Contact pressure is related to sinkage, and these two parameters are highly influenced by terrain cohesion and angle of friction [4]. Slip is also a crucial indication of terrain cohesion and angle of friction, which are the two terrain parameters which most divide terrains from each other. These two features combined (pressure and slip) offer significant insight into the behaviour of terrain. The motor torque is a variable needed to calculate shear stress τ at the wheel-terrain interface as seen in Section 2.2. The shear stress is influenced once again by cohesion and angle of friction. Slip and shear stress offer insight into the shearing performance of the wheel-terrain interaction, while pressure offers insight into the vertical loading performance of the wheel-terrain interaction. Further features were to be added if found to be necessary, but this was not the case.

4.1.1 K-Fold Cross Validation and Hyperparameter Tuning

Cross validation is used in Machine Learning to determine the generality of a machine learning model, or, its robustness to unseen data. K-fold cross validation splits up the dataset, *K* number of times, into training and testing sets (sometimes referred to as validation sets) that are different for each *K*. In this way, it can be seen what the average performance of the model is and testing can be done on many different portions of data selected from the overall dataset. Figure 4.3 shows an example of how the dataset is split for each *K*.

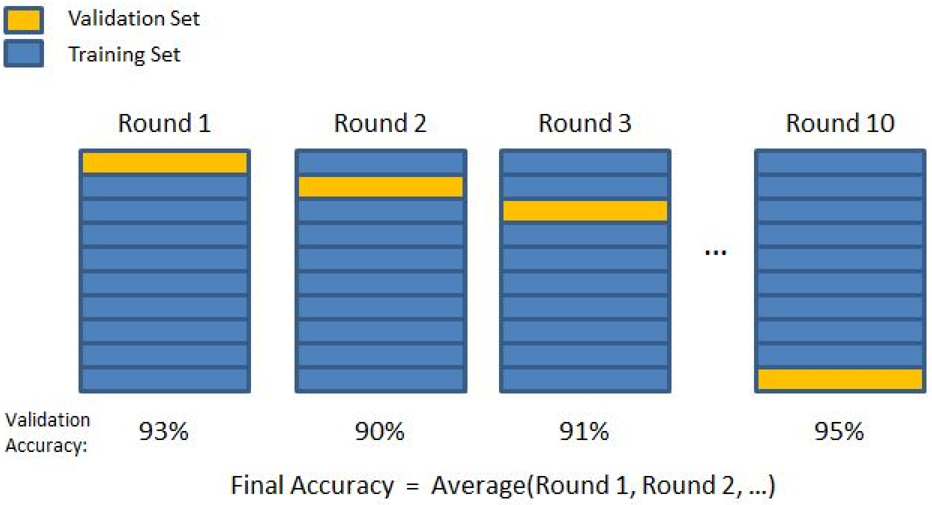


Figure 4.3: An example of K-fold cross validation [35]

In Figure 4.3, *K* is equal to ten, and one can see that the accuracy of the model changes when varying the training and test/validation sets used. For each K, a different 10% of the data is selected for testing/validation, but never the same data, and there is no overlap. The accuracies are averaged to obtain a more holistic perspective on the model's accuracy.

When multi-class classification is being done, cross validation becomes even more important [35]. In general, values of 5 or 10 for K are used, depending on the size of the dataset, as these values of K historically do not suffer from high variance of results nor high bias [35].

Hyperparameter tuning is done concurrently with K-fold cross validation. Hyperparameters are metrics of the machine learning algorithm, and for the Random Forest, these are:

- Number of decision trees in the random forest. Usually, the higher the number, the better, but there is a limit to the benefit [36].
- The maximum depth of the trees. This parameter governs the number of splits made. If unconstrained, the tree will split until every leaf is pure, where every sample at that leaf is from the same class [37].
- The minimum number of samples (feature examples) to make a split. This value can vary from at least one sample at a node to every sample at a node [37].
- The split criterion, either gini or entropy.
- The minimum samples leaf, which is the number of samples required to be contained within a lead node [37].
- Maximum features, which is the number of features that are considered when looking for the best split [37].

The metrics to tune for the SVM model are:

- Kernels, these are used to transform the data from low dimensional input space to a higher dimensional space. This transformation is particularly useful in non-linear problems [17].
- C (regularisation), is the term which tells the SVM optimization how much error or misclassification is tolerable [17].

- Gamma, which governs the distance which is taken into account to calculate the plausible line of separation [17].

One can determine the “best” model and its corresponding hyperparameters when performing K-fold cross validation.

K-fold cross validation can require a significant amount of computational power to run as the model must be trained and tested for each k, and the data split into their respective sets. If sufficient computational power is not available, this can limit one’s ability to properly perform cross validation.

4.1.2 Experiment Zero: Binary Classification of GRC-01 and MMS 2mm Mars terrain simulants

To test the Random Forest’s and SVM’s ability to classify terrain based upon the features selected, a simple binary classification was performed. The non-linear kernel options were explored for SVM multi-class classification rather than the linear SVM ”one-vs-all” methods in an effort to reduce computational complexity. K-fold cross validation was used to tune the hyperparameters and assess the model’s average accuracy. For cross validation, K was selected as 5, and the validation was iterated 20 times on portions of the hyperparameter ranges provided, for a total of 100 fits for validation. Table 4.1 and 4.2 show the range of values, or setting options, that were provided as possible solutions to the hyperparameter settings. The ranges were taken from [36] and [17] with adjustment with respect to the number of features in this thesis work. Table 4.3 and 4.4 show the tuned hyperparameters after K-fold cross validation.

