TECHNISCHE UNIVERSITÄT
CHEMNITZ

Fakultät für Informatik

CSR-24-07

Development of a Material Classification Model for for Multispectral LiDAR Data

Túlio Gomes Pereira · Wolfram Hardt · Ariane Heller

August 2024

Chemnitzer Informatik-Berichte

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Development of a Material Classification Model for Multispectral LiDAR Data

Master Thesis

Submitted in Fulfilment of the
Requirements for the Academic Degree
M.Sc.

Dept. of Computer Science
Chair of Computer Engineering

Submitted by: Túlio Gomes Pereira
Student ID: 540825
Date: 02.07.2024

Supervising tutor: Prof. Dr. W. Hardt
Dr. Ariane Heller
Dipl.-Ing. Marco Meinig

Abstract

The advance of the new generation of LiDAR systems, with the capability of measuring the spectrum, suffers from the lack of available hardware and the elevated costs of supercontinuum white light laser systems. This study addresses this issue by assessing the feasibility of creating a spectral dataset with target materials using a commercially available FT-IR spectrometer and applying machine learning with embedded feature selection techniques to identify the most informative wavelengths that allow the classification of materials using lower spectral resolution. Here, the performance of versions of L1-regularized logistic regression and random forest with recursive feature elimination are compared for up to 100 wavelengths. The results indicate that random forest demonstrates superior performance in accuracy for all the versions, with a selected version achieving an accuracy 30.4% higher than the L1-regularized logistic regression while using one less feature. This advantage comes with the drawback of predicting the material of one spectrum 60 times slower. In the end, the challenges of the multispectral LiDAR hardware are discussed, and the random forest version with 5 wavelengths is tested in an example scenario created in a laboratory multispectral LiDAR demonstrator. The results show that this approach is promising for reducing hardware costs through the use of discrete laser systems, as the selected model achieved an accuracy of 84.77% in the scenario created using the amplitude of only 5 wavelengths.

Keywords: LiDAR, multispectral, material, classification

Contents

List of Figures

List of Tables

List of Abbreviations

LiDAR Light Detection and Ranging

EM Electromagnetic

VIS Visible

IR Infrared

NIR Near-Infrared

SW-IR Short-wave Infrared

FT-IR Fourier Transform Infrared Spectroscopy

ToF Time of Flight

FS Feature Selection

RFE Recursive Feature Elimination

IMU Inertial Measurement Unit

ML Machine Learning

AI Artificial Intelligence

SVM Support Vector Machine

RF Random Forest

OOB Out-of-bag

LASSO Least Absolute Shrinkage and Selection Operator

MSE Mean Squared Error

RF-RFE Random Forest Recursive Feature Elimination

AOTF Acousto-optical Tunable Filter

SPCM Single Photon Counting Module

TCSPC Time-Correlated Single Photon Counting

1 Introduction

Autonomous systems, such as robots and vehicles, are systems that can control themselves and move freely in the environment without the necessity of human intervention [60]. Their advance has introduced exciting challenges for technology development. These challenges appear from the high complexity tasks that these systems need to undertake and the strict requirements that they have to fulfill to be deployed [26]. Modern autonomous systems utilize an elevated number of sensors, such as cameras, ultrasounds, and radars combined with data processing algorithms to understand and interact with the environment [57].

These sensors acquire information from the surroundings that are used to feed the embedded algorithms, assisting their decision-making process. Therefore, they apply different techniques to measure a variety of details, such as the distance, and shape of the objects, that will help the system to position itself and interact with the environment [47]. Due to the dependency of the decisions of the autonomous systems on sensor measurements, the progress of these systems intensifies the demand for the development of new and more efficient environmental sensing techniques.

To analyze the data coming from these new technologies, systems need algorithms capable of learning complex patterns. Therefore, most modern autonomous systems utilize machine learning techniques to make their decisions. A basic configuration used by the majority of these systems uses cameras to recognize the objects around [57]. However, the performance of systems based only on camera images degrades under harsh environmental circumstances such as foggy, snowy, rainy, and dark night conditions [60].

In this regard, an alternative that appears as a potential solution to address this issue is the use of light detection and ranging (LiDAR) sensors. These sensors can perform active illumination of the surroundings. Hence, they have the potential to mitigate some of the climate effects, such as measuring during dark illumination situations [43]. The active illumination principle also allows these sensors to determine the distance of the objects in the territory, which leads to the creation of a three dimensional point cloud data of the environment [57].

This amount of information can be further increased when a deeper analysis of the objects in the terrain is necessary. In such cases, LiDAR sensors can be combined with multispectral cameras to acquire spectral data that can be analyzed by the underlying processing algorithm [1]. However, combining these two different sensors increases the challenges of the development of the autonomous system, as it requires precise sensor alignment and rises the system costs [26]. Hence, the alternative to acquire the spectral data of the objects is through the use of multispectral LiDAR sensors. They can measure the three dimensions of the environment and the spec-

tral characteristics of the object simultaneously. The combination of this spectral information with machine learning algorithms can lead to the identification of the material composing the target [54]. This recognition can assist the development of several application domains such as in autonomous vehicles during environment understanding [54], robots in remote sensing and interacting with objects of different materials [16], and aerial vehicles during vegetation mapping [11].

However, the novelty of multispectral LiDAR sensors causes a lack of commercially available hardware capable of acquiring this data. This results in a reduced number of studies on machine learning algorithms that are capable of analyzing these measurements [30]. Thus, this work addresses this issue by comparing different machine learning models applied to materials classification using spectral measurements, and evaluating their behavior when applied to multispectral LiDAR data.

1.1 Motivation

With the novelty introduced by the creation of multispectral LiDAR sensors, there is a demand for machine learning models capable of interpreting this new type of data. This data should allow the classification of the material given the information contained in the spectral dimension of the multispectral point cloud. However, there is a shortage in available hardware due to their elevated costs of development [30]. This complicates the acquisition of large datasets for the training of sophisticated models. An alternative to this problem consists of decreasing the hardware costs by identifying the wavelengths that contain the most relevant information for the classification.

Thus, this work proposes the use of machine learning techniques to identify the most informative wavelengths from spectroscopy measurements of everyday life materials. This approach may allow the possibility of using wavelength selection on spectroscopy data to guide the development of multispectral LiDAR hardware. Since the elevated costs of multispectral LiDAR hardware come mainly from the use of supercontinuum laser systems [46], the selection of wavelengths may allow its substitution for discrete monochromatic laser sources resulting in the reduction of the hardware development costs.

The results of this work should be evaluated on the classification of a multispectral LiDAR point cloud from a multispectral LiDAR demonstrator developed by Fraunhofer ENAS in cooperation with TU Chemnitz.

1.2 Research Goals

1.2.1 Main Goal

The main goal of this work is to develop a classification model that uses spectral measurements in the NIR region to predict the corresponding material, within a pre-defined set of materials.

1.2.2 Specific Goals

To achieve the main goal, this work will present the creation of a dataset with spectral measurements of 4 classes of materials. This dataset will be used to train different versions of two classifiers with embedded feature selection capabilities. Afterwards a detailed comparison of the performance of the models created will be performed and the configuration of the most suitable one will be adopted. The comparison of the models with feature selection capabilities should allow a selection of a subset of wavelengths that contain enough information to perform the classification. In the end, it is expected that this reduced subset of wavelengths allow the classification of materials at each point of the multispectral point cloud measured with a multispectral LiDAR demonstrator.

1.3 Structure of the work

This work is structured into eight chapters. In the first chapter the necessity of studies in the multispectral LiDAR data analysis and its future application on mobile systems domain is introduced. Also the motivation and the goals that guide the development of this work are described.

The second chapter presents a review of the theory behind the techniques used in this work. It starts with the explanation of the theory for spectral data acquisition, followed by the description of the working principle of a LiDAR system, where the difference between a normal LiDAR and a multispectral LiDAR is exposed. Then, the most important concepts of machine learning are covered, and a detailed explanation of the fundamental principles behind the algorithms selected for the application in this work, random forest and L1-regularized logistic regression, is presented. In addition, this chapter introduces the concept of wavelength selection, by the use of feature selection capabilities of the algorithms. Here, the underlying principle of the recursive feature elimination method used in this work is described. Finally, the evaluation metrics employed for comparing the algorithms are illustrated.

The third chapter presents a review of the state of the art and the research directions in the field of multispectral LiDAR data processing. It starts with an overview of the research areas that contribute to the development of mutispectral LiDAR systems and details similar works in the classification of materials based on multispectral LiDAR data. In the end, the conclusions from the literature review that guided the selection of the algorithms for this work are presented.

In the fourth chapter, the methodology proposed for the development of the study is illustrated. The methodology consists of three primary steps. First of all, a material's spectral dataset is created using a calibrated spectrometer. In the second step, the dataset is used to train and evaluate the performance of distinct classification models with and without feature selection. Finally, a comparison of the model's performance is conducted and the most suitable is selected and evaluated in the

point cloud acquired from a multispectral LiDAR demonstrator.

The fifth chapter describes the equipment and tools employed in the implementation of this work. It starts by describing the functionalities of the FT-IR spectrometer used for the dataset creation. After, the Python environment and libraries used for the deployment and training of the machine learning models are explained and the working principle of the multispectral LiDAR demonstrator used for the acquisition of the multispectral point cloud is exposed.

The sixth chapter describes the development of the method proposed. It starts by the acquisition and creation of the dataset, where the materials utilized and the acquired spectrum are discussed. Then a detailed description is presented for the analysis of different versions of the L1-regularized logistic regression with and without feature selection. The same is done with the random forest, where the necessity of wrapping the model with recursive feature elimination for automatic feature selection is exposed. Finally, a sample scenario with different materials is introduced, the multispectral point cloud acquisition is described and the algorithm to perform the model evaluation in this data is explained.

In the seventh chapter the results obtained during the evaluation of the models developed are shown. It starts by the evaluation of the performance of the random forest and the L1-regularized logistic regression without feature selection. The random forest is tuned for the number of estimators, and an optimization is performed to select the model with the smallest forest while maintaining a high accuracy. Afterwards, the feature selection capabilities of the models for 100 wavelengths are evaluated. Therefore the random forest model is placed inside the recursive feature elimination wrapper. Then the performance of the models before and after feature selection are compared and one version is selected to be evaluated in the multispectral point cloud. Finally, the data acquired from the multispectral LiDAR demonstrator is presented, the spectrum of each point is contrasted with the spectrum obtained from the FT-IR spectrometer, a basic simulation of the behavior of the multispectral laser system is performed, and a multispectral point cloud is acquired where the algorithm selected is evaluated.

The final chapter presents the conclusions of this project and gives insights for future work.

2 Background

2.1 Spectroscopy

Spectroscopy is the scientific field that explores the relationship between electromagnetic (EM) radiation, such as light, and materials or substances. The EM radiation is created by the simultaneous oscillation of the magnetic and the electric field at identical frequencies, resulting in the emission of an electromagnetic wave. This wave propagates with an energy level corresponding to the energy of photons traveling at the oscillation frequency. The wavelength (λ) represents the distance between two points of the wave with the same frequency and phase [55]. Figure 2.1 shows the wavelength of an electromagnetic wave, illustrated by the propagation of the electric field.

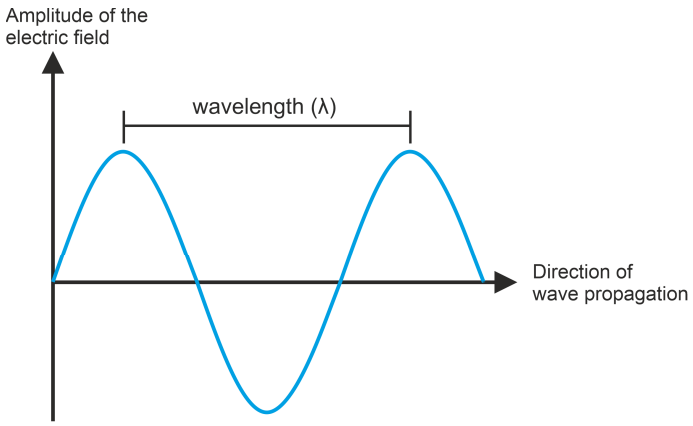


Figure 2.1: Representation of the wavelength of the electric component of an electromagnetic wave.
Source: Adapted from [24].

The EM spectrum is a visual representation of all possible electromagnetic waves, which are categorized by their wavelengths or frequencies. It is segmented into regions with various nomenclatures that are associated with the energy level of the waves. The visible (VIS) range, which varies from 400 to 750 nanometers (nm), denotes the energy levels equivalent to the colors of the rainbow that are perceptible by the human eyes [55]. The region that contains waves with lower energy levels than the red color is called infrared (IR). This region is further divided into different categories. The division of wavelengths ranging from 750 to 1100 nm is called near-infrared (NIR). Although the section from 1100 to 2500 nm of the spectrum is also

part of the NIR [24], some authors consider a different nomenclature for this area, calling it short-wave infrared (SWIR) [18]. The application of light waves in this region of the spectrum to a material, allows the study of the vibrational patterns of the chemical compounds of the matter, producing a unique spectral fingerprint [55].

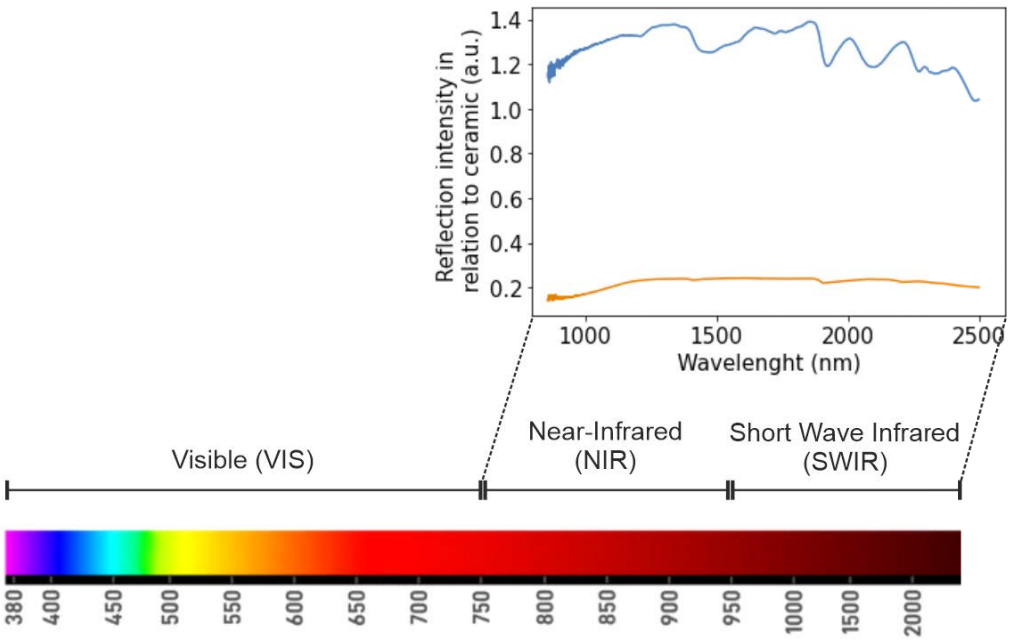


Figure 2.2: Example of the position in the electromagnetic spectrum of a NIR spectroscopy measurement from two materials.
Source: Adapted from [55].

Figure 2.2 shows an example of the spectral fingerprints of two different materials in the NIR region of the electromagnetic spectrum.

The spectrometer is the equipment capable of acquiring the spectral information. A classical spectrometer is an optical system, with three primary components: a light source, a detector, and a dispersing or grading element. The material is measured against a reference, and the dispersing element changes the frequency of the EM wave that reaches the detector, covering the entire spectral region of the material being measured [55]. The analysis and processing of data acquired using a spectroscopy measurement is called chemometrics [40].

Modern spectrometers use Fourier Transform Infrared Spectroscopy (FT-IR) to capture the NIR spectrum. This technique allows waves across all frequencies to reach the detector. The FT-IR method employs a Michelson interferometer and a Fourier Transform to acquire the spectral data. The optical system is comprised of a mobile mirror, a beam splitter, a light source, and a detector. A slight variation in the mirror's position changes the path length traveled by the light source to reach the detector. The light passes through the beam splitter and the sample before hitting the detector. This process generates an interferogram over time, which is then

transformed into the spectrum using the Fourier Transform. The precise location of the moving mirror is determined with high precision using the laser light [55].

2.2 Light Detection and Ranging (LiDAR)

The optical system known as light amplification by stimulated emission of radiation, abbreviated as laser, emits an extremely focused beam or pulse of monochromatic radiation when triggered by an external energy source [49]. The system emits an EM wave that is coherent, and parallel, resulting in a narrow beam in the direction of the emission. These characteristics distinguish the lasers from conventional light sources, since the last diverge the light beams in all directions. An ideal laser system is considered to be monochromatic, which is the nomenclature for systems that emit EM radiation in a single wavelength. However, practical lasers suffer from broadening effects that spreads the spectrum of the emitted EM waves in a finite bandwidth. The spectral waveform emitted can be approximated by a gaussian distribution and the bandwidth is characterized by the spectral distance, in which the amplitude is half of the maximum value, it is known as the full width half maximum (FWHM) parameter of the system. The same gaussian distribution appears in the intensity of light at the end of the laser beam [31]. Even with these imperfections, lasers have emission properties that are narrower and more focus than ordinary light sources. Therefore, these systems are used for measuring distances, ranging, and scanning objects in the surroundings [49]. Figure 2.3 shows the comparison between the spectral emission pattern of a laser and a conventional light source.