4.1.3 Experiment One: Multi-Class Classification of GRC-01, MMS 2mm, and WED-730 Mars terrain simulants

Once binary classification of terrain was verified as possible, a third terrain, WED-730, was added to the dataset. WED-730 was added with the intention to examine the capabilities of multi-class classification of these terrain simulants given the features used. Table 4.5 and 4.6

Parameter	Range
Number of Trees	10:1000
Maximum Depth	10:110
Minimum Samples Split	2, 5, 10
Criterion	‘gini’, ‘entropy’
Maximum Features	1, 2, 3
Minimum Samples Leaf	1:10

Table 4.1: Hyperparameter ranges/options for Random forest using K-fold cross validation

Parameter	Range
Kernel	‘rbf’, ‘sigmoid’
C	0.1, 1, 10, 100
Gamma	1, 0.1, 0.01, 0.001

Table 4.2: Hyperparameter ranges/options for SVM using K-fold cross validation

Parameter	Tuned Setting
Number of Trees	560
Maximum Depth	54
Minimum Samples Split	2
Criterion	‘entropy’
Maximum Features	2
Minimum Samples Leaf	1

Table 4.3: Tuned hyperparameters for Random Fores binary classification of GRC-01 and MMS 2mm.

show the tuned hyperparameter settings of this experiment.

Parameter	Range
Kernel	‘rbf’
C	1
Gamma	0.001

Table 4.4: Tuned hyperparameters for SVM using K-fold cross validation

Parameter	Tuned Setting
Number of Trees	450
Maximum Depth	54
Minimum Samples Split	2
Criterion	‘entropy’
Maximum Features	2
Minimum Samples Leaf	1

Table 4.5: Tuned hyperparameters for binary classification of GRC-01 and MMS 2mm.

Parameter	Range
Kernel	‘rbf’
C	1
Gamma	0.001

Table 4.6: Tuned hyperparameters for SVM using K-fold cross validation

4.1.4 Experiment Two: Multi-Class Classification of GRC-01, MMS 2mm, WED-730, and MMS Coarse Mars terrain simulants

Once the multi-class classification capabilities of terrain with the dataset was determined, a fourth terrain, MMS Coarse, was added. This terrain is particularly interesting for use in discussing the performance of the features used because MMS 2mm and MMS coarse are the same terrain ground into different grain sizes. Wheel performance will be different on each terrain, though marginally. Experiment Two was to examine how well the model performed in classifying similar terrains. Table 4.7 and 4.8 show the tuned hyperparameter settings for this experiment.

Parameter	Tuned Setting
Number of Trees	450
Maximum Depth	54
Minimum Samples Split	2
Criterion	‘entropy’
Maximum Features	2
Minimum Samples Leaf	1

Table 4.7: Tuned hyperparameters for binary classification of GRC-01 and MMS 2mm.

Parameter	Range
Kernel	‘rbf’
C	1
Gamma	0.001

Table 4.8: Tuned hyperparameters for SVM using K-fold cross validation

When considering the options to improve model performance, it is customary to first consider feature engineering and gathering supplemental data [36]. When it was seen that the selected features performed well, further feature engineering was not done. It was also not possible to obtain more data (the data set was already quite large) due to the assembly of a new wheel (with no change to design) and the phasing out of the existing wheel as it’s sensors degraded. The data used in this thesis work was collected over a year ago when the wheel’s components were new. For these reasons, hyperparameter tuning was done. Hyperparameter tuning, and thus cross validation, work to reduce the effects of overfitting [36].

4.2 Classification Results

The subsequent sections discuss the performance characteristics of the two classifiers used in this work, including confusion matrices and metrics.

4.2.1 Experiment 0: Binary Classification of GRC-01 and MMS 2mm Mars simulants

The performance metrics of the two classifiers are shown in Table 4.9. One can see that the Random Forest model outperformed the SVM model. The confusion matrix is shown in Figure 4.4.

Model	Accuracy	Precision	F1	Recall
Random Forest	0.94	0.9423	0.9418	0.9407
SVM	0.6149	0.6125	0.6105	0.6106

Table 4.9: Average 10-fold cross validation performance scores.