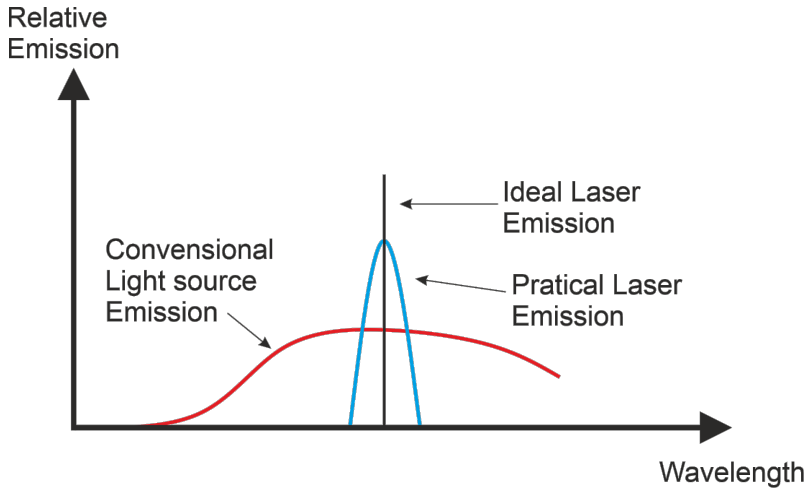


Figure 2.3: Illustration of the emission pattern of a laser compared to an conventional light source.
Source: Adapted from [31].

Laser ranging devices measure the precise distance of a designated target. They first emit a short pulse of laser radiation that travels to the target and is reflected

to the instrument. Afterwards, the time of flight (ToF) of the pulse is calculated by measuring the precise difference in time between emitting the laser pulse and receiving the reflected signal [49]. Figure 2.4 illustrates the working principle of the laser ranging measurement of a target object.

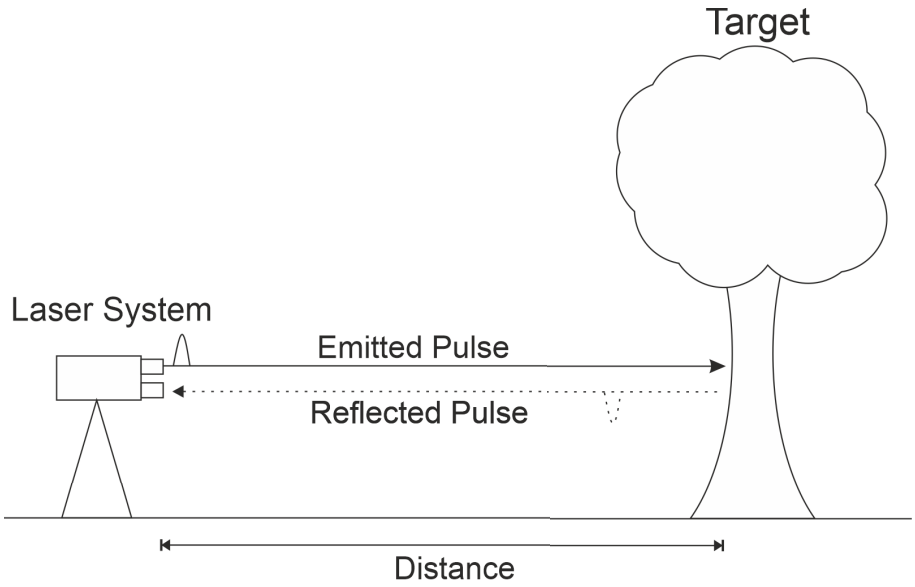


Figure 2.4: Illustration of the laser ranging principle.
Source: Adapted from [49].

Considering the ToF measurement (ΔT) and the speed of light (c), the following formula can be used to calculate the distance (d) [49]:

$$d = \frac{\Delta T c}{2} \tag{2.1}$$

Incorporating a scanning mechanism to the laser ranging system, that moves the laser beam in vertical and horizontal directions, allows the measurement of a three-dimensional (3D) model of the scene. This mechanism can be a motor drive, a rotating mirror, or a prism [49]. The working principle of the scanning measurement is shown in Figure 2.5.

These are the fundamental techniques employed by light detection and ranging (LiDAR) sensors. These sensors are used to perform remote measurements of the environment. They are integrated optical systems consisting of a laser, which emits monochromatic radiation pulses, a detector that detects the reflected light from the target, and the scanning mechanism, which allows a 3D measurement of the scene, to create what is known as LiDAR point cloud [20].

When coupled to an autonomous system, such as robots, drones, or vehicles, these LiDAR devices are known as mobile LiDAR. They are normally equipped with positioning systems such as the Global Navigation Satellite System (GNSS) and the inertial measurement unit (IMU) to assist in the mapping of the environment. In

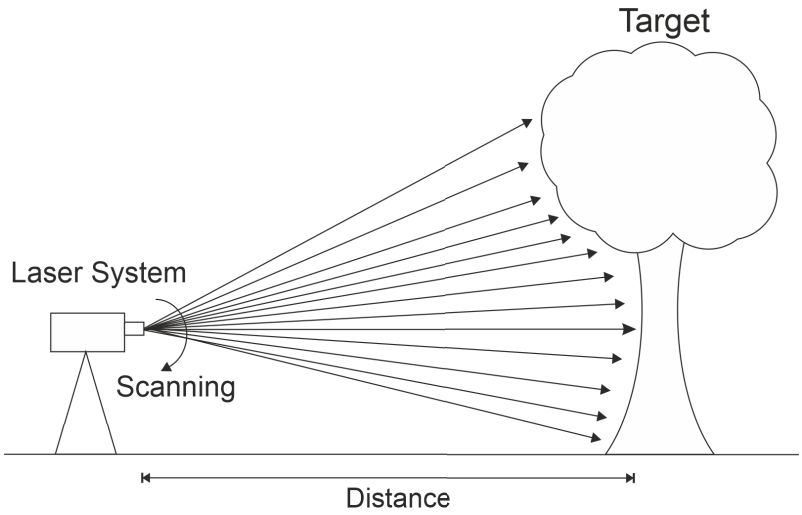


Figure 2.5: Illustration of the laser scanning principle.
Source: Adapted from [49].

some cases, cameras are also used to acquire the color information of the targets in the point cloud, to assist in the post-processing of the data during feature recognition and classification tasks [22].

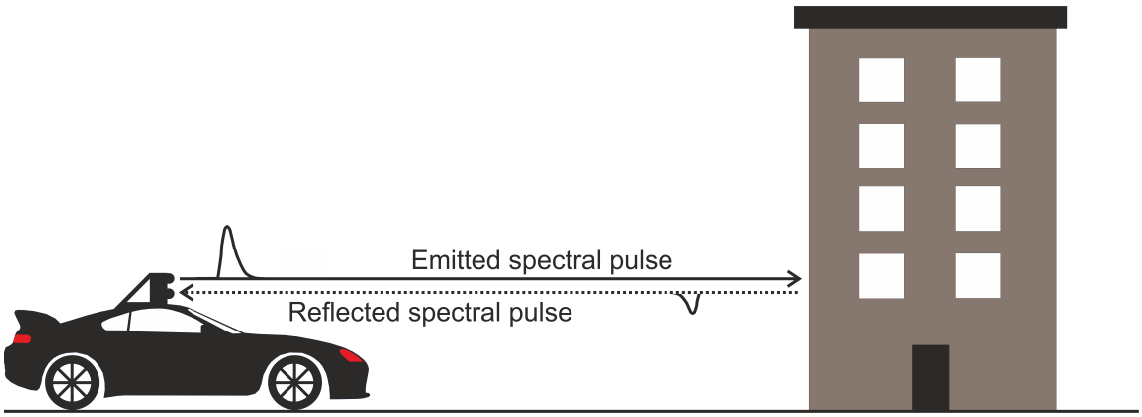
Another method for acquiring color data for LiDAR point clouds includes the use of laser radiation at different wavelengths within the visible spectrum. Systems that use this technique are known as multispectral LiDAR systems [12]. They combine spectroscopy with LiDAR measurements and are capable of creating a hypercube of data containing 3D information about the environment. In addition, they include the measurement of the amplitude of the reflected light at different wavelengths, resulting in a hypercube of multispectral LiDAR point cloud [30].

The comparison between the measurement principles of a standard and a multispectral LiDAR system is shown in Figure 2.6. Figure 2.6(a) illustrates the behavior of a LiDAR system emitting light in a single wavelength for performing distance measurements, whie Figure 2.6(b) represents a mobile multispectral LiDAR system emitting laser pulses in various wavelengths, to add the acquisition of spectral information of the target.

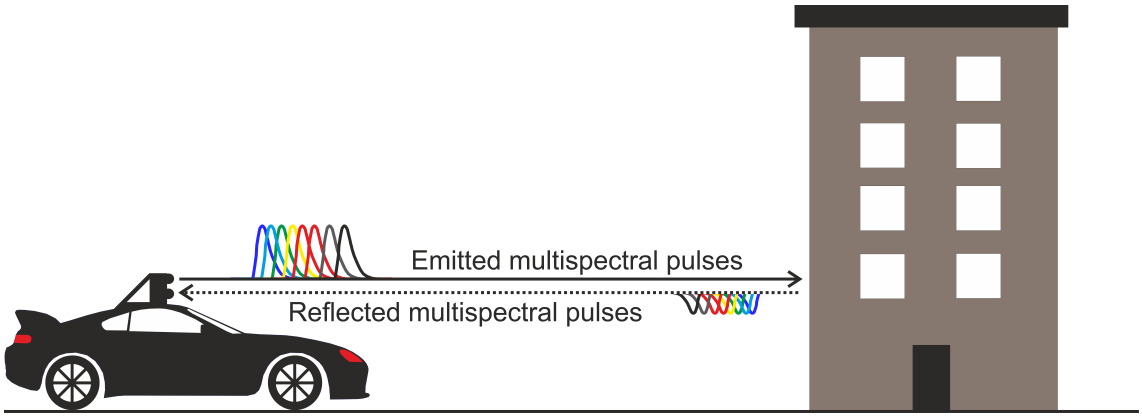
The creation of supercontinuum laser sources, which spread the light power over the spectrum, where several wavelengths in narrow bands can be emitted in parallel, allowed a radiation pattern that simulates white light. Thus, systems that employ this technique are called hyperspectral LiDARs, since they can measure a spread spectral range in several wavelengths with fine resolution [51].

2.3 Machine Learning

The machine learning (ML) term represents a set of theories and methods that are used by the scientific field of artificial intelligence (AI) to select an action for



(a) Example of a monospectral LiDAR measurement.



(b) Example of a multispectral LiDAR measurement.

Figure 2.6: Comparison between monospectral and multispectral LiDAR measurement principles.
Source: Adapted from [28].

interacting with the environment. The aim is to optimize the functions that better describe patterns in a group of data, named dataset, to achieve a high return in the actions performed [29]. ML includes methods and mathematical models to describe the behavior of data, typically acquired from the real world, in specific application domains [62].

ML techniques examine various hypotheses to describe a provided dataset. Depending on their evaluation criteria, the methods can be divided into three groups: unsupervised learning, supervised learning, and reinforcement learning [45]. In supervised learning, the dataset consists of labeled data, this is the name given when each data point in the input space has a corresponding known label in the output space. On the other hand, in unsupervised learning, the dataset only includes the input space, and the goal is to find mathematical functions that map the data points to an unknown output space. Reinforcement learning involves an entity called the agent, which performs an interaction with the environment and receives a numeric

reward as feedback. The agent aims to identify a function to maximize the reward. The reward received after each decision is independent of all previous actions and information [62].

In supervised learning the dataset generated within a specific application domain consists of features and labels. Features are properties that can be easily computed for each data point, while labels are properties that require human expertise in the application to be generated. The process of assigning labels to data points is called annotation, and a labeled dataset represents a set of data composed of annotated data points [29].

Supervised ML methods assume that there exists a relationship between the input features of the data points and their corresponding output labels. The objective of these methods is to discover a mathematical function capable of mapping input features to their respective output labels [62]. If the output space is composed of numeric continuous or discrete variables, the task is called regression, and the model used is called regressor. However, if the output space contains distinct categorical variables, the task to be performed is called classification, and the model is known as classifier [29].

Features refer to the properties of the data points that are being analyzed. They can also be considered as the input variables utilized by the machine learning algorithms. However, selecting which attributes to use can be one of the most challenging aspects of designing and implementing ML models [29]. In certain cases, there might be a inherent choice of features to use, depending on the application domain. For instance, for spectral data analysis, the attributes of each data point represent a series of intensity measurements at different wavelengths of the spectrum. Therefore, using these attributes as features for input in the ML models would be the logical choice in this application domain [13].

The goal of machine learning models is not only to find a hypothesis that best maps the features to the corresponding labels. Instead, it is to find the hypothesis that better generalizes the results to data not included in the dataset. When the models learn completely how to represent the data used for training, but do not maintain the same performance when the unseen data is tested, it is called overfit. Overfitting is one of the most common problems in machine learning models [45].

Another common problem that appears in the ML domain, happens when the generalization of the model is negatively influenced by the inclusion of redundant information into the dataset. To overcome this issue, it is possible to select the most informative features to be used in the model. This process is known as feature selection [4].

2.3.1 Feature Selection

Feature selection (FS) is the process of selecting the most informative features of a set of data or removing redundant information from the dataset. It can be realized by either considering only the information present inside the dataset or using specific technical knowledge about the application [62].

There are mainly three categories of methods data use data-based techniques to evaluate the quality of each feature subset. They are called wrapper, filter, and embedded methods [59]. However, some authors also suggest a fourth approach that involves the manual selection of features based on prior knowledge of the application [48].

Filter methods work independently from the ML algorithm applied. They aim to find the subset of optimal features by considering the results of statistical analysis, or correlations between the features and labels, to filter the original feature set, instead of analyzing their interaction with the ML algorithm [62].

Alternatively, embedded feature selection methods use the intrinsic characteristics of the ML algorithms to identify the features that are most relevant to achieve a desired output. In these methods, there is no clear distinction between the feature selection and the learning phase of the model. Instead, the structure of the ML method already identifies the most important features used for achieving a result by ranking the features according to their relevance, or eliminating unnecessary features [4].

Similarly, wrapper methods use the performance of specific ML algorithms to evaluate the importance of a subset of features. The search strategy used by these methods involves selecting a candidate subset of features before training the ML algorithm. Then the performance is evaluated by a chosen evaluation metric, such as the model's accuracy in this subset. This procedure is repeated several times considering different subsets of features. Finally, the optimal subset is selected. In addition, an evaluation rule, usually the cross-validation technique, can be applied at each run to select the subset of features that result in the highest evaluation score [62].

One example of a wrapper method is called Recursive Feature Elimination (RFE). It was first proposed by the authors in [23] to select a subset of the most informative features used by a support vector machine (SVM) classifier. The authors argue that only ranking the features individually may not result in the best subset of features. Thus, they suggest the use of RFE which is an iterative technique based on three main steps. First, the algorithm is trained using the normal training procedure, then the features are ranked using a chosen ranking criteria, and finally, the features with the smallest ranking value are removed from the features subset. Afterward, the whole process is repeated until a certain stop criterion is met. This algorithm has the advantage that it can be modified to any classifier and any feature ranking criteria. The authors also state that it can be extended for removing more than one feature per interaction, decreasing the computational costs with the possible drawback of classification performance degradation.

2.3.2 Random Forest

Random forests are a particular type of machine learning algorithm built upon a collection of decision trees [7]. Thus, to understand the working principle of the random forest it is first necessary to dive into the theory of decision trees.

A decision tree is composed of nodes interconnected by directed edges. It is a graphical model with a flowchart format that maps the features of the data points to the label space. The nodes receive a special nomenclature according to their position in the tree structure. The nodes at the beginning and end of a tree structure are called root and leaf nodes, respectively. The root and the intermediate nodes are called decision nodes since they hold a hypothesis to evaluate the value of a feature against a binary answer. The result of each decision node drives the flow of the data inside the decision tree [29].

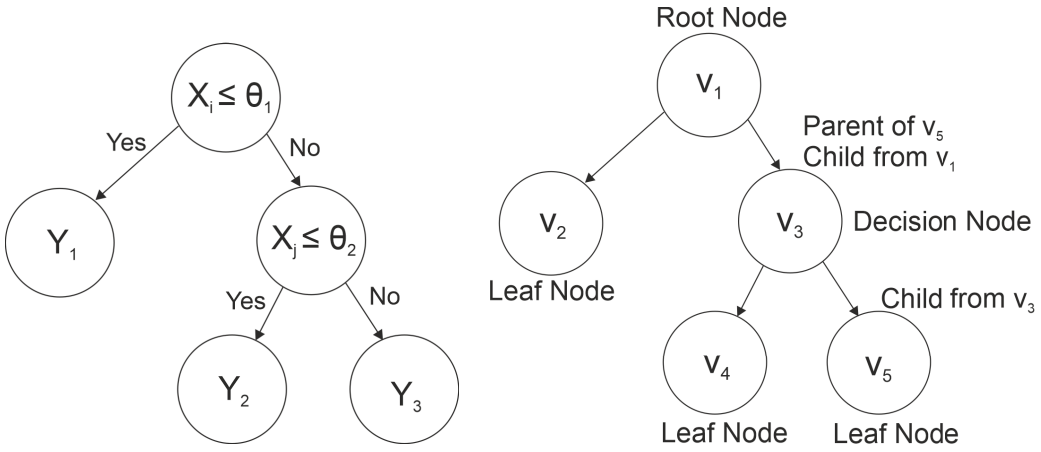
There is only one path from the root to any other node in a tree. Following the direction of the data flow, the node where the path originates is referred to as the parent node, and the subsequent node is called the child node. The root is the only node without any parents, and the leaf is the one that has no child nodes. A single tree may comprise various leaf nodes, but only one root [62].

To create a decision tree for a classification task, the decision nodes search for a binary test question that splits the training dataset into two subsets. This test evaluates a feature's value against a predetermined threshold. In some cases, not all the data points in the subsets created from the split belong to a specific class. This represents the uncertainty of the split about the class labels. The metric that measures the uncertainty is called impurity. The root node has the highest impurity of the tree. The training process of a decision tree seeks to discover the optimal combination of feature and threshold that results in minimal impurity after each split. This iterative process continues through all decision nodes until the leaf nodes are reached. Thus each leaf node represents a distinct class of the classification problem [62].

There are three common ways to calculate the impurity of a split. One of them utilizes the calculus of entropy, which employs a logarithmic function combined with the sum of the probabilities of a data point to be part of a class to evaluate the quality of a split. The highest entropy value indicates the highest level of impurity for the split. Therefore, the split that results in the lowest entropy is selected for the respective node [44]. The second approach uses a function called Gini, which can be considered as an optimization of the entropy calculation, since it eliminates the necessity of calculating the logarithmic function. This results in an optimization of the computation time and is the most common approach used in the literature. Finally, the third technique uses the loss function, or the miss classification error as an evaluation metric for selecting a split [62].

To illustrate these concepts one example of a decision tree applied to the classification of three classes y = (Y_1,Y_2,Y_3) using a set of features x = (X_i,X_j) and two thresholds θ = (θ_1,θ_2) is presented in Figure 2.7(a). Figure 2.7(b) presents the nomenclature given to each node of the same decision tree.

The parameters of the decision trees that are design dependent are called hyperparameters. The depth of the decision tree is an example of them. Deeper trees can effectively learn how to handle the training data, becoming susceptible to overfitting as their performance decreases during the analysis of the test data. Another drawback of the decision trees is their sensitivity to feature noise, due to the split



(a) Example of a classification tree. (b) Name of the components of a decision tree.

Figure 2.7: Example of a decision tree.
Source: Adapted from [62].

evaluation at each node, resulting in poor classification performance in noisy data.

To overcome these drawbacks, Breiman [7] proposed the use of random forests. It builds a group of randomly created decision trees and performs a majority vote among the classes predicted by each of the trees. Therefore, random forests are more robust against the noise influence and the overfitting.

To generate the trees in the forest, the dataset undergoes an initial division into smaller subsets, each of them consisting of randomly selected data points from the original dataset. This technique is often called bootstrapping [6]. Some of the data points can be repeated inside a subset due to the use of a procedure called sampling with replacement. The unused data points from the original dataset are called out-of-bag (OOB) of a subset. For each bootstrapped subset, a set of randomly chosen features is utilized to train a decision tree. The process is repeated for all the subsets, resulting in a forest of decision trees [7]. Finally, the majority vote on the prediction of each tree is performed, aggregating their results. This process of employing bootstrapping and aggregating in sequence is known as bagging [6].

One characteristic of the random forests is the possibility to calculate an importance metric for each feature in the data set. There are several approaches that can be applied to achieve this metric [8]. One of them starts by evaluating the prediction performance of each tree when tested on the respective OOB data points. Then, to calculate the importance of a feature, a permutation is applied to this feature in all OOB data points, and the tree is re-evaluated. The change in prediction performance is then computed by comparing the results with the first evaluation. The increase in the miss classification rate is saved. This procedure is carried out for all the trees in the forest, and the average of the results is used as the feature importance for this specific feature [7].

An alternative approach is capable of calculating the feature importance during

the training phase by evaluating a node in any tree that uses a selected feature. The improvement in the impurity measure is then computed from the parent to the child node of the current node under evaluation. This procedure is repeated for all the nodes in the forest that use the selected feature. Finally, the sum of the improvement results is computed and used as the feature's importance. The same process is performed for all the features in the dataset. When the gini impurity metric is employed, the method is called gini importance [8].

Due to the intrinsic characteristic of random forest to assign importance to features, it can be classified as an embedded feature selection method [62].

2.3.3 Least Absolute Shrinkage and Selection Operator - LASSO

The concept of the least absolute shrinkage and selection operator (LASSO) was initially introduced by the authors in [56]. Their proposal involved adding the L_1-norm of the coefficients as a penalty term for the loss function of linear least squares regression. They have shown that the use of the lasso penalty forces the weight of the coefficients to be zero, thus performing feature selection. The principle behind the lasso proposition relies on the comprehension of linear regression.

In the context of linear regression, also referred to as least square regression, the algorithm seeks to identify a linear function that effectively describes a given training dataset while maintaining the ability to generalize. This can be achieved by fitting a weighted (ω) sum of the inputs, along with adding a bias term (b), to a specified set of features (x), and subsequently assessing the prediction of the function ($\hat{y}$) in comparison to the true output data (y). Equation 2.2 presents an example of the function that the linear regression aims to approximate, particularly in a simple scenario with k features [39].

$$\hat{y} = \sum_{i=1}^{k} \omega_i x_i + b = \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 ... + \omega_k x_k + b = \omega^T x + b \tag{2.2}$$

In order to assess the quality of the fit, the algorithm computes the mean squared error (MSE), which is the average of the distance between each predicted value to the respective ground true of the training set. This is also known as the loss function for this algorithm [39].

$$L(w, b) = \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - \hat{y}^{(i)})^2 \tag{2.3}$$

An example of the MSE of the whole dataset considering N training data points is shown in Equation 2.3. The objective is to minimize this loss function by updating the value of the weights and bias interactively using the calculus of the gradient descent in relation to each variable. For the sake of computational performance, the loss function can be simplified for the loss of an individual data point, shown in Equation 2.4 [39].

$$l_i(w, b) = (y^{(i)} - \hat{y}^{(i)})^2 \tag{2.4}$$

However, the interactive minimization of the loss function may introduce overfitting, since the continuous update of the weights and bias, often leads to a model that excessively describes the training data. Consequently, to address this issue, lasso regularization was added as a term to the loss function, to mitigate the overfitting problem. The regularization term is the L_1-norm, showed in Equation 2.5, which is the absolute value of the weights of the model summed with a regularization parameter (λ) [56].

$$||\omega||_1 = \sum_{i=1}^{k} |\omega_i| \tag{2.5}$$

Thus, the regularized individual loss function can be seen in Equation 2.6. As shown by the authors in [56], the regularization forces some of the weights to be zero, and its strength depends on the value of the regularization parameter. A higher value for λ results in a faster decay of the weights to zero. When the number of zeros weights is high the model is called sparse. From Equation 2.2 it is possible to see that features with weights equal to zero are not used by the model to perform a prediction and, hence can be removed. For this reason, this sparsity allows the lasso to perform embedded feature selection.

$$l_i(w, b) = (y^{(i)} - \hat{y}^{(i)})^2 + \lambda||\omega||_1 \tag{2.6}$$

The same method can be expanded to classification problems using L1-regularization in logistic regression. Logistic regression can be viewed as an expansion of linear regression, where instead of using a linear function for the prediction of the output, it applies an activation function, called a logistic function, on top of the linear one before calculating the prediction. This logistic function is also known as sigmoid ($\sigma(x)$), and presented in Equation 2.7 [45].

$$\hat{y} = \sigma(w, b) = \frac{1}{1 + e^{-(\omega x + b)}} \tag{2.7}$$

Figure 2.8 presents a visual representation of the logistic regression model when the activation function is the sigmoid. As can be seen from the Figure, this diagram also corresponds to a basic artificial neuron with multiple inputs which can be modified for various activation functions [45].

Due to the intrinsic characteristics of the sigmoid function, the results of the model are forced to be bounded between zero and one, as can be seen in Figure 2.9. Thus, setting a threshold value to the output of the logistic regression allows binary classification of the results. In addition, a confidence level of the classified prediction can be acquired using the computed value from the sigmoid function [45].

Considering the binary classification scenario, the training procedure evaluates the probability that the predicted sample belongs to the positive class [39]. Therefore,

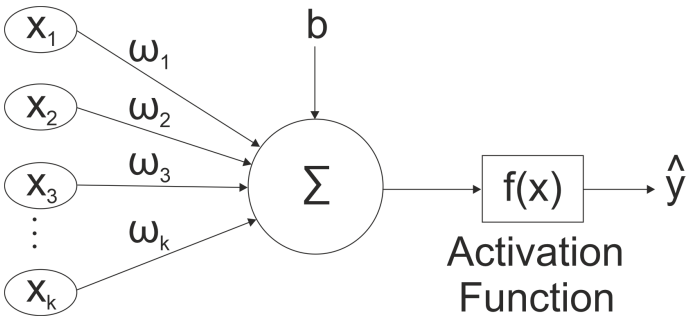


Figure 2.8: Diagram of the logistic regression model.
Source: Adapted from [45].

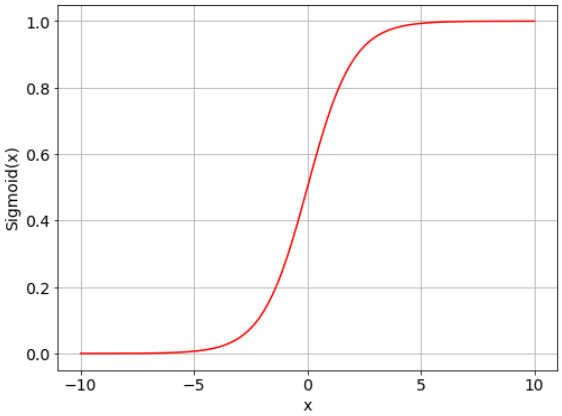


Figure 2.9: Example of the sigmoid($\sigma(x)$) function.
Source: Adapted from [45].

similar to the linear regression context, the objective is to minimize the loss function. In this scenario, the loss function comprises the negative log-likelihood along with the inclusion of the lasso regularization term as shown in Equation 2.8 [34].

$$L(w, b) = \sum_{i=1}^{N} - \log p(\hat{y}^{(i)}|x^{(i)}; w) + \lambda||\omega||_1 \tag{2.8}$$

Extending these concepts for multiclass classification problems, a binary classifier can be trained for each class using logistic regression. So, each class receives a classifier that considers it as positive, and all the other classes are evaluated as negative. This technique is called one-vs-the-rest and uses the confidence score of the logistic regression to take the final classification decision [39]. Another approach to perform multiclass classification consists of changing the activation function for the softmax. First, a linear function is used to compute scores for each class, which are called logit scores. Then the softmax activation function is used to transform these logit scores to the normalized probability of the sample belonging to each class. The loss function, that has to be minimized, is then the cross-entropy between the multinomial

probability distribution predicted and the target probability distribution [3].

2.3.4 Evaluation Metrics

The objective of machine learning algorithms is to create a model that can generalize its predictions to new, unseen data. To ensure this generalization, various techniques can be employed to measure the model´s performance before deployment [27].

One technique used for evaluation is called K-fold cross-validation. It first divides the training data into K different folds of the same size with randomly selected data points in each fold. The algorithm is trained using the K-1 fold and tested using the data of the remaining fold, which is called the validation set. The process is iterated K times until all the folds are used for testing the performance of the algorithm. This method permits averaging the MSE outcomes across various test sets, which can be particularly advantageous when comparing different algorithms. Employing various folds prevents the exclusive evaluation of the algorithm's performance on a unique test set, which could lead to biased selections susceptible to test overfitting [37].

Figure 2.10 shows an example of the working principle of the K-fold cross-validation with five folds, where the dark gray regions represent the validation set and the remaining light gray data illustrate the training set. Each line corresponds to a version of the method trained and evaluated using not overlapping random data from the training set, and the resulting MSE is the average of the MSEs.

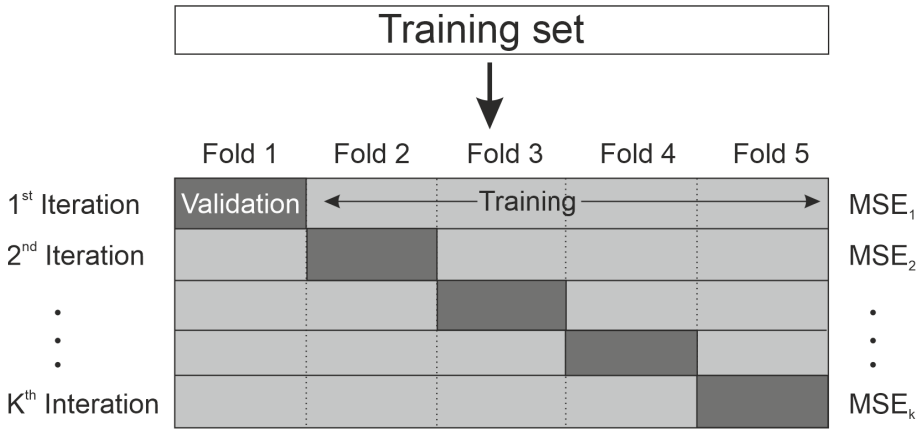


Figure 2.10: Example of a K-fold cross-validation with K=5.
Source: Adapted from [27].

K-fold cross-validation has computational advantages when compared to other methods such as leave-one-out. This method uses the same principle of the K-fold, however it removes only one data point for the validation set. This results in the necessity to train the algorithm for all data points of the training set [27].

The K-fold cross-validation can also be used to choose the best hyperparameters of the model. For instance, in regularized logistic regression, it can compare between

models using different λ values, to select the one that performs better on average, before evaluating this model in the final test set. The same can be done using the parameters of other models, such as the quantity of trees in random forest [37].

In classification tasks, a method that facilitates a visual understanding of the outcomes generated by the algorithms is known as a confusion matrix. This matrix highlights the classes in which the algorithm committed the most confusion. Within the matrix, the ground true labels are arranged along the rows, while the predicted labels are arranged along the columns. Thus, the diagonal of the matrix corresponds to the classes that were correctly predicted, and the rest of the matrix shows the miss classification predictions of the model. Therefore it is possible to visualize which classes produced the highest confusion of the model, which can assist the data interpretation [27].

An example of a confusion matrix for a problem with three balanced classes, and ten observations of each class, is shown in Figure 2.11. It is possible to observe the correct predicted classes in the diagonal of the matrix, as well as the miss classified classes in white.

		Predicted Classes			
True Classes	Classes	Label A	Label B	Label C	Total
	Label A	5	2	3	10
	Label B	2	7	1	10
	Label C	1	0	9	10
	Total	8	9	13	30

Figure 2.11: Example of a confusion matrix.
Source: Adapted from [21].

Different metrics can be computed from the confusion matrix to assess the model's performance. Accuracy is one of the most commonly used metrics. It represents the ratio of correct predictions to the total number of predictions obtained. Therefore it provides the probability that the model's prediction is correct [21]. From the example case shown in Figure 2.11 the accuracy can be calculated as shown in Equation 2.9, where seventy percent of the predictions were performed correctly.

$$Accuracy = \frac{CorrectPredictions}{TotalNr.Predictions} = \frac{5+7+9}{30} = 0.7 = 70\% \tag{2.9}$$

3 State of the Art

The field of multispectral LiDAR is emerging and gaining attention from the scientific community in recent years. Due to their capability to remotely sense different environmental variables in parallel, they can acquire huge amounts of data in a short period of time. To process this data, machine learning algorithms have been majorly applied. Their choice is a subject of current research, given the inherent properties of the data [51].

In order to acquire environmental information, these sensors combine techniques studied separately by various scientific fields. For instance, the analysis of the LiDAR point cloud, where researchers have been using concepts of feature selection to identify the most relevant spatial features that improve object detection [33]. A second example is the field of multispectral imaging, which has been studied by many authors over the years in applications such as vegetation species [35], plastic varieties [5], and waste materials [36] classification. The theory behind multispectral cameras is similar to multispectral LiDAR since they measure the intensity of the light passively reflected in various wavelengths. The same machine learning approaches used in this field can be mapped to the data from multispectral LiDARs [52].

In this domain, a study recently presented in [48], reinforces the lately increased interest by the scientific community. They extensively reviewed 799 sources in the literature and observed the importance of the use of feature selection, also called wavelength selection, in extracting the most informative spectral wavelengths to improve the performance of the ML models. They highlight the lack of agreement in the community for choosing a specific feature selection model. According to them, this choice is highly dependent on the application. Since distinct models can possess an elevated number of hyperparameters, they may be difficult to compare. The authors also state that models that select a high number of wavelengths are not feasible for real-time applications, as the performance of the processing algorithm is dependent on the amount of data available.

On the other hand, another field of study can be reached when dealing with the raw spectral information of a single LiDAR data point. The spectrum of this point is represented by a vector with reflectance intensities for distinct wavelength channels. This is the same principle of a spectroscopy measurement [15]. Hence, the techniques applied in two well-known scientific fields that use machine learning to process this spectrum in the NIR region, spectroscopy [16], and chemometrics [40], can be borrowed to the analysis of multispectral LiDAR data [52].