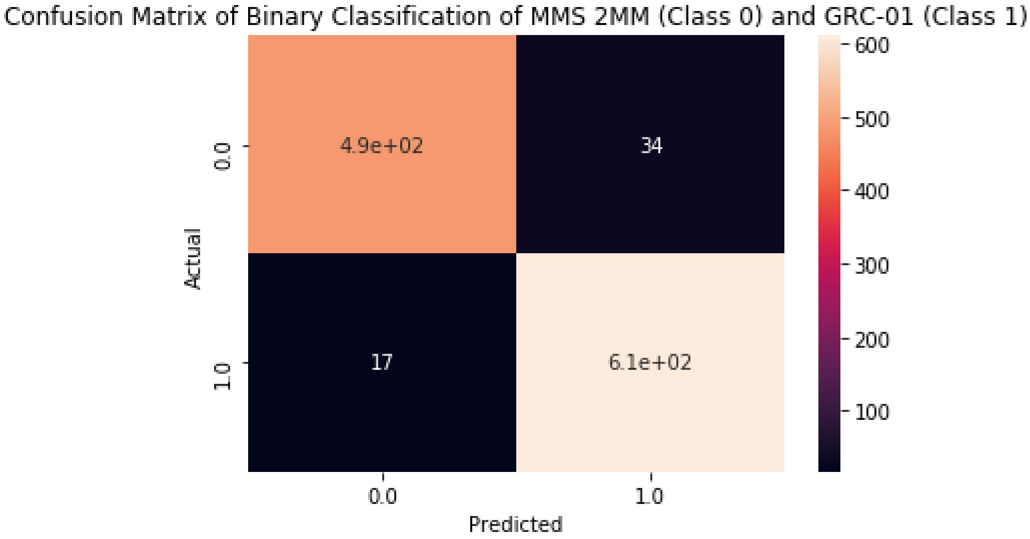


Figure 4.4: Confusion Matrix for Random Forest binary classification of MMS 2mm (Class 0) and GRC-01 (Class 1).

4.2.2 Experiment 1: Multi-Class Classification of MMS 2mm, GRC-01, and WED-730 Mars simulants

The performance metrics of the two classifiers are shown in 4.10. Once again, the Random Forest performed better than the SVM. The confusion matrix is shown in Figure 4.5

Model	Accuracy	Precision	F1	Recall
Random Forest	0.909	0.9083	0.9077	0.9080
SVM	0.5983	0.5688	0.5717	0.6006

Table 4.10: Average 10-fold cross validation performance scores.

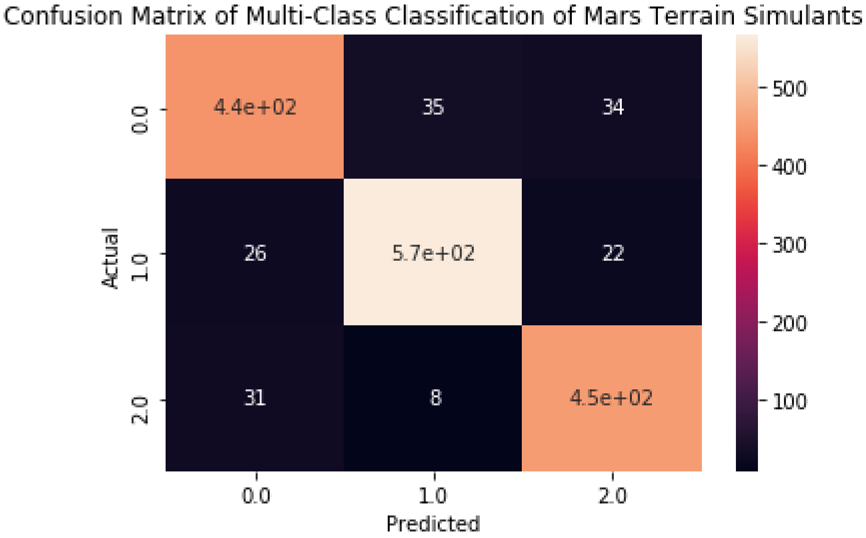


Figure 4.5: Confusion Matrix for Random Forest multi-class classification of MMS 2mm (Class 0), GRC-01 (Class 1), and WED-730 (Class 2).

4.2.3 Experiment 2: Multi-Class Classification of MMS 2mm, GRC-01, WED-730, and MMS Coarse Mars simulants

The performance metrics of the two classifiers are shown in Table 4.11. Its clear that through all three experiments Random Forest is able to better capture information from the data to classify terrain. The confusion matrix is shown in Figure 4.6

Model	Accuracy	Precision	F1	Recall
Random Forest	0.8114	0.8122	0.8115	0.8135
SVM	0.4669	0.4342	0.4359	0.4767

Table 4.11: Average 10-fold cross validation performance scores.

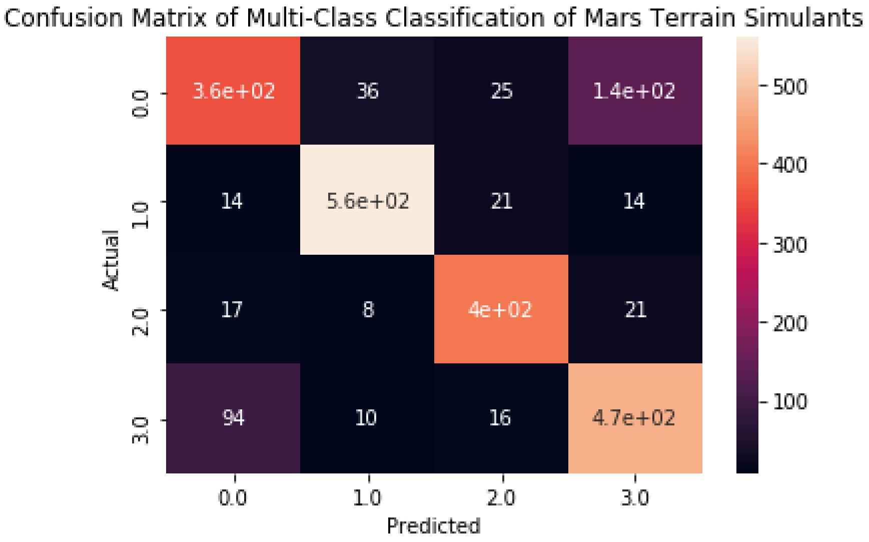


Figure 4.6: Confusion Matrix for Random Forest multi-class classification of MMS 2mm (Class 0), GRC-01 (Class 1), WED-730 (Class 2), and MMS Coarse (Class 3).

Chapter 5

Estimation of Terrain Grain Size

An area of interest to planetary scientists is the estimation of terrain grain size. Grain size can be used to infer the history of an area of a celestial body and what geographical processes may have occurred throughout it's history. According to Nadine Barlow [31], the size of terrain materials at a specific location on a planet's surface depend on the geological processes that occurred or are occurring in that area. Weathering breaks material into smaller pieces of that material. Weathering can be an indication of the existence of water in the area in the past [31]. Differences in colour on the surface of Mars are, in addition to other processes and the presence of various minerals, are due to changing grain size. It can be an indication of the composition of regolith at that location. Grain size can be interpreted from thermal inertia data, as larger rocks will hold energy longer than fine grained sands [31]. While external estimations provide insight, the ideal case is to gain a more direct measurement of grain size from a planetary rover in-situ. It was investigated whether this could be done with *Barefoot Rover* data. The ability of a sand to compress under load is influenced by the grain size of the sand. Larger grains are unable to pack as closely together as smaller grains. Grain roughness also influences sand compressibility. The sinkage exponent is a terramechanics terrain parameter that sheds light on the compressibility of terrain and is shown as N in the following equation [4]:

$$F_n = rb(\theta_1 - \theta_2)[K_s r^N (\cos\theta_m - \cos\theta_1)^N] \cos\theta_m / 2 \tag{5.1}$$

Where F_n is the load felt by the wheel, r is the wheel radius, b is the contact width, K_s is the sinkage modulus and is influenced by b. θ_1, θ_2, and θ_m are contact angles and where the maximum pressure occurs, respectively. $K_s = K_c/b + K_\phi$. Values for K_c and K_ϕ were taken from [5] for dry sand and were 3.9 and 0.1, respectively. K_s can then be calculated for at each time bin using information about b from pressure pad data. All other terms in 5.1 except for N (which is the sinkage exponent, and is to be estimated) can be determined through *Barefoot Rover* wheel data. The goal of sinkage exponent estimation is to determine if a correlation of reasonable bounds exists between sinkage exponent and grain size, in order for grain size estimates to be made in situ from *Barefoot Rover* wheel data. The sinkage exponent estimation was performed in Python, where the scipy minimize function was used to regress each time step of data to determine sinkage exponent values for eight data collection runs each for MMS Coarse, MMS Intermediate, and MMS 2mm Mars simulants. The least squares method 'SLSQP' was used for the minimize function. Grain sizes of the three simulants in question were taken from data collected by an individual at NASA-JPL and was available in the *Barefoot Rover* project repository. The three simulants were chosen for their sequential decrease in grain size from Coarse to 2mm, which would make it more obvious if a strong relationship between sinkage exponent and grain size existed. A mean sinkage exponent value for each simulant was determined, along with a variance. These values were plotted against sinkage exponent of the corresponding simulant and figure 5.1 shows the results of this.

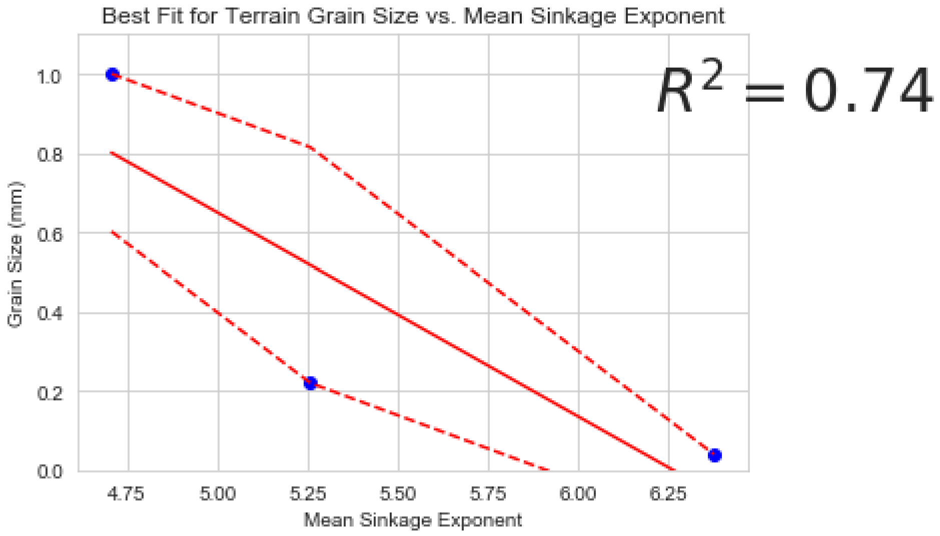


Figure 5.1: Grain size vs. Mean sinkage exponent values for MMS Coarse, MMS Intermediate, and MMS 2mm (in that order) with error ranges.