To this end, one may look at a review of the use of machine learning for NIR spectroscopy presented by the authors in [59]. They highlight the importance of

feature selection to avoid the utilization of redundant spectral data. The authors also discuss the two ML architectures most commonly applied in this field, deep neural networks and feature selection followed by traditional ML algorithms to classify a wide range of materials. Despite the fact that deep architectures have been gaining attention recently, they require a huge amount of data to achieve satisfactory performance. Additionally, they suggest pre-processing the NIR spectrum, with normalization of the spectral data, before executing the ML model. This results in a better performance. Finally, the authors recommend further studies to improve the efficiency of models so that they can be incorporated into portable devices.

Interestingly, the works in [48] and [59] show similar results. They agree on the importance and popularity of the feature selection methods for the analysis of spectral data. In addition, some ML algorithms are being applied in both fields of study, such as the embedded feature selection methods of LASSO and random forest. The authors also consent, that wrapper methods are the most common type of feature selection used in the literature. They further expose the importance of improving the model's performance for the application in embedded systems, by reducing the amount of data through wavelength selection. The conclusions of these researches influenced the choice of the algorithms studied in this work, due to the clear similarities of the data from multispectral LiDARs with both research fields. However, multispectral LiDAR data has certain particularities that, unfortunately, have not been discussed by these authors.

In this case, the study in [30] provides an overview of multispectral LiDARs for terrestrial applications. First of all, the authors argue for the lack of available hardware for acquiring 3D multispectral data and reinforce the necessity for studies of methods focused on this data analysis. The authors also expose the challenges that appear when dealing with hardware development, such as the beam alignment of multilaser systems and the sensitivity of the detectors to noise. Furthermore, they point out the problem of processing a high amount of data when the number of spectral channels increases. They also affirm that future techniques should optimize the processing, storage, and analysis of these data, mainly to deal with real-time applications. In addition, the authors expose a necessity for studies that deal with the challenges of data processing from discrete wavelengths laser systems, since these systems can fulfill eye safety requirements easier than systems based on supercontinuum lasers.

The conclusions presented in the three different research fields show the recent effort of the scientific community toward the improvement of multispectral LiDAR systems for use in real-time and embedded domains. They reinforce the motivation of this work in decreasing data amount to increase the performance of the system. Moreover, the challenges in the hardware development of this new technology open the opportunity for the investigation of techniques that can facilitate the analysis of this new data type.

One advantage of multispectral point cloud data is the possibility of acquiring spectral and spatial information of the environment at each measurement. These two types of features were investigated by the authors in [52]. They used multi-

spectral laser measurements in the red, green, and blue channels, to evaluate the influence of the features in the classification of 14 objects. Random forest was used to perform feature selection and classification of the most important features previously engineered from the spectral and spatial measurements. The authors conclude that spectral features had a higher contribution to improving the classification accuracy than spatial features. Although their research adds strong contributions to the evolution of multispectral LiDAR systems, they do not clarify the choice of the wavelengths used in their analysis. Consequently, the chance that the number of spectral measurements influences the classification performance must be investigated.

The authors in [15] address this issue by evaluating the performance of multispectral LiDAR measurements for classifying seven distinct types of ore. They measured a single point in each ore sample with 17 wavelengths in the VIS and SWIR regions of the spectrum. They also propose a technique using feature contribution degree (FCD) for identifying a vector of wavelengths that contribute most to the classification accuracy. This method is based on the number of spectral measurements variance at each wavelength. To evaluate the proposed method, they applied multiple support vector machines (SVM) to classify the spectral data. The authors concluded that at least seven wavelengths were needed to achieve satisfactory performance, and vectors with less than five wavelengths achieved accuracy values below 50%. Even though they could achieve a high performance in classification, they do not clarify the choice of the 17 wavelengths used in their work and they do not compare the results with other feature selection techniques.

A second work in the analysis of the spectrum of a single data point, evaluated the feasibility of the use of a hyperspectral LiDAR in the autonomous vehicles sphere [54]. They used random forest to classify ten classes of materials found in the road environment. For the analysis, they proposed a LiDAR system containing 30 wavelength channels in the spectral region from 1200 to 1570 nm. The authors used 5-fold cross-validation to investigate the influence of binning the spectrum intensities into spectral channels in the accuracy of the model. Additionally, they also evaluated the influence of photon accumulation at different numbers of frames. Even though the focus of their work was on the development of the hardware for a single photon multispectral LiDAR system, they could reinforce the relevance of the increased spectral resolution to improve classification accuracy. Through the use of random forest, they could classify materials with very distinct spectral footprints with up to 90% accuracy. Since the authors focused on acquiring measurements from the whole spectral range at once, they do not evaluate the importance of the spectral channels calculated intrinsically by the random forest model. This opens an opportunity for further research into the intrinsic capabilities of the random forest model when applied to this data type.

One step further in the direction of wavelength selection, considering the feature importance ranking of the random forest algorithm, is presented in [14]. They applied a supercontinuum laser source, combined with a 355 nm monochromatic laser, to acquire spectral reflectance and fluorescence of different types of leaves. Then, the authors proposed a combination of random forest Gini importance metric

with adaptative band selection to evaluate the relation between the classification accuracy, by random forest, and different wavelength bands. In the end, they found that the introduction of fluorescence signal has improved classification accuracy, with the highest performance using only seven wavelengths. Despite the fact that they compare the result of random forest with five wavelength bands, the authors do not present the accuracy that this classifier would achieve when the whole spectrum data is used as a comparison metric. Nevertheless, their work gives a good basis for a detailed analysis of wavelength selection techniques in multispectral LiDAR point clouds.

The authors in [11] also used the random forest for the classification of tree species in multispectral LiDAR measurements. The researchers employed a commercially available airborne multispectral LiDAR system with three discrete wavelength channels, at 1550, 1064, and 532 nm, to evaluate the relation between the scan angle of the laser and the intensities of the spectral channels. Although their primary goal was to study the influence of the scanning angle on the results, they also evaluated the influence of the wavelength intensities on the accuracy of the RF classifier. Therefore, they manually selected all possible combinations of the three wavelength channels, and they found that using the three wavelength intensities resulted in higher accuracy than other combinations of channels for classifying six tree species. The proposed research shows an interest in multispectral systems with discrete wavelengths and indicates a possible relation between the accuracy and the number of available wavelength channels. However, the optimal number of spectral points is not discussed due to the hardware limitations and still has to be investigated.

This problem was partially covered in [50]. The researchers propose a method for selecting wavelengths from four different applications. They used the variance at each wavelength to select the most informative spectral channels to classify samples of rocks, ore, plant leaves, and wood. Their method was able to reduce the amount of spectral information from 91 channels, in the spectral region of 650 to 1100nm, down to two channels depending on the data to be classified. They have shown that the number of selected wavelengths varies according to the application. Combining their method with SVM and Naïve Bayes they could perfectly classify the samples inside each material class. In addition, the authors present the correlation contained in the spectral data, and the importance of calibration of reflected light to acquire the reflectance of the materials. Unfortunately, they do not discuss the results that the model would achieve when classifying the samples into different material categories. Moreover, they do not compare the results from the model with other methods, nor consider the SWIR region of the spectrum in their evaluation.

Following a similar trajectory, scientists have explored the potential of utilizing hyperspectral LiDAR data in the VIS region to recreate the colors of objects. In a study presented by [12], a two-step approach was employed to estimate the three most relevant wavelengths for color reconstruction in a LiDAR measurement. Initially, measurements of paper cards in various colors were used to select the optimal optical bands through a combination of principal component analysis with spectral correlation measures. Subsequently, the classification performance of SVM and

Naïve Bayes was evaluated using the selected wavelengths in measurements of real objects. Even though the main objective of the research was to identify only three wavelengths for color reconstruction, given the limitation of human eye perception, their approach could be extended to other applications where data acquisition for spectral analysis is challenging. Since the spectrum of the objects can also be measured by a spectrometer, a spectral dataset can be created and analyzed by algorithms prior to actual multispectral LiDAR measurements, considering the current hardware limitations and deficiencies.

Based on the analysis of research directions presented in this section, this work proposes a novel model for examining multispectral LiDAR data. The use of a spectrometer to acquire spectral information should allow the evaluation of the performance of the machine learning algorithms prior to their application in real multispectral LiDAR measurements. Thus, this work aims to acquire spectral data from different categories of materials and apply two embedded feature selection methods, which proved to be an important topic of interest in the literature, to identify the most relevant wavelengths that allow samples to be distinguished into distinct material classes. Besides, a wrapper method will also be evaluated, due to its popularity in the spectroscopy and multispectral image fields. Finally, the results of the evaluation should guide the choice of the wavelengths used by the measurements in a multispectral LiDAR under controlled laboratory conditions. The results of this work can contribute to the research in the development of multispectral LiDAR systems, by addressing the evaluation of techniques that may allow a reduction in hardware costs before its implementation in distinct application domains.

4 Methodology

The architecture of the model proposed in this work consists of three main steps. It starts with the creation of a dataset containing spectral data from four classes of materials acquired using a calibrated spectrometer. The second step is the creation and evaluation of two embedded feature selection algorithms and one wrapper method in the created dataset for two different scenarios, before and after the wavelength selection. The final step is the selection of one model, and a subset of wavelengths, for the analysis of their performance in data from a multispectral LiDAR demonstrator. The overall block diagram of the methodology of this work is presented in Figure 4.1 where the colors highlight the three main steps described.

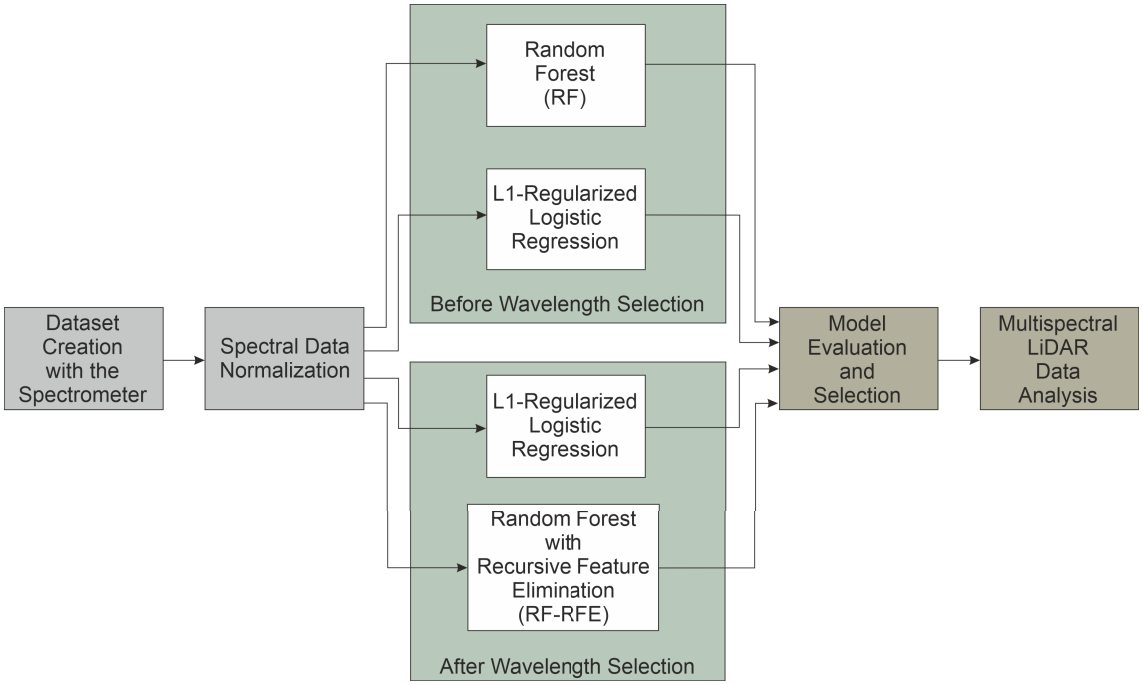


Figure 4.1: Block diagram of the proposed model.

The dataset created for the evaluation of this work consists of four different classes of materials commonly found in urban street environments, they are: organics, metals, fabrics, and plastics. The selection of these categories was based on the intention to investigate the complexity of classification of materials found on real world scenarios. Besides, the choice was influenced by the approach used in similar works [54][16] with the ambition to increase the spectral fingerprint diversity of the dataset. The

amount of spectral measurements for the creation of the dataset was also chosen based on the results obtained by similar works. The authors in [16], showed that a dataset of 5000 spectra from household materials was enough to evaluate the performance of a deep learning classification model. On the other hand, in [54], the authors argue that 300 measurements was sufficient for a random forest model to classify 10 different materials from a road environment. Therefore, the dataset created in this work contains 2000 spectral measurements, to maintain a balance between the dataset sizes presented in the literature, since less data-intensive methodologies were evaluated. In addition, to increase the spatial and spectral variance of the data acquired, 5 samples of each class were measured in various spatial positions. As stated in [59], the pre-processing of the spectral data is an important step, and should be performed before the application of the classification models on the dataset, to eliminate the influence of the offset and improve the performance of the algorithms. Therefore, each spectrum of the dataset is normalized to guarantee zero average and unit variance.

The machine learning algorithms chosen for the analysis of this work are: random forest, logistic regression with L1-regularization, and recursive feature elemination. Random forest was selected due to its popularity in multispectral LiDAR data evaluation and its capacity to attribute importance to the features [54][52][14] [11]. On the other hand, the L1-regularized logistic regression was chosen, given popularity of the L1-regularization, also known as lasso, for wavelength selection of NIR spectrum [59] and multispectral images [48]. This choice allows the comparison of two well-known classifiers with distinct working principles and the evaluation of the feasibility of their application in multispectral LiDAR data.

The recursive feature elimination wrapper was also chosen for the analysis of this work, given the popularity of wrapper methods in multispectral wavelength selection field [48] and the possibility of determining a specific number of wavelengths as a hyperparameter [23]. Therefore, it allows to access the performance in accuracy of the random forest in different spectral resolution domains. Furthermore, the choice of using a wrapper method was made given the intrinsic characteristics of the RF model in performing feature selection. This classifier does not remove the unused features, instead it only attributes an importance for each feature used [8]. Hence, the combination with the recursive feature elimination wrapper (RF-RFE) allows an automatic selection and evaluation of the number of wavelengths utilized, as the RFE can reduce the number of wavelengths in defined steps. To select one version of the RF model to be incorporated into the wrapper, an optimization of the hyperparameters has to be performed. In this work, the hyperparameter chosen for optimization is the forest size. The accuracy performance of the random forest is not determined by size of the forest, in fact, large forests introduce a waste of computational resources [7]. Therefore, the optimal relation between the random forest size and its performance is to be found before the application of the recursive feature elimination.

Afterwards the models are evaluated in two different scenarios. In the first scenario, the entire spectral resolution of the samples available in the dataset are

used to train and compare the random forest and the logistic regression with L1-regularization. This scenario allows the optimization of the size of the random forest. The utilization of the entire spectral resolution available make the results of this evaluation to be considered the benchmark for this work.

In the second scenario, the feature selection capabilities of the algorithms are analyzed. Therefore, versions of the L1-regularized logistic regression and the RF-RFE model are created in different spectral resolution domains. This permits the evaluation of the relation between the accuracy of the model with the amount of spectral information used.

Later, a comparison of the performance of the models created in both scenarios is realized. The models are compared for the number of wavelengths used, the accuracy, the time to predict one spectral sample, and the size occupied in memory. These metrics are chosen given their valuable amount of information that can contribute for the development of applications in diverse domains. For instance, the prediction time of a single spectral may be crucial for applications where real time response is necessary. The memory size evaluation might give insights for applications in embedded systems domain with limited amount of resources. Besides, these applications can also benefit from the number of wavelengths used, which can drastically reduce the amount of data needed for these systems [48][59]. Furthermore, the accuracy gives a general overview of the quality of the classification through the percentage of corrected predictions in relation to the total number of predictions.

The results of the evaluation of the metrics are utilized to guide the selection of a specific version of the model for analysis using multispectral LiDAR data. This selection process considers both the capabilities of the demonstrator and the performance of the models in the comparison described earlier.

In the end, a balanced compromise is achieved between the number of wavelengths used by the classification model and its accuracy. The wavelengths identified through this analysis are subsequently measured in an example scenario using a multispectral LiDAR demonstrator to access the model's performance on real data. This approach aims to provide a detailed investigation into the potential of using spectroscopy measurements to guide the development of discrete multispectral LiDAR systems.

5 Implementation

This Chapter describes the working principle of the equipment and tools utilized for data acquisition and model development in this study. It starts explaining the functionalities of the spectrometer used for the spectral data acquisiton, followed by the Python framework used for the machine learning models training and evaluation, and finishes with the description of the multispectral LiDAR demonstrator used for the generation of the multispectral LiDAR point cloud.

5.1 FT-IR Spectrometer

One of the most important steps of the project is the creation of the materials spectral dataset for the training of the machine learning models. To obtain reliable spectral measurements at a fine resolution, a calibrated commercially available spectrometer was utilized. This equipment is a FT-IR spectrometer from the company Bruker [10]. It is composed of the Bruker Vertex 70 spectrometer coupled to a Bruker Hyperion 3000 microscope imaging system. This configuration allows an automatic scanning of the sample's surface area, which facilitates the acquisition of large amounts of data. A view of the coupled equipment used for the spectrum acquisition of the materials is presented in Figure 5.1.

To perform the spectral measurements, the spectrometer generates electromagnetic waves through a tungsten lamp. These waves travel through the microscope's optical path to illuminate one spot on the surface of the sample. Then they are reflected by the sample surface and guided by the optical path to an InGaAs detector, which measures the amplitude of these reflected waves.