An R^2 value of 0.74 is considered a reasonable indication of a linear correlation given the noise in the data. More sensitive sensors and less noisy data should improve the correlation fit more.

Chapter 6

Discussion and Conclusion

Sandy terrains found on the surface of Mars are characterized by their cohesion, angle of friction, and grain size. Cohesion and angle of friction are important features in terramechanics mathematical models. Various methods have been employed to determine these characteristics from in-situ rovers, like *Spirit* and *Curiosity*. R. E. Arvidson et al. [2] used a combination of in-situ data and orbital thermal data to estimate the physical properties of Martian terrain encountered by *Curiosity*. Though their results are sound, their method is highly computationally complex and is not feasible for deployment on a rover. In a separate study, R. Sullivan et al. [1] used *Curiosity* wheel scuffs and trenches to estimate cohesion and angle of friction of the terrain. However, they acknowledge that the estimation of sinkage, which as discussed in this thesis is an important value for estimating terrain characteristics, is not exact and relied on imagery in addition to rover telemetry.

This previous work, *Spirit*'s immobilization in duricrust terrain, the limited knowledge about Martian terrain characteristics, and the need for a low-weight and robust solution motivated this thesis work. The *Barefoot Rover* instrumented wheel is a platform to test the feasibility of an instrumented rover wheel in providing information to gain insight into terrain characteristics, with in-situ measurement and without relying on inaccurate visual aids. This thesis investigated and demonstrated that the information collected by the *Barefoot Rover* wheel can be used to estimate cohesion and angle of friction of different terrains. This thesis also demonstrated a Random Forest model is capable of utilizing the *Barefoot Rover* data to predict the class of three Mars regolith simulants with 90% accuracy, and four simulants with 81% accuracy, the reason for this decrease in accuracy will be discussed in the following subsection. This thesis also verified that terramechanics based features are effective features for use in the classification of terrain. In addition, this thesis has been able to show that it is possible to estimate grain size of the simulant based on the calculated mean sinkage exponent values for individual simulant types from *Barefoot Rover* data.

6.1 Cohesion and Angle of Friction Estimation

Cohesion and Angle of Friction were able to be estimated within known ground truth bands for MMS 2mm and GRC-01 using data collected by the *Barefoot Rover* wheel. The estimations occur at every time step in the data, and thus the values are 'apparent' and not a bulk value like the ground truth. The apparent values offer greater resolution to changes in terrain than one single value for each data collection run. It was shown that the data contained enough information in it to distinguish cohesion and angle of friction between loose, gradually cemented, and fully duricrusted terrain.

6.2 Machine Learning Model Performance

The Random Forest decision tree outperformed the SVM model in every experiment. The Random Forest was initially chosen as the algorithm of choice because it's solution is not a black box, one can examine the decision tree and gain physical insight into the problem. The algorithm is not computationally complex which is important for Mars rovers with limited computing power, and Random Forest is known for performing well in multi-class classification problems. Given the performance of the Random Forest it was deemed unnecessary to expand to other possible machine learning solutions like neural networks. Table 6.1 shows the time each model took on average to predict on a test set on the author's 16 GB RAM computer. The SVM model takes over twice as long to predict and performs worse. In addition, Nitze notes that computation effort for neural networks has been shown to be much more intensive than for decision trees, making Random Forest a lightweight and robust solution for limited computation power [38]. The *Perseverance* Mars rover has a computer with 256 MB of RAM [39]. A newer model cellphone has as much as 2 GB of RAM. With that computer power disparity in mind, it is evident that utilizing that lowest computationally expensive machine learning model is important.

Model	Time to Predict (seconds)
Random Forest	0.24
Support Vector Machine	0.59

Table 6.1: Model computational performance.

Both algorithms were fed the same data and both were subject to K-fold cross validation and subsequent hyperparameter tuning. For the Random Forest, accuracy, precision, F1, and recall were all significantly higher than corresponding values for the SVM model. For the three class experiment, these were 0.9091, 0.9083, 0.9077, 0.9080, respectively, in comparison to 0.5983, 0.5688, 0.5717, 0.6006, respectively for the SVM model.

Experiment 1 was multi-class classification of three Mars simulants; MMS 2mm, GRC-01, and WED-730. This experiment had an accuracy of 91% using the Random Forest. Experiment 2 was multi-class classification of four Mars simulants; MMS 2mm, GRC-01, WED-730, and MMS Coarse and had an accuracy of 81%. As seen in the Results section, the confusion matrix of Experiment 2, figure 4.6, showed that the model became confused between class 0 (MMS 2mm) and class 3 (MMS Coarse). The reason for the mis-classification is because the two simulants are the same terrain at the fundamental level, the only difference being their grain size. As such, they will share similar characteristics. The model's confusion between the two classes was actually a positive result, as it showed that the model was reflecting reality. If the model had been able to distinguish these two simulants without difficulty, that would have been cause for concern as for the validity of the data.