The microscope owns a precision stage that performs movements in x, y, and z directions [10]. It allows the system to be configured to measure different coordinates of the sample area. Thus, automatic scanning of the sample surface can be carried out, increasing the spatial diversity of the measured spectrum in the dataset.

The equipment allows the user to choose between several parameters of the hardware, such as the light source, the detector aperture size, and the spectral resolution to adjust the operation in different modes [10]. The configuration applied to acquire the data in this work uses the tungsten lamp as a light source, 6 millimeters for the detector aperture size, and 8 wavenumbers of spectral resolution, to acquire spectral measurements in the range from 781 to 2500 nm. In addition, the equipment performs a differential analysis, against a reference sample during spectrum acquisition, to avoid the influence of the light source and the optical path in the measured spectrum [55].

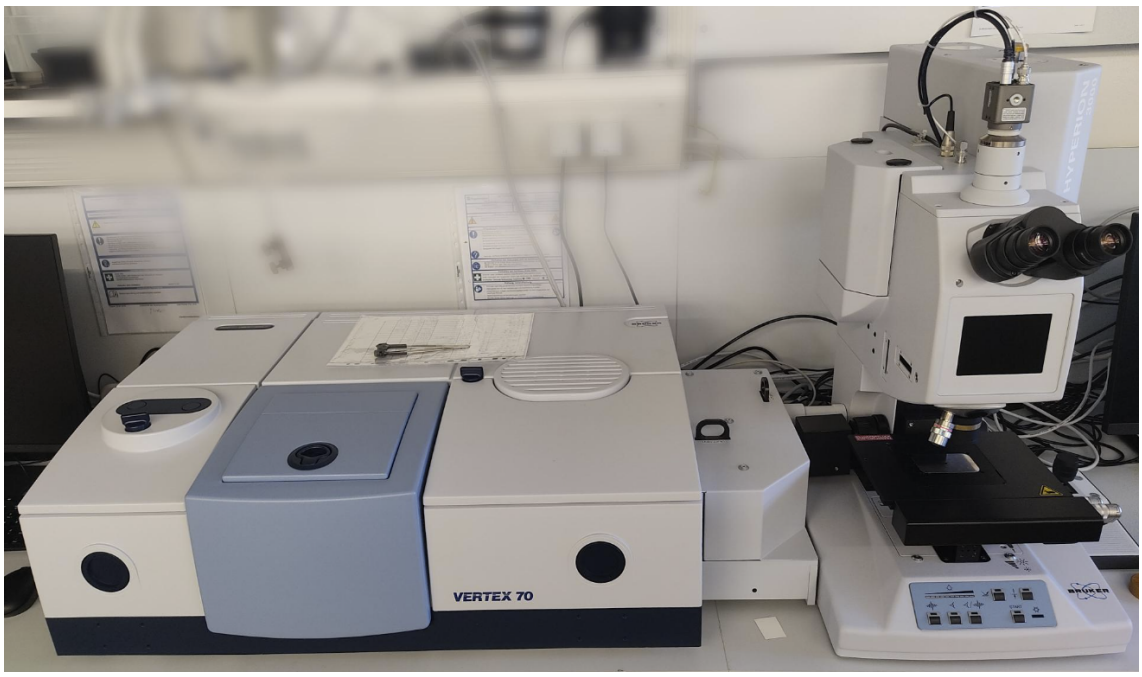


Figure 5.1: Optical setup consisting of a Vertex 70 Spectrometer coupled with a Hyperion 3000 microscope.

5.2 Python Frameworks

The programming language chosen for this work is Python, which has gained significant attention due to its extensive collection of third-party packages and libraries available for public use. The possibility of integrating various packages with different capabilities creates an ideal environment for developing and testing software prototypes in a short period. This has turned Python into the dominating programming language for machine learning development in the last years [62].

In the machine learning context, these third-party packages are called frameworks, and they count with diverse data handling optimization techniques that assist the training processes of the ML models, facilitating the development and decreasing the training and testing time. Typically, these frameworks come with pre-defined mathematical and statistical models, that offer the users the possibility to customize and evaluate the model's hyperparameters according to their application.

From the frameworks available in Python, scikit-learn is widely utilized in the field of data analysis [62]. It can be combined with other data manipulation and visualization packages and libraries, such as Matplotlib, Numpy, and Pandas to increase the user's control and interpretation of the results.

To develop Python software, it is necessary to use an interface. Python is an interpretative programming language. Thus, it allows the execution of the software lines independently. Jupyter Notebook is a famous interactive computational environment used for the development of machine learning models since it permits the

division of the software into disassociated cells that can be executed separately. This allows the user to write software and text into the same interface, which increases the organization and interpretability of the code. In addition, the use of cells facilitates the debugging of the software, as the stepwise execution enables the visualization of the data in between software lines [32]. The cells run on top of the Ipython that works as a kernel to allow the development of interactive Python software on the web application [42].

The high number of third-party packages, demands the development environment to be integrated into a package manager that manipulates the execution of different libraries in an organized manner. Therefore, the Anaconda was selected to be used in this project. It is a package manager that contains all the most common packages needed for the development of machine learning models. It decreases the development time by handling the integration of the libraries into the Jupyter Notebook environment [2].

In addition, Pandas was employed to import and manipulate datasets inside the Python environment. It is a data structure library, that consists of data handling functions that facilitate the manipulation of different data types and the access of specific positions of the dataset. Furthermore, it standardizes different formats of data into data frames, making it an ideal option to deal with spectral measurements coming from different measurement equipment [38].

To illustrate the data distribution, and the results obtained for the ML models, Matplotlib is one of the most common Python libraries applied in this field. This library is specific for graphical representations, and it is capable of managing commands for generating visualizations of the data through plot functions. It can be combined with other data manipulation tools, such as NumPy and Pandas, enabling interactive visualization of the results within the environment [25].

The main library selected to assist the development of the machine learning models was Scikit-learn. It contains a basic structure of the models selected for the analysis in this work, with the option of diverse hyperparameters tunning for each model. Furthermore, scikit-learn also incorporates metrics for the evaluation and comparison of models, such as the K-Fold cross-validation. These metrics can be integrated during the model's development helping to compare different versions. This comparison provides a detailed analysis of the influence of the parameters in the performance of the model [41].

As an example for the random forest model, scikit-learn counts with an implementation that allows the selection of hyperparameters such as the number of trees, the maximum depth of each tree, and the split metric. Furthermore, it allows access to the feature's importance calculated after the model's training.

Another machine learning technique present in scikit-learn is logistic regression. The library allows the user to select the regularization technique, such as the lasso norm, and the regularization strength parameter. In addition, the library makes available options for optimization of the algorithm's training time, such as the selection of the number of cores used by the computer during the model's training [41].

Finally, scikit-learn also integrates two implementations of the recursive feature elimination wrapper method. The first one permits the user to define the desired number of features selected by the model. The second uses a model with an integrated cross-validation technique to find the optimal number of features in the used dataset. For this case, the library allows the user to choose the stopping criteria for the cross-validation, along with the number of folds utilized by the evaluation algorithm. In both cases, some hyperparameters are available for tuning, such as the number of features to remove during each interaction, and the number of processors used by the computer [41].

To train and evaluate the models in this project a computer equipped with a 2x Intel Xenon E5-2650 v4 processor boasting 12 Cores each, running up to 2.2 GHz, with a 30 Mb L3 Cache CPU was utilized. Additionally, the computer features 256 GB (8x 32GB) DDR4 RAM and 240GB SSD. With this computational setup, the training times of the algorithms could be accelerated, and sufficient memory capacity was available to effectively train and store different models.

5.3 Multispectral LiDAR Demonstrator

The capabilities of a specific version of the models developed in this work were evaluated in real multispectral LiDAR data. This data is acquired by a prototype of a multispectral LiDAR system developed by Fraunhofer ENAS in cooperation with TU Chemnitz. The prototype uses a SuperK EXTREME supercontinuum white light laser source, from NKT Photonics, combined with a composition of a SuperK SELECT acousto-optical tunable filter (AOTF) with SuperK CONNECT and FD7/FD8 optical fibers to generate light pulses in specific wavelengths. To scan the environment, an optical system containing a moving mirror, Galvoscanner GVS002 from Thorlabs, was utilized to modify the direction of the laser beam. The prototype operates in two modes that vary according to the data to be acquired.

The first operation mode acquires the spectral data through a grating spectrometer. The setup consists of a Shamrock 303i with an iDUS 1.7µm InGaAs line detector from Andor, combined with a beam splitter, model BSW29R from Thorlabs. The basic working principle of the hardware system starts with the emission of the white light pulse in the range from 450 nm to 1700 nm by the laser source. This light pulse travels to the acousto-optical tunable filter, where it is filtered to a specific wavelength selected by the user. Subsequently, it travels through the optical fiber to the optical bench. In the optical bench, the scan mirror deflects the laser beam at different angles to illuminate point-by-point the target object at different wavelengths, creating a 2D image of the target. Finally, the reflected light is guided back to the entrance slit of the grating spectrometer by the beam splitter. An example of the test bench, containing the optical path of the laser pulse in the spectrum acquisition operation mode, is shown in Figure 5.2.

The position of the moving mirror is determined by a LabView software. The software is responsible for changing the direction of the light pulse to hit different

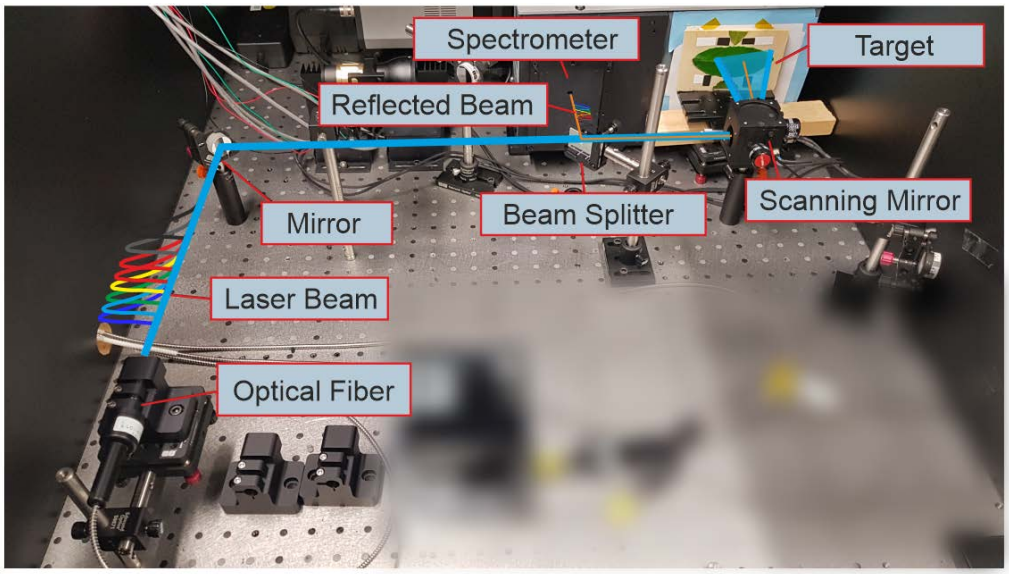


Figure 5.2: Multispectral LiDAR demonstrator in spectral measurement mode.

spatial positions of the target, creating the pixels of an image. First, the software sets a voltage to the scanning mirror to control the direction of deflection of the laser beam, via a NI-USB 6343 multifunctional I/O device, from National Instruments. Then, the desired wavelength is selected and the laser pulse is emitted. After a few milliseconds, a trigger signal is sent to the spectrometer, through the I/O device. Consequently, the spectrum of the point is recorded. Afterwards, the wavelength is changed and measured again until the entire spectrum of the specific point is acquired. The process is iterated for all the points, which can be seen as pixels, of the image. Once the measurement is finished, a measurement file is created containing all the measured wavelengths at the entire image following the configured scanning pattern.

The measurement file is created using the Andor's proprietary software. It contains the amplitude of light acquired by the spectrometer at each wavelength measured for each spatial point. The data is organized in columns separated by a tabulator, and exported as a .txt file. The first column contains the position of the spectrometer sensor, while the subsequent columns contain the amount of photons acquired at the specified positions. Therefore the data has to be filtered and preprocessed before the multispectral images can be created. The LabView software has the logic to control the hardware of the test bench in both operation modes. It also allows the user to selected the mode under test and configure the system parameters.

The second operation mode, measures the distance of the object through a Single-Photon-Counting-Modules (SPCM), model SPCM-AQRH-14 from Excelitas, combined with a Time-Correlated Single Photon Counting (TCSPC), model MultiHarp 150 from the company PicoQuant. This setup is used to measure the ToF of the

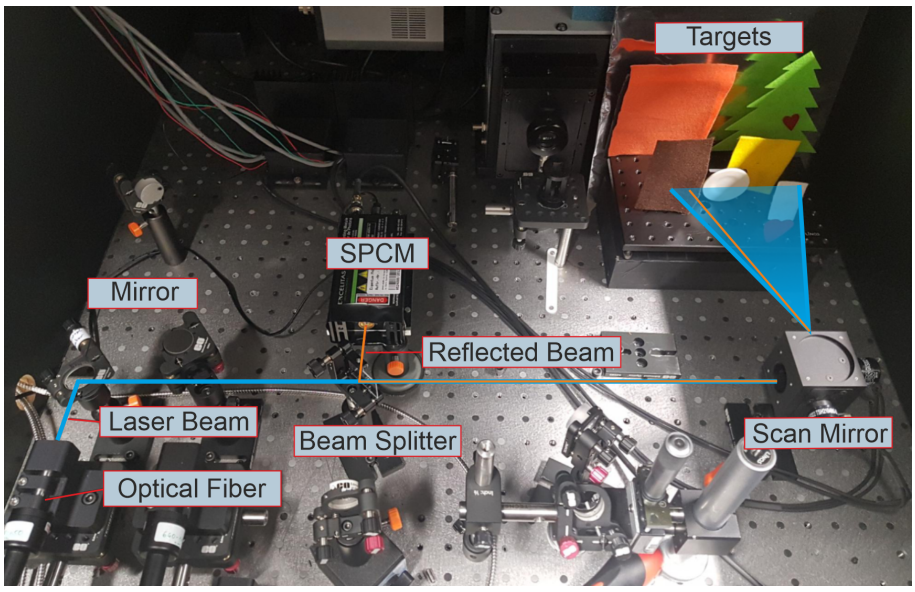


Figure 5.3: Multispectral LiDAR demonstrator in distance measurement mode.

laser pulse with a maximum resolution of 10 ps. When the laser pulse is emitted a trigger signal is sent to the TCSPC to count the interval between the emission of the laser pulse and the acquisition of the first photon by the SPCM. This ToF measurement can be later applied to calculate the distance of the reflecting object. Similarly to the first mode, a beam splitter is used to guide the reflected light to the entrance of the SPCM. In addition, the LabView software controls the angle of the moving mirror, to scan the position of all target points. An example of the prototype in the distance acquisition mode with the illustration of the light path is shown in Figure 5.3.

The necessity of using two measurement modes come from the limitation of the SPCM in the SWIR range. Since this equipment only measures in the VIS spectrum, the system has to be modified for using a specific wavelength during the distance measurements. The data acquired during this measurement represents a histogram of the amount of photons received during a pre-defined time window. The resulting data is saved as a .txt file separeted by a tabulator where the first column represents the time in nanoseconds and the second column represents the histogram of the amount of photons received in the specified time. The time point with the highest amount of photons is used as the ToF measurement. This process is iterated for all the points of the image. Therefore the files acquired have to be pre-processed, the data grouped and combined with the spectral measurements of the spectral mode to create the multispectral 3D point cloud of the environment.

The system allows the configuration of diverse parameters of the equipment connected that can improve the data quality and speed up the process, such as the exposure time of the spectrometer, the laser power, the triggering time, and the grating aperture of the spectrometer.

6 Development

6.1 Dataset Creation

6.1.1 Data Acquisition

The development started with the acquisition of the spectral dataset using the Bruker spectrometer. Therefore, four classes of materials were selected, each class containing 5 distinct samples. The classes divide the materials into metals, organics, fabrics, and plastics. The samples used for each class are presented in Figure 6.1. They were mapped to their respective spectral fingerprint through an enumeration that increases the interpretability of the measurements. In addition, these samples were chosen to allow the positioning of a flat surface on the microscope stage, perpendicular to the direction of the NIR light beam. This decreases the influence of the incidence angle at the material surface and encourages the acquisition of the highest amplitude of the spectral measurements [16].

As can be seen in Figure 6.1(a), one of the samples selected for the metal class is composed of rusty iron. This sample was modified removing half of the rust from its surface. Hence, half of the sample surface was added to the dataset as clean iron, and the other half as rusty iron, to include the influence of the rust in the classification data.

For the acquisition of the data, the spectrometer was configured to operate in the reflection mode and to use the tungsten lamp as the light source, to emit EM radiation in the NIR spectrum. The aperture of the detector was defined as 6 millimeters. This aperture controls the amplitude of the acquired reflected light, and helps to reduce environmental noise in the measured spectrum.

The spatial differences of the measurements were obtained by configuring the microscope to operate in the video mode. This configuration allows the definition of cartesian coordinates to position the microscope stage. Thus, when the sample was positioned on top of the stage, it was possible to automatically move the platform and measure different locations on the surface of the same sample. The control of the movements is performed automatically by Bruker's proprietary software.