6.3 Feature Importance

The most important feature in the Random Forest model was motor torque, which shear stress is a direct function of. This feature was engineered from motor current data. Figures 6.1, 6.2 and 6.3 show the feature importances for all three machine learning experiments. In terramechanics, the shear stress at wheel-terrain interface is a defining feature of the wheel-terrain interaction and encompasses information about the cohesion and angle of friction. The fact that shear stress is a more important feature than contact pressure and wheel slip is physically valid as these other two features are lesser pictures of the overall wheel-terrain interaction but still hold valuable information. The feature importance was obtained using the built-in feature_importances_ function in the Scikit-learn Python library.

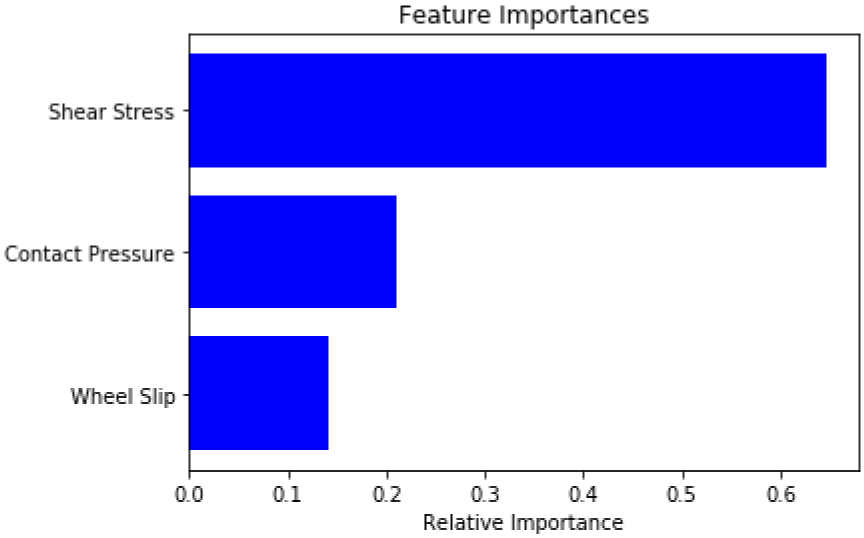


Figure 6.1: Feature importances for Experiment Zero.

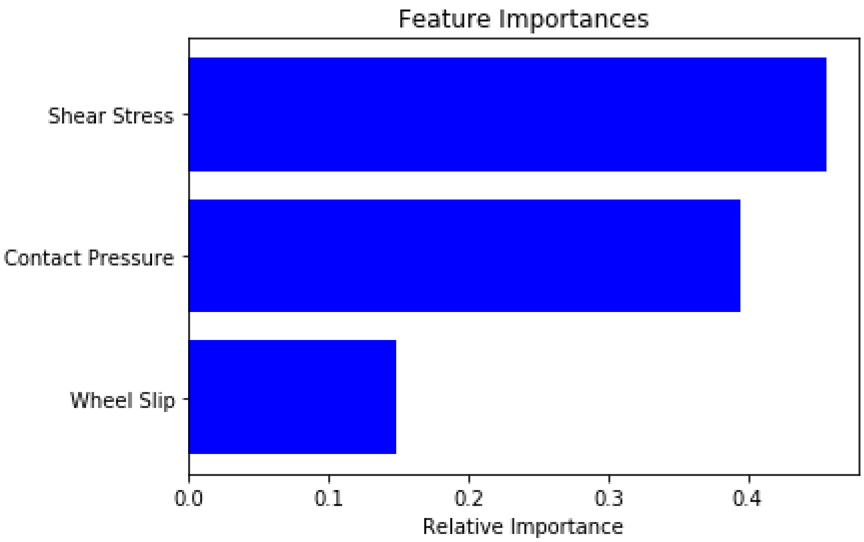


Figure 6.2: Feature importances for Experiment One.

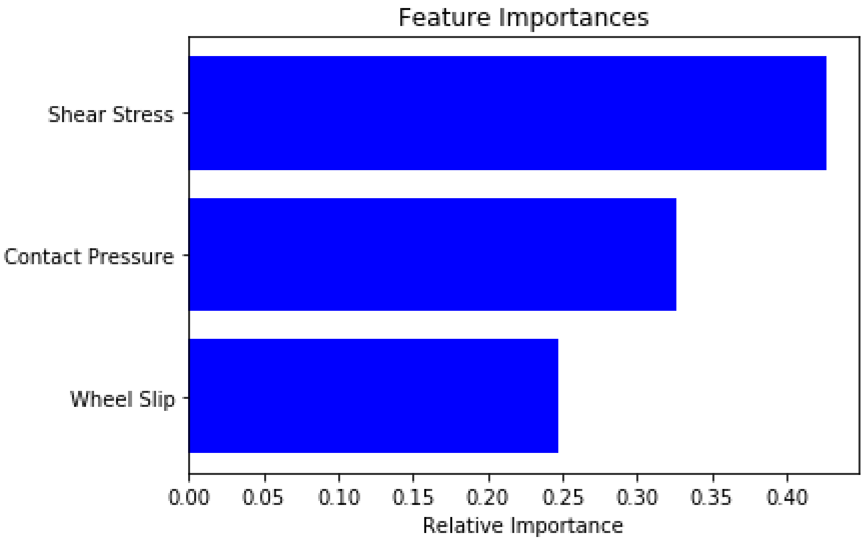


Figure 6.3: Feature importances for Experiment Two.

6.4 Conclusion

This thesis work proves the viability of an instrumented rover wheel in determining terramechanics terrain parameters such as cohesion and angle of friction, the use of machine learning for classifying terrain using data collected by this instrumented wheel, and again using such data to estimate terrain grain size. This thesis has given geologic cause that instrumented wheels should be used in future rover missions. The planetary scientist can benefit greatly from knowledge of cohesion, angle of friction, and grain size of terrain. These three pieces of information can be used to understand what geologic processes may have or be occurring in specific regions. Previously, the estimation of terramechanics parameters relied on the aid of often inaccurate imagery estimations and orbital data.

This thesis has also shown that terrain can be classified from instrumented wheel data, supporting both safe driving and autonomous driving endeavours. The rover driver can use terrain classifications to make driving decisions and an autonomous rover can use this information in its control system. Random Forest and Support Vector Machine algorithms were utilized to classify Mars simulants in binary and multi-class classification. Random Forest outperformed the Support Vector Machine model with a binary terrain classification accuracy of 94% and a maximum multi-class classification accuracy of 81%.

6.4.1 Future Work

The natural next evolution of this research is the development of a machine learning solution for planetary deployment. An unsupervised solution is needed, as the terrain types on Mars are not well understood and they will need to be clustered as data is collected in-situ. This area of future work can be included within the development of autonomous rover systems, as the knowledge of terrain features can serve as flags for unsafe terrain or terrains of scientific interest. A possible evolution of this research is the use a deformable rover wheel (such as those used during the Apollo landings) to collect similar data. NASA-JPL is currently focusing research work into deformable wheels for future Mars rovers.

6.4.2 Machine Learning Techniques

This thesis discussed a proof-of-concept demonstration that from instrumented rover wheel data, terrain simulants can be classified with reasonable accuracy. It indicated that such technology and methodology can be useful on the surface of Mars with some adjustment for unsupervised techniques.

6.4.3 Grain Size Estimation

This thesis provided a brief proof-of-concept that instrumented rover wheel data can be used to estimate sinkage exponent of terrain and establish a correlation between grain size and sinkage exponent values. Knowledge of grain size provides insight into geological processes that the area may have undergone and increases our understanding of celestial surfaces.

Bibliography

[1] R. Sullivan *et al.*, “Cohesions, friction angles, and other physical properties of martian regolith from mars exploration roverwheel trenches and wheel scuffs,” *Journal of Geophysical Research*, vol. 116, 2011.

[2] R. E. Arvidson *et al.*, “Terrain physical properties derived from orbital data and the first 360 sols of mars science laboratory curiosity rover observations in gale crater,” *Journal of Geophysical Research: Planets*, vol. 119, Issue 6, pp. 1322–1344, 2014.

[3] R. E. Arvidson *et al.*, “Physical properties and localization investigations associated with the 2003 mars exploration rovers,” *Journal of Geophysical Research*, vol. 108, 2003.

[4] L. Ding, “Identifying mechanical property parameters of planetary soil using in-situ data obtained from exploration rovers,” *Planetary and Space Science*, vol. 119, pp. 121–136, 2015.

[5] M. G. Bekker, *Introduction to Terrain-Vehicle Systems*. The University of Michigan Press, 1969.

[6] J. Wong, *Theory of Ground Vehicles*. John Wiley & Sons, Inc., 2001.

[7] D. F. Lapidus, *Collins dictionary of geology*. BennettBooks Ltd, 2006.

[8] A. Mehta. *et al.*, “The dynamics of sand,” *Reports on Progress in Physics*, vol. 57, p. 383, 1999.

[9] NASA, “Wheels and Legs.” `https://mars.nasa.gov/mars2020/spacecraft/rover/wheels/`, 2020. 2020-03-24.

[10] H. Shibly *et al.*, “An equivalent soil mechanics formulation for rigid wheels in deformable terrain, with application to planetary exploration rovers,” *Journal of Terramechanics*, vol. 42, pp. 1–13, 2005.

[11] M. Cross, *Estimating Terrain Parameters With a Rigid Wheeled Rover Using Neural Networks*. Thesis, Carleton University, 2012.

[12] S. Higa *et al.*, “Measurement and modeling for two-dimensional normal stress distribution of wheel on loose soil,” *Journal of Terramechanics*, vol. 62, 2015.

[13] W. Sullivan, *Machine Learning Beginners Guide Algorithms: Supervised & Unsupervised Learning, Decision Tree & Random Forest Introduction*. Healthy Pragmatic Solution Inc, 2017.

[14] R. Saxena, “How Decision Tree Algorithm Works.” `https://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/`, 2017. Accessed: 2020-01-14.

[15] W. Mongwe, “A Random Forest Test For Jumps in Stock Markets Using R.” `https://www.wilsonmongwe.co.za/an-interactive-random-forest-test-for-jumps-in-stock-markets-using-r/`, 2017. Accessed: 2020-01-25.