The first step for the acquisition of spectral data from one sample of material was the manual positioning of the VIS objective in the microscope. After that, a specific area of the sample surface was defined for the analysis. Subsequently, the microscope software moved the stage along the x and y axes and captured sequential images that covered the entire area of interest. This step allowed a visual inspection of the sample surface and guaranteed that the sample was positioned in the focus of the

(a) Metals.

(b) Fabrics.

(c) Plastics.

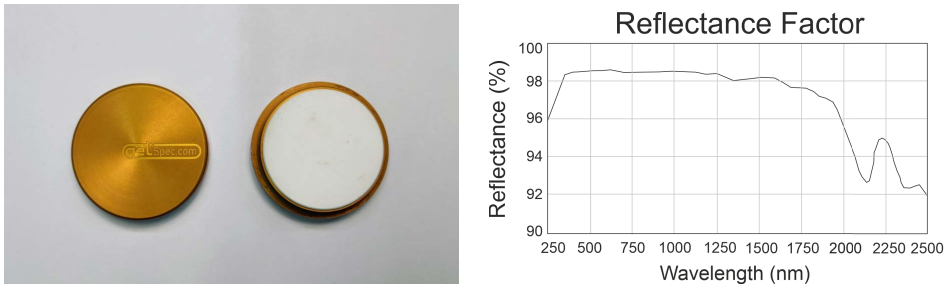
(d) Organics.

Figure 6.1: Image of the material samples used for spectral acquisition.

optical system. Afterward, the objective lenses were manually switched to the IR, and a reference measurement was conducted for the differential spectral processing.

The choice of the reference sample took into account the reflection properties of the material. These properties must match the reflection amplitude range of the samples under analysis. The use of a reference with high reflection properties, like gold, would require a high dynamic measurement range, resulting in a loss of resolution and information when compared against samples with low-reflection properties. Since the material samples used in this study generally exhibit low reflectivity, due to the roughness of their surface, a ceramic material was a suitable choice to be used as a reference in these experiments. Hence, the reference sample getSpec STD 5101 CIN [19], which is composed of tile with diffuse PTFE, was selected to be applied in this work due to its spectral response characteristics. This sample presents a high and linear reflectivity around 98% for the spectral range between 350 and 1800 nm, as can be shown in Figure 6.2(b). The reference selected and its spectral reflectance are presented in Figure 6.2.

After the reference spectrum acquisition, Bruker's software waited for the user to configure the position of the spectral measurements on the material surface. In this project, a grid arrangement with 10x10 measurement points was created for the analysis. This incorporated spectral measurements of different locations into the



(a) Example of the getSpec STD 5101 CIN reference sample.

(b) Spectral reflectance of the getSpec STD 5101 CIN.

Figure 6.2: Reference Sample.
Source: Adapted from [19].

dataset. Subsequently, the software moved the microscope stage along the configured cartesian directions to align each specified point with the center of the optical path. This process was repeated until all defined points were automatically measured.

This configuration measured 100 spectral points in the grid arrangement. The grid was configured based on the previous VIS image of the defined surface region of the sample. The spectrum resulting for each data point measured corresponds to the average of 32 spectral measurements.

The configured points were measured sequentially in a column-wise manner. The measurements were performed from the bottom left to the top right of the image. After all points in the image were measured, the software allowed their exportation in a DPT file format. This file organizes the measured data in a table format, where they are separated by a tabulator. The wavelengths were organized in rows, and the measured points in columns. The first column contained the wavelengths measured in wavenumber. All the following columns contained the intensity of the reflected light for each measured point at the corresponding wavelength. A flowchart of the process of data acquisition for one material sample is shown in Figure 6.3.

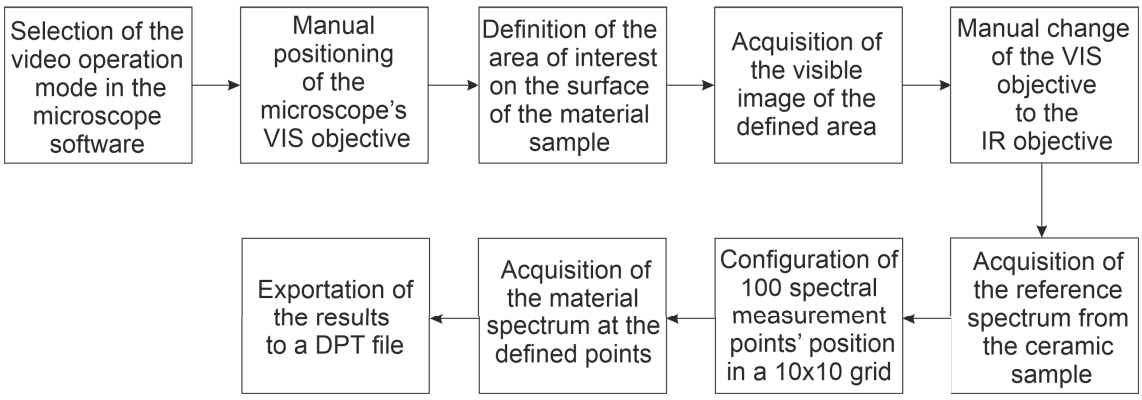


Figure 6.3: Flowchart of the spectrum acquisition process for one material sample.

This process was repeated for all the samples presented in Figure 6.1. An example of the raw spectrum of a fabric sample acquired using the automatic mode of the FT-IR spectrometer is shown in Figure 6.4.

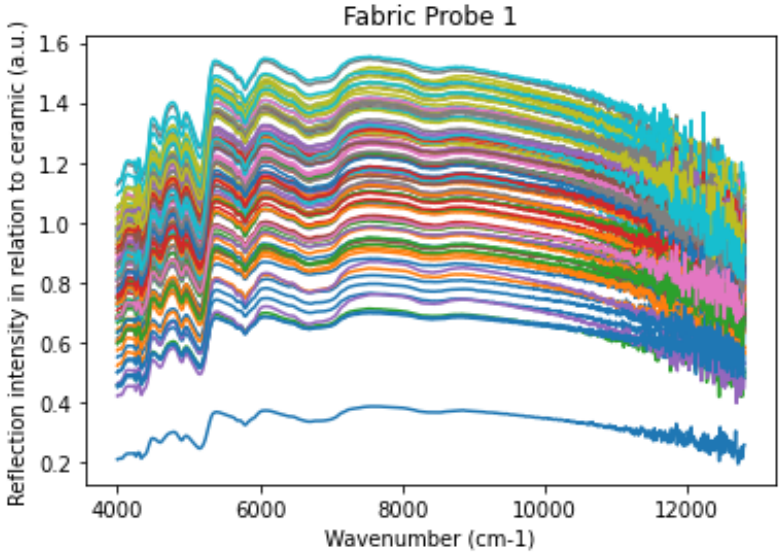


Figure 6.4: Example of the spectrum acquired for 100 points of a fabric sample.

From the example presented, it is possible to observe that the raw measurements of the spectrometer contain noisy values for data in high wavenumbers. This noise is intrinsic to the measurement equipment, mainly caused by factors such as the decrease in the efficiency of the tungsten lamp, and the InGaAs detector in this region of the spectrum [9]. In addition, the measurements present a large standard deviation. There are two main factors that interfere in the standard deviation of the spectral measurements they are the roughness of the surface of the material and the amount of concentration of the substances in the point measured. The first, changes the scattering pattern of the reflected light and the length of the optical path. The second, changes the amplitude of the light reflected by absorbing different amounts according to the concentration [59]. Therefore, to obtain a fair comparison of the data, it is necessary to pre-process the spectrum by removing the noise and the offset to avoid a biased training of the classification algorithms.

6.1.2 Data Pre-Processing

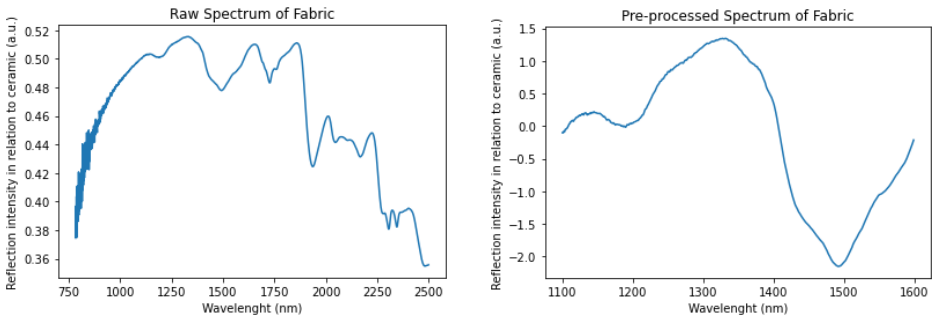
The pre-processing is one important step after the acquisition of the data [59]. The measurements are saved by the Bruker software in a table format (.dpt). Therefore, they need to be imported into the Python environment and converted to a data structure that can be manipulated by third-party packages. This was performed through the functions available in Pandas, which created a data frame from the table of measurements and loaded it in the Python kernel for further processing.

Furthermore, the equipment acquired the spectral data in wavenumber, so the first step of the processing involved converting it from wavenumber to wavelength. This conversion facilitated the data comprehension when evaluating the spectral characteristics visually, as wavelengths are commonly used in the literature [48]. To convert from wavenumbers to wavelengths, it is necessary to calculate the inverse of the value [55]. Thus, Equation 6.1 was applied in each spectrum measured.

$$wavelength = \frac{1}{wavenumber} \tag{6.1}$$

After that, the data had to be pre-processed before training the machine learning models. The first step consisted of removing the noisy region of the spectrum presented in Figure 6.4. Therefore, to obtain machine learning models that could be later applied in the multispectral LiDAR demonstrator, the characteristics of the prototype had to be considered. Thus, the measurement spectral range of the demonstrator was used to avoid training the machine learning models with data that cannot be measured. Since the demonstrator can only measure spectrum from 1100 nm to 1600 nm, the measured data was truncated in this range. So only the spectrum inside these limits was considered for further evaluation.

Finally, a common pre-processing method was applied to the data. There are different functions for pre-processing NIR spectral data, such as the use of the Savitzky-Golay filter, mean removal, and standard variate. Some of the pre-processing techniques modify the characteristics of the spectrum. For instance, the Savitzky-Golay filter smooths the NIR spectrum by removing high-frequency noise and has the drawback of being computationally expensive [59]. Therefore, to preserve the initial characteristics of the data and avoid the influence of offset in the analysis of the machine learning models, the approach used in this work only performed the normalization of each spectrum, by removing the average and converting to unit variance.

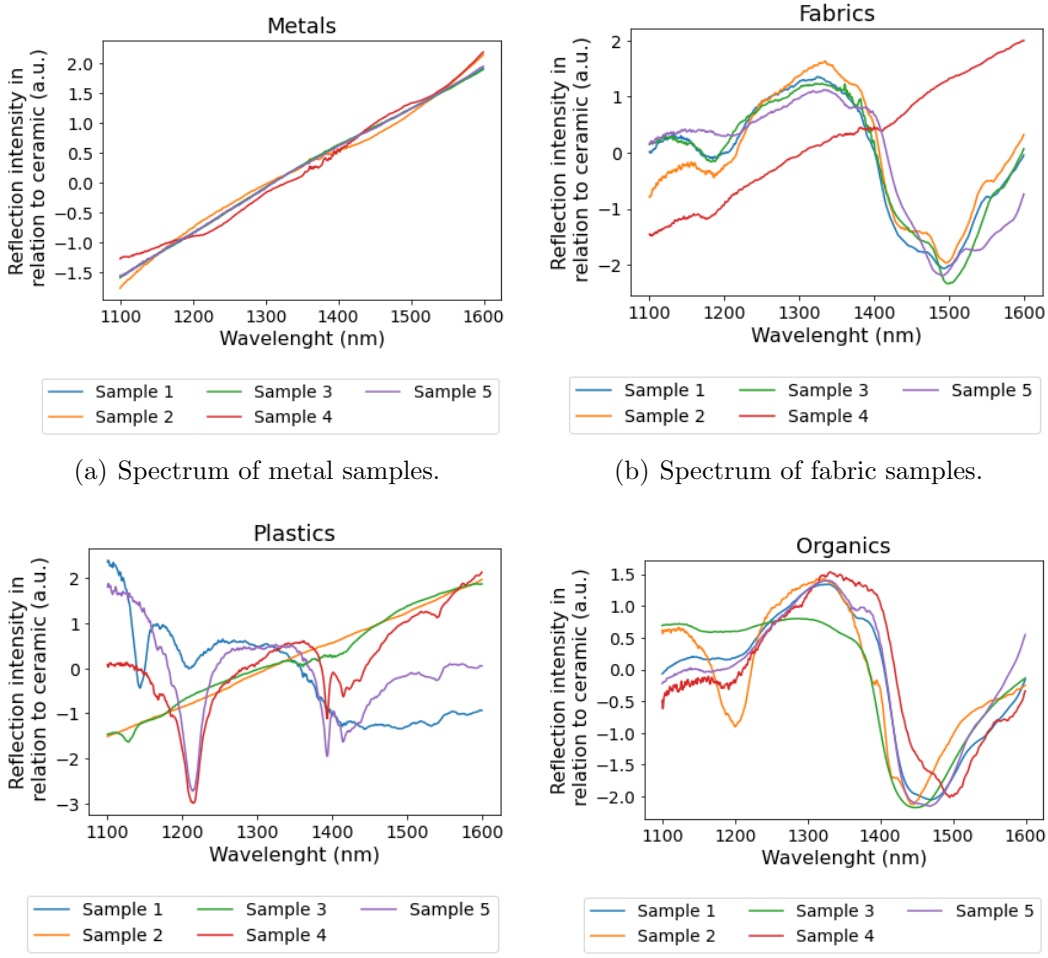


(a) Raw spectrum. (b) Pre-processed spectrum.

Figure 6.5: Comparison of a measured spectrum before and after pre-processing.

An example of the effects of pre-processing spectral data is presented in Figure 6.5. In Figure 6.5(a) an example of the raw spectrum acquired by the spectrometer is

shown. The spectrum was converted to wavelengths to facilitate the interpretation. Figure 6.5(b) shows the same spectrum after the pre-processing steps. As can be observed from the graphs, the spectrum is truncated and has zero mean and unit variance. It can also be observed that a large amount of information is not considered given the limitation of the spectral range of the demonstrator. This opens the opportunity for further studies in different spectral ranges.



(a) Spectrum of metal samples.

(b) Spectrum of fabric samples.

(c) Spectrum of plastic samples.

(d) Spectrum of organic samples.

Figure 6.6: Average spectral fingerprint of each material sample.

After all the spectral data was pre-processed, they were concatenated to create a standard dataset. Since each sample of the material was measured in 100 points and a total of 20 different samples were used, the final dataset contained 2000 measured spectra. Each spectrum has a resolution of 736 wavelengths. Thus, the final dataset is formed by 736 x 2000 measured points. To enhance the visualization of the data, Figure 6.6 shows the average of the spectral measurements of each material sample. The average spectrum is enumerated following the sample distribution presented in

Figure 6.1. The spectral measurements are shown after the pre-processing steps, so the Figure represents an overview of the dataset used for the training of the algorithms.

In Figure 6.6 it is possible to observe patterns that differentiate the classes. These patterns should contain enough information for a model to learn a classification function that differentiates the material based on its spectral fingerprint. However, some of the fingerprints have similar characteristics, such as the shape and the inclination, that could confuse the model. For instance, sample 2 of plastics and sample 4 of fabrics could mislead the algorithm into classifying them as metal due to their similarity with the metal fingerprints. Therefore, the analysis of the dataset is an important step in understanding the behavior of the results of the models.

Before the training and analysis of the machine learning models, the order of the measurements was shuffled to avoid the influence of ordered data patterns in the dataset. Afterwards, the dataset was divided into training and testing sets. The test set is composed of 20% of the size of the dataset. Hence 400 measurements were separated to be used for testing the performance of the model, and 1600 measurements were used for training. Finally, the dataset was prepared for utilization in the analysis of the performance of the models in this work. The first model developed for this analysis was the L1-regularized logistic regression.

6.2 L1-Regularized Logistic Regression

The logistic regression model employed for the development of this study was based on the implementation presented in Scikit-learn [41]. The library contains functions that allow the modification of the hyperparameters of the model, as well as the controlling of the training and evaluation processes. To address the feature selection capability, the model was configured to use the L1-norm as regularization. This mode requires the selection of the regularization strength that forces the weights of the features to zero. This regularization is present in the function through the hyperparameter C. It represents the inverse of the λ value of the LASSO regularization, shown in Equation 2.8. Therefore, setting a lower value of C leads to a model with less features selected.

The goal of this research is to assess the dependency of the accuracy of the model on the number of wavelengths used. To achieve this goal, the hyperparameter C was adjusted to generate models that use different numbers of features. In total 100 different models were analyzed. They were developed to work within a range of 1 to 100 features. The creation of the models started with a value for C that resulted in more than 100 features select, the number 0.4 was found to meet this requirement. Then the number of features selected was evaluated after each training run. If the number of features used was equal to 100, the first model was saved, otherwise, the regularization strength parameter was reduced and the model was trained again. After the first model was saved, the expected number of features was decreased by one and the procedure was repeated. These steps continued until the model using

only one feature was saved.

During the development of the algorithms, it was observed that the relation between the regularization strength and the number of features selected is non-linear. Hence, to obtain models with a decreased number of features in unitary steps a dynamic update of the regularization strength was necessary. Therefore, if the strength parameter resulted in a model with fewer features than expected, the parameter was reverted to its previous value, and the decrement step was adjusted to finely tune the value of C.

In the end, 100 versions of the model were saved. This approach allowed a detailed analysis of the influence of the number of features (in this case, the number of wavelengths) on the classification of materials. Consequently, it covered the analysis of hyperspectral and multispectral system domains due to the wide range of features used.

The configuration applied to train the algorithms performed the one-vs-rest classification, a maximum of 1 thousand interactions, and a tolerance of 0,0001. The tolerance is the stopping fitting criteria for the training interactions. To increase the comparability of the models, the 5-fold cross-validation was applied to each model created, and the model with the highest validation score was selected for the analysis.

The results saved for each of the 100 models developed, are the accuracy for validation and test sets, the value of the regularization strength, the memory size of the model, the training and prediction time, and the wavelengths selected. Figure 6.7 presents the flowchart of the algorithm developed to train and evaluate the models created.

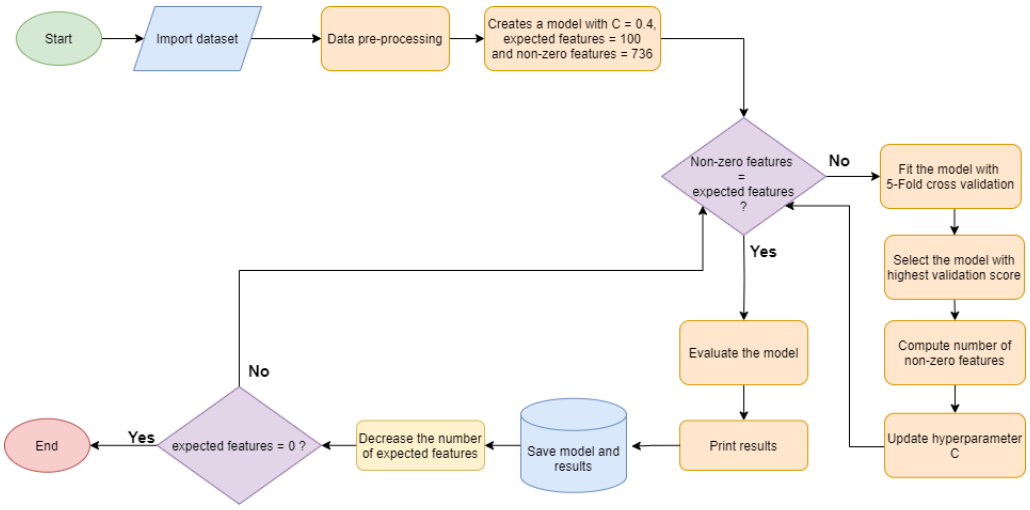


Figure 6.7: Flowchart of the logistic regression model evaluation.

The Scikit-learn library also counts with an implementation of the logistic regression combined with cross-validation. This function was employed to address the best value for C that leads to the highest accuracy of the model when the whole dataset

is available. The configuration applied created an array with 100 linear distributed values between 0,0001 to 10000 of possible C values that were analyzed by 5-fold cross-validation. The result of this evaluation was utilized as a benchmark reference, for the comparison with models that use lower features. To compare the results with a different algorithm, the random forest model was implemented in this work.

6.3 Random Forest

The random forest was developed in Python with the support of the functions available in Scikit-learn [41]. The library incorporates the training process of the model and allows the user to configure various hyperparameters, such as the number of trees, their maximum depth, and the split quality metric. To perform the evaluations proposed in this work, the model was first trained with all the available training samples of the dataset. In this case, the configuration used the bootstrap technique with the Gini split measure and no maximum depth limitation. This configuration was implemented due to the advantage of Gini's computation time over the entropy calculation, and the increased randomization during the creation of trees by the bootstrap method, which avoids trees with similar splits. In addition, the trees were allowed to grow indefinitely to achieve models with the lowest impurities.

The only hyperparameter tuned during the model's analysis was the number of trees. Even though random forests with an elevated number of trees do not present improvements in the miss classification rate, they result in a more stable feature importance metric [7]. Therefore, to achieve the best algorithm performance and feature importance stability, different models were trained logarithmically increasing the number of trees from 1 to 100 thousand. This extrapolation was performed to achieve a high stability in the feature importance calculation, given the elevated number of features available in the entire spectral resolution.

On the other hand, to avoid unnecessary large forests, the tunning of the number of trees allows a selection of the smallest forest to achieve a high classification performance. One technique to find this forest starts with an elevated number of trees, and removes one tree step-wised until a drastic decrease in accuracy occurs. Hence, the forest within one standard deviation of the accuracy of the largest forest is selected [58]. A similar approach was applied in this work, by logarithmically increasing the size of the forest and saving the performance of the models. This reduces the amount of time needed to train large forests and allows the analysis of the behavior of the models with extrapolated number of trees.

To select the forest size, the relationship between accuracy and number of trees was evaluated. A logarithmic dependence of the variables was observed. Consequently, the standard deviation and the mean of the accuracy for half of the models were computed. Models with the largest forests were chosen for evaluation due to the stability shown in their accuracy results. Subsequently, the smallest model that achieved accuracy within one standard deviation of the maximum calculated accuracy was selected for further analysis.

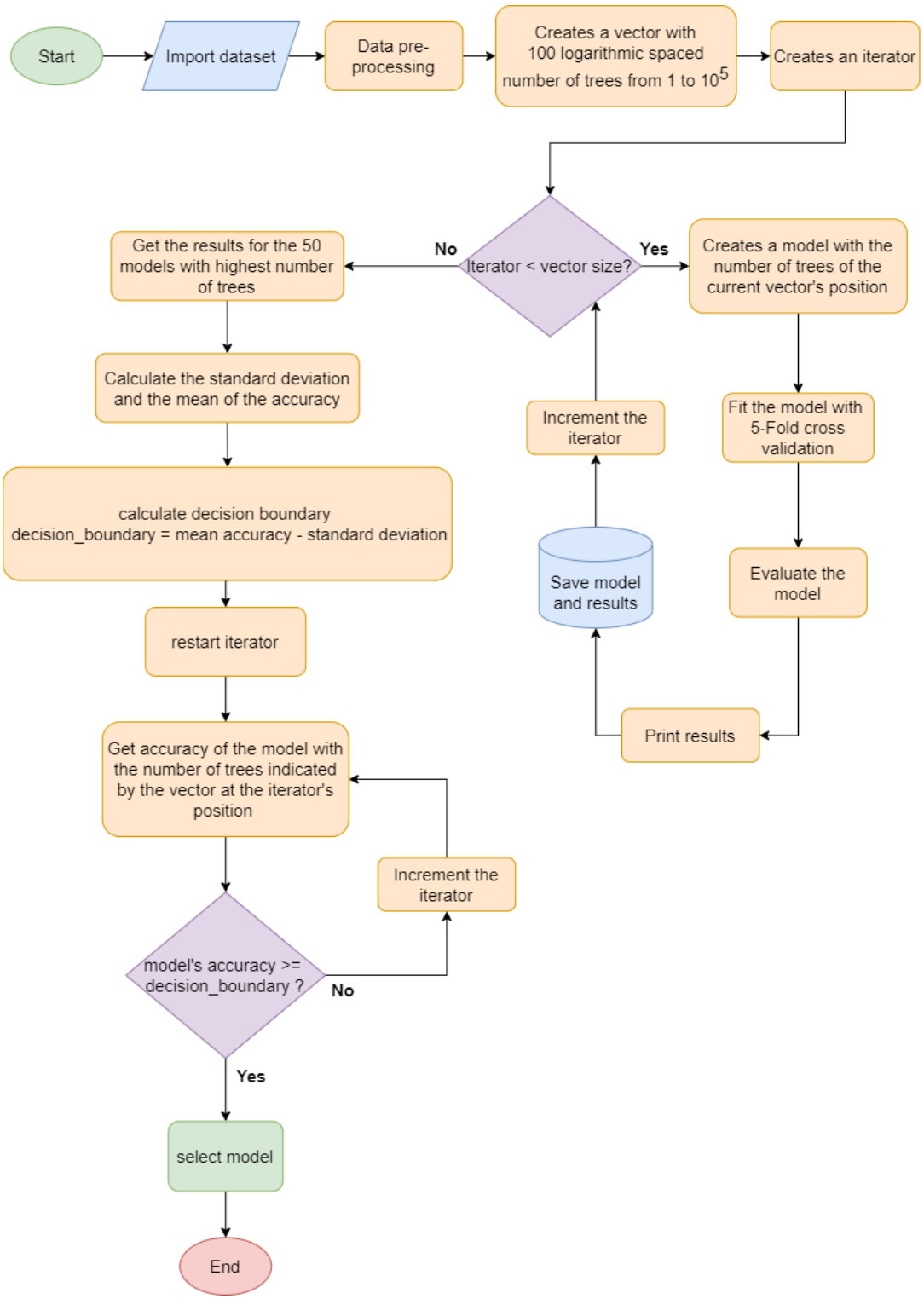


Figure 6.8: Flowchart of the random forest model creation and evaluation.

Figure 6.8 presents the flowchart of the algorithm used for the analysis of forest sizes and the selection of the optimal forest.

The results saved for each model were the accuracy in the test set, the training and execution time, and the model size. For the largest forest, the feature importance metric was also computed. In addition, the performance of the model with the optimal size was evaluated in more detail for the available dataset. Therefore, the confusion matrix and the feature importance were computed to be used as comparison metrics for other algorithms.

The feature importance analysis of the random forest model revealed elevated values for neighboring wavelengths. Hence, to achieve a simplified and automatic wavelength selection process, the optimal model was included in the RFE wrapper. This evaluation aimed to address the performance of the model in the best feature subset.

6.4 Recursive Feature Elimination

Similarly to the other models developed, the implementation of the recursive feature elimination wrapper was based on the functions available in Scikit-learn [41]. The Python library contains the structure of the RFE algorithm, and allows the user to choose parameters such as the model under analysis, the number of features to select, and the step of feature elimination. In this analysis, the random forest model with the optimal number of features described in Section 6.3, was applied in this wrapper for the comparison with the feature selection capabilities of the L1-normalized logistic regression.

To achieve comparable results, the relationship between the accuracy of the optimal random forest and the number of wavelengths was analyzed. Therefore, the number of features selected by the RFE algorithm was iteratively reduced, decrementing by one feature at a time from 100 down to 1. The results are similar to the analysis performed in Section 6.2 and provide a comprehension of the performance of the algorithms for the processing of data from hyperspectral or multispectral systems.

Since the random forest model is retrained at each run of the algorithm, the performance of the models were saved for the detailed comparison with the results obtained for the logistic regression model. The results computed for the analysis of this model were the accuracy, the memory size, the number of selected features, the selected wavelengths and the prediction time.

The flowchart of the algorithm for the selection of the subset of the most important wavelengths is presented in Figure 6.9.

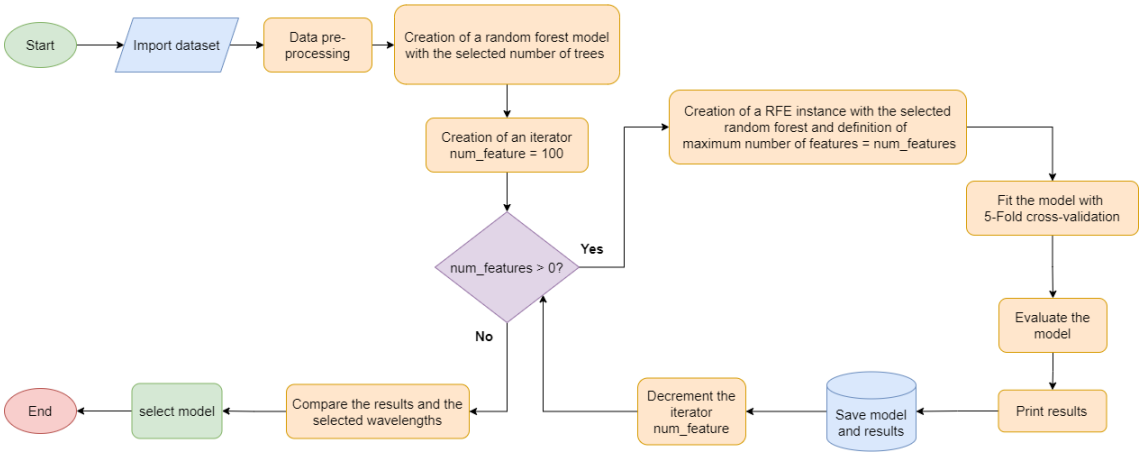


Figure 6.9: Flowchart of the recursive feature elimination analysis.

6.5 Multispectral LiDAR Data Evaluation

To evaluate the capabilities of the algorithms developed in this work, the model with the best performance in the dataset created was selected and applied in the classification of data from a prototype of a multispectral LiDAR system. The choice of this model took into consideration the capabilities of the demonstrator for spectral data acquisition. The prototype can emit light within a bandwidth of 10 nm, therefore, the model that resulted in wavelengths with a spectral distance higher than 10 nm while maintaining a high accuracy was selected. In addition, the number of wavelengths selected by each model was also considered as a choice metric, since the lower spectral resolution allows a further implementation of the system using discrete laser sources.

After the selection of the model and the subset of wavelengths, an evaluation of the data acquired by the multispectral LiDAR demonstrator was carried out. Therefore, sample 1 of the class organic was placed in front of the scanning mirror and measured at a fixed position, using a high resolution spectrum with 1000 wavelengths. These wavelengths were divided equidistantly in the entire measurement spectral range from 1100 nm to 1600 nm. Afterwards, the results were compared with the measurements acquired from the spectrometer, to evaluate the effects of using the collimated light source of the laser compared to the tungsten light source of the FT-IR spectrometer.

Later, the effects of the scanning pattern of the demonstrator were addressed. At the demonstrator, the position of the light beam is changed during the scanning by the Galvoscanner. On the other hand, in the FT-IR spectrometer, the position of the light beam is fixed, while the sample is moved by the microscope plate. This difference in scanning the sample surface was investigated by configuring the moving mirror to move the light beam at the surface of each sample in a grid of 5x5 points.

Following the evaluation of the influence of the demonstrator parameters in the spectral data, a setup with 6 samples of materials was created in the laboratory

test bench. The selection of the materials was performed based on the samples used for the creation of the dataset. This approach allowed the evaluation of the accuracy of the model in classifying spectra that are known since they were included in the training procedure. This facilitates the isolated evaluation of the laser light's influence on the acquired data. Hence, one sample of each class was selected to be used in the scene. The samples selected were sample 1 of organics, sample 1 of fabrics, sample 5 of plastics, and sample 3 of metals shown in Figure 6.1. The background was made of a sample of metal similar to the sample 4 of this class. The sample 5 of organics and a similar unknown piece of wood were added to the scenario to increase the spatial distribution of the samples. The samples used in this experiment were positioned in the field of view of the system starting at 45 cm of distance from the moving mirror, and separately by approximately 5 cm creating a scenario for the point cloud measurements. To facilitate the visual distinction between the samples, the metal was placed on top of an organic sample, and the fabric was glued to the plastic sample. The visible image of the materials used and the scenario created is shown in Figure 6.10.

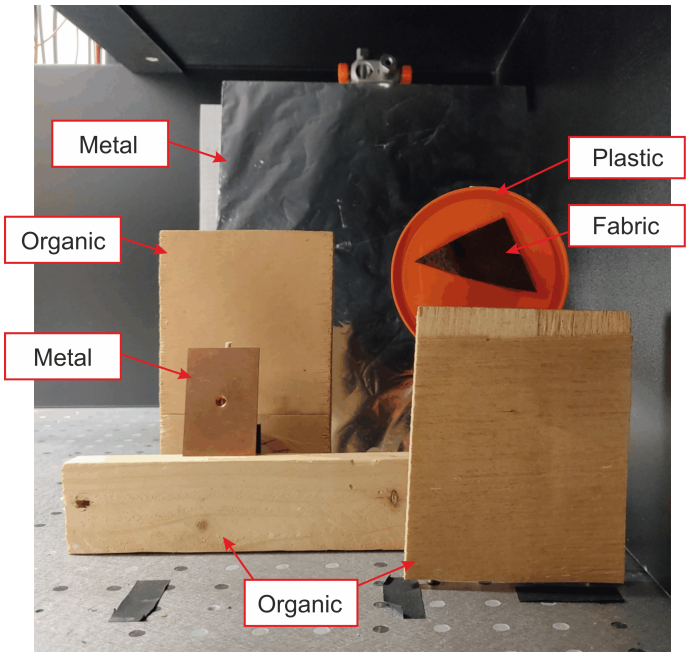


Figure 6.10: Measurement scenario of the multispectral LiDAR point cloud.

Following the creation of the scenario, the environment was measured by the multispectral LiDAR demonstrator. Therefore, the subset of wavelengths selected by the chosen model was used to configure the LabView application of the demonstrator. A detailed explanation of the selection of the subset of wavelengths is presented in Section 7.2. The LabView application was configured to operate in the spectral acquisition mode and to measure the selected wavelengths at each point of a grid of 101x101 pixels to generate a multispectral image of the environment created. After-

wards, the operation mode was changed for distance measurements, which acquired the ToF of each point in the same configured grid, generating the multispectral point cloud of the environment created.

To address the accuracy of the model in the point cloud, a ground truth image of the environment was created. Therefore, the multispectral image generated was saved as a .PNG file, and edited using a standard image edition software. A color scheme was created, to map each material to a specific color. Hence, each pixel of the image was manually painted with the color corresponding to its material and the resulting image was saved to be used as a ground truth for the analysis.