[16] D. Jedamski, “What is Support Vector Machine?.” `https://www.lynda.com/Python-tutorials/What-Support-Vector-Machine/806167/2809619-4.html`, 2019. Accessed: 2020-02-22.

[17] C. Liu, “SVM Hyperparameter Tuning using GridSearchCV.” `https://towardsdatascience.com/svm-hyper-parameter-tuning-using-gridsearchcv-49c0bc55ce29`, 2020. Accessed: 2020-03-10.

[18] C.-W. Hsu and C.-J. Lin, “A Comparison of Methods for Multi-class Support Vector Machines.” `https://www.csie.ntu.edu.tw/~cjlin/papers/multisvm.pdf`. Accessed: 2020-02-08.

[19] H. A. Park, "An introduction to logistic regression: From basic concepts to interpretation with particular attention to nursing domain," *Journal of Korean Academy of Nursing*, vol. 43, no.2, pp. 154–164, 2013.

[20] A. Saci, "Lecture Notes in ECE 9309: Intro to Machine Learning."

[21] A. Jhunjhunwala, "Is Logistic Regression a good multi-class classifier ?." https://medium.com/@jjw92abhi/is-logistic-regression-a-good-multi-class-classifier-ad20fecf1309, 2019. Accessed: 2020-02-22.

[22] G. Drakos, "Support Vector Machine vs Logistic Regression." https://medium.com/@george.drakos62/support-vector-machine-vs-logistic-regression-94cc2975433f, 2020. Accessed: 2020-03-16.

[23] P. Płoński, "Random Forest vs Neural Network (classification, tabular data)." https://mljar.com/blog/random-forest-vs-neural-network-classification/, 2019. Accessed: 2020-08-22.

[24] Pathmind, "A Beginner's Guide to LSTMs and Recurrent Neural Networks." https://wiki.pathmind.com/lstm, 2020. Accessed: 2020-08-22.

[25] Uniqtech, "Multilayer Perceptron (MLP) vs Convolutional Neural Network in Deep Learning." https://medium.com/data-science-bootcamp/multilayer-perceptron-mlp-vs-convolutional-neural-network-in-deep-learning-c890f487a8f1, 2019. Accessed: 2020-08-22.

[26] H. Ferreira, "Confusion matrix and other metrics in machine learninga." https://medium.com/hugo-ferreiras-blog/confusion-matrix-and-other-metrics-in-machine-learning-894688cb1c0a, 2018. Accessed: 2020-02-08.

[27] Anonymous, "Confusion Matrix." https://www.waytoliah.com/1222, 2017. Accessed: 2020-02-08.

[28] L. Jedamski, "Applied Machine Learning: Algorithms." https://towardsdatascience.com/machine-learning-multiclass-classification-with-imbalanced-data-set-29f6a177c1a, 2018. Accessed: 2020-02-08.

[29] U. Today, “Ok, Spirit Rover, Let’s Blow This Pop Stand!.” `https://www.universetoday.com/44920/ok-spirit-rover-lets-blow-this-pop-stand/`, 2009. Accessed: 2020-03-24.

[30] M. M. Battler *et al.*, “Development of martian regolith duricrust simulants for use in rover hazard detection and avoidance experiments.,” *Proceedings of the 49th Lunar and Planetary Science Conference*, 2018.

[31] N. Barlow, *MARS: AN INTRODUCTION TO ITS INTERIOR, SURFACE AND ATMOSPHERE*. Cambridge University Press, 2008.

[32] E. Bressert, *SciPy and NumPy*. O’Reilly Media Inc, 2013.

[33] G. H. Peters *et al.*, “Mojave mars simulant—characterization of a new geologic mars analog,” *Icarus*, vol. 197, pp. 470–479, 2007.

[34] C. Testing, M. Testing, and L. Angeles, “Mars Simulant Mechanical Testing,” tech. rep., California Testing and Inspections, Material Testing & Geotechnical Laboratory, Los Angeles, 2019.

[35] G. Drakos, “Cross-Validation.” `https://medium.com/@george.drakos62/cross-validation-70289113a072`, 2018. Accessed: 2020-02-27.

[36] W. Koehrsen, “Hyperparameter Tuning the Random Forest in Python.” `https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74`, 2018. Accessed: 2020-03-04.

[37] M. B. Fraj, “In Depth: Parameter tuning for Random Forest.” `https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d`, 2017. Accessed: 2020-03-04.

[38] I. Nitze, “Comparison of machine learning algorithms random forest, artificial neural network and support vector machine to maximum likelihood for supervised crop type classification,” *Proceedings of the 4th GEOBIA*, 2012.

[39] NASA, “Rover Brains.” https://mars.nasa.gov/mars2020/spacecraft/rover/brains/, 2020. Accessed: 2020-03-30.

Curriculum Vitae

Name:	Bryan Southwell
Post-Secondary Education and Degrees:	Carleton University Ottawa, ON 2014 - 2018 B.Eng
	University of Western Ontario London, ON 2018 - 2020 MESc
Honours and Awards:	Global Opportunities Award 2019
Related Work Experience:	JPL Visiting Research Internship NASA Jet Propulsion Laboratory 2019
	Teaching Assistant The University of Western Ontario 2018 - 2020