Finally, an algorithm was developed to classify the material at each point of the point cloud measured. It started by importing the classification model, the multispectral point cloud, and the ground truth image of the scenario. The algorithm uses the horizontal, vertical, and spectral dimensions of the datacube to generate a multispectral image of the environment. Then it scans the pixels in column-wised pattern of the image, starting at the upper left and moving into the x axis until the lower right. At each pixel, the spectral dimension of the datacube is used as input for the classification algorithm. Then a material is classified for the respective pixel. The result of the classification is later compared to the ground truth material of the same pixel. This allows the evaluation of the accuracy of the classification model for each pixel on the entire image. Afterwards, a color scheme is created mapping one color to each class of material. Hence, the pixel under analysis receives a color respective to the material resulting from its classification. Afterwards, the accuracy of the whole image is computed and the distance measurements are added to the painted image to generate the multispectral point cloud of the environment. Figure 6.11 illustrates the generation of the classified multispectral point cloud.

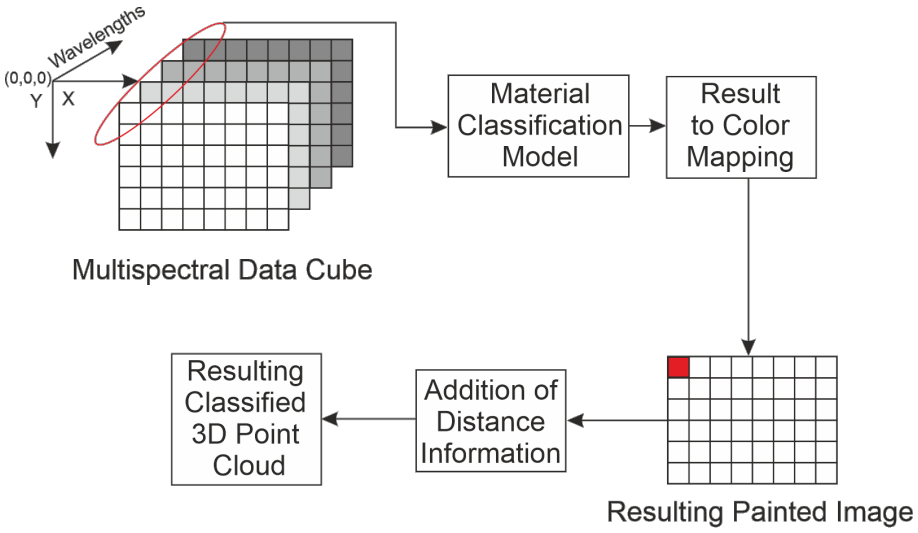


Figure 6.11: Illustration of the generation of the classified multispectral point cloud.

The flowchart of the algorithm used for the classification of the multispectral point cloud is presented in Figure 6.12.

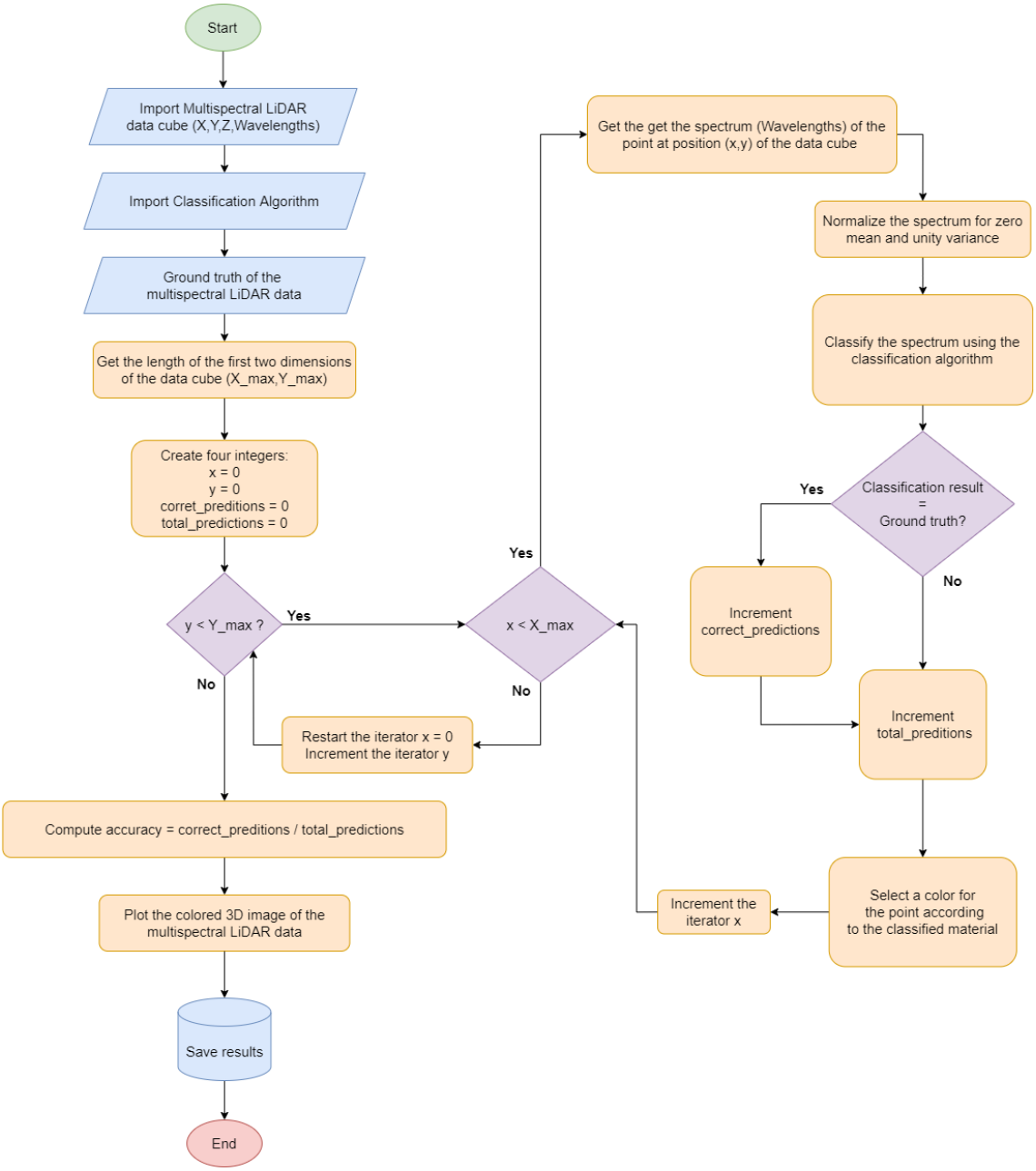


Figure 6.12: Flowchart of the evaluation of the classification algorithm on the demonstrator data.

7 Results

7.1 Classification Performance

This section outlines the results acquired from the project's analysis. It is divided into three main evaluations. First, the classification performance of the algorithms on the entire dataset is addressed. Afterwards, the capabilities of the algorithms to perform feature selection are analyzed. Finally, one model is selected to classify the data from the multispectral LiDAR and the results are discussed.

The performance evaluation of the models began by classifying the spectral data based on the complete spectrum acquired by the FT-IR spectrometer. As this spectrum comprises the maximum number of wavelengths accessible for analysis in this study, the results served as the benchmark for comparing performance with the models post feature-selection. Furthermore, the vast quantity of spectral data enabled a fair comparison of the models during the optimization procedures.

7.1.1 L1-Regularized Logistic Regression

The first algorithm to be analyzed was the L1-regularized logistic regression. To address the best performance of the model in the entire dataset, the regularization strength, hyperparameter C of the Sckit-learn library [41], was tuned to its optimum value. Therefore, the algorithm implementation with cross validation was applied. This function of the Scikit-learn library seeks to find a value for the regularization strength that maximizes model's accuracy. In this project, 5-fold cross-validation was used, to increase the reliability of the results while maintaining a low training time. The 5-fold resulted in 320 spectral samples for each validation set.

The resulting model created four classifiers, one per class, due to the one-vs-rest approach. The model was evaluated for its accuracy in test, memory usage, number of features used, and prediction time. Each classifier created, uses different number of features to perform the classification. Thus, the cross-validation technique resulted in a vector with one regularization strength for each classifier. Since the final classification of the model considers the results of all the classifiers, the one with the highest number of features was selected for the analysis. This was the classifier created for the class of fabric which used all the 736 features available for the classification. The regularization strength used by this classifier was 75. The entire model achieved an accuracy on the test set of 95.75% and performed a prediction a single sample in 1.12 miliseconds.

On the other hand, the classifier with the lowest number of features used 524

wavelengths from the available data. Given the elevated number of features used by each of the internal classifiers, the final model occupied 11.29 Mbytes in memory.

The miss classification of the algorithm was also analyzed and is shown by the confusion matrix in Figure 7.1.

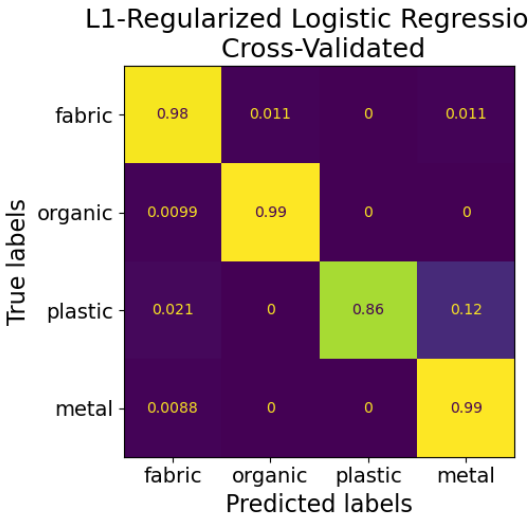


Figure 7.1: Normalized confusion matrix of the best L1-regularized logistic regression.

The normalized confusion matrix presents the accuracy achieved by the algorithm at each class. It can be observed that the model made the highest miss classification when analyzing samples of plastic. This class also required the classifier to use more features to perform the classification. One possible explanation for the difficulty in the classification of this class, is the variance of the sample's spectrum. The confusion matrix also shows that the highest confusion of the model happened between the plastic and the metal classes.

A closer analysis in the mean spectrum of both classes can support the explanation of this fact. Samples 2 and 3 of plastic present a linear mean spectrum with an inclination similar to most of the spectrum of the metal class. This similarity in shape and inclination can be the main reason behind the elevated confusion. It is reinforced by the miss classification of the model from plastic to fabric. The sample 4 of the class fabrics also presents the similar characteristics as the above mentioned. Since the logistic regression is a weighted sum of the amplitude at each wavelength, the similar inclination of the spectrum can cause the model to achieve comparable weights that result in the confusion when these spectral samples have to be classified.

7.1.2 Random Forest

The random forest model was evaluated in the entire dataset to compare its performance with the L1-regularized logistic regression. The model was optimized by

fine-tunning the number of estimators inside the forest as described in Section 6.3. The described evaluation resulted in 100 models that were compared considering their mean accuracy during 5-fold cross-validation. The results of this comparison supported the choice of the smallest forest.

The relationship between the accuracy and the number of trees of these models was evaluated. The results are presented in Figure 7.2. The blue dots represent the mean of the accuracy during 5-fold cross-validation for each model and the gray lines represent their respective standard deviation. The number of estimators, or trees, inside each version of the model, is shown in a logarithmic scale.

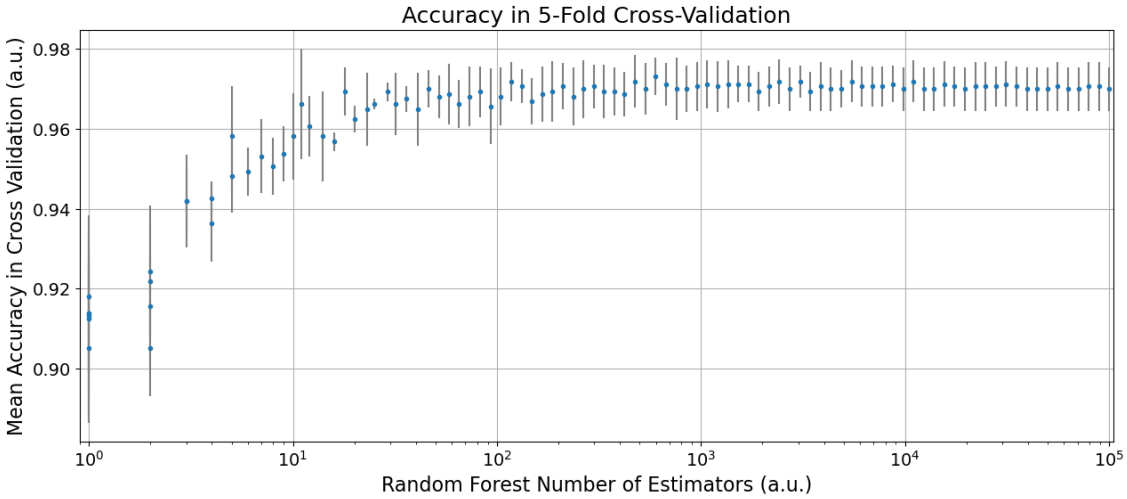


Figure 7.2: Relation between random forest size and model's accuracy.

The graph illustrates a logarithmic relation between the accuracy and the size of the forest. Due to the utilization of a logarithmic function to generate the models some instances contain the same amount of estimators, therefore they appear repeatedly in the graph. From the result, it is possible to observe that the mean accuracy stabilizes after a certain forest size. Hence, increasing the number of trees indefinitely does not improve the accuracy of the model, which has been already discussed by [7]. The model with 599 estimators achieved the highest mean accuracy of 97.31% during this analysis. However, it is possible to observe a fluctuation in the mean accuracy measurements. This variation comes from the randomization included during the tree creation process by the bagging technique. Therefore, to decrease the influence of this effect in the analysis of this project, the procedure used selected the model with the smallest forest able to achieve a mean accuracy within one standard deviation of the stable region of the graph.

The selection of the smallest forest is supported by the analysis of the size occupied in memory by these models. As can be in Figure 7.3, the amount of memory occupied by the models increases exponentially with the number of estimators inside the forest. Therefore, the smallest forest to achieve high accuracy is the best choice to maintain a balance between the model's performance and memory usage.

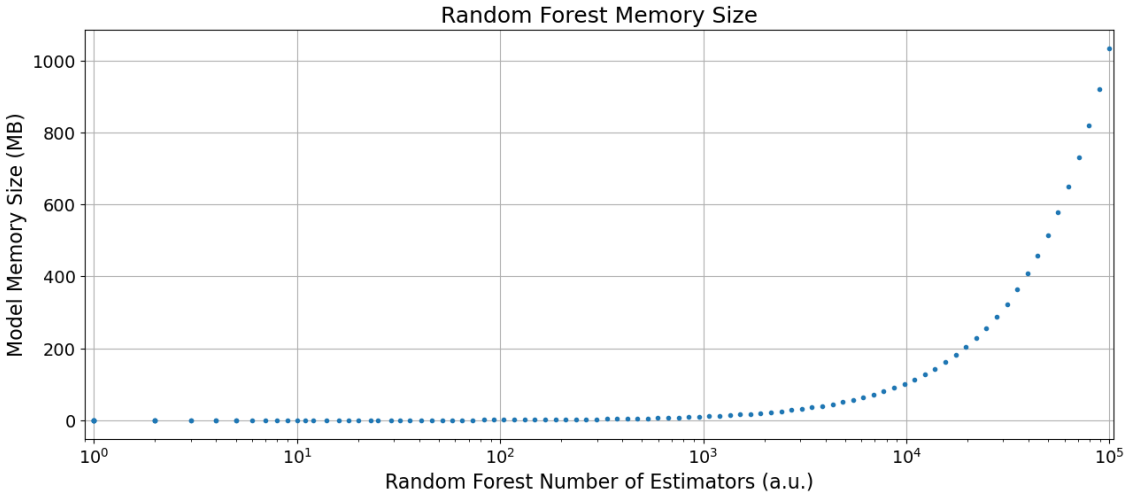


Figure 7.3: Relation between occupation in memory and quantity of estimators of a random forest model.

To find the smallest forest the average of the average accuracy for half of the models was calculated, as well as their standard deviation. For this computation, the 50 models with the highest number of trees were selected, as they belong to the stable region of the results shown in Figure 7.2. The average achieved from their mean accuracy was 0.97 and the standard deviation was 0.0008. Hence, the smallest model to achieve the mean accuracy within one standard deviation of the calculated average was selected for further analysis.

The results of this evaluation are better illustrated in Figure 7.4. The rectangular blue area highlights the models used for the calculus of the average and standard deviation. The yellow line represents the average of the mean accuracy of these models. The green lines expose the region between one standard deviation of the average calculated. Finally, the red line shows the position of the resulting model with the smallest forest selected for the analysis of this project.

This model contains 46 estimators and achieved an accuracy of 97.75% in the test set. The memory size occupied by this model was 0.48 MB, and the prediction of one spectrum lasted 46.85 ms. For comparison purposes, the number of features used by the model was also evaluated. Therefore, the feature importance metric was calculated, and the number of features which the metric was different from zero was computed. This resulted in 694 wavelengths that were necessary for this model to perform the classification.

A more detailed analysis of the classification capabilities of the selected model was obtained through the evaluation of the confusion matrix. This normalized matrix is shown in Figure 7.5. From the results, it is possible to observe that differently from the L1-regularized logistic regression, the random forest model achieved the highest accuracy when predicting samples of fabric, and the lowest when predicting samples of metal.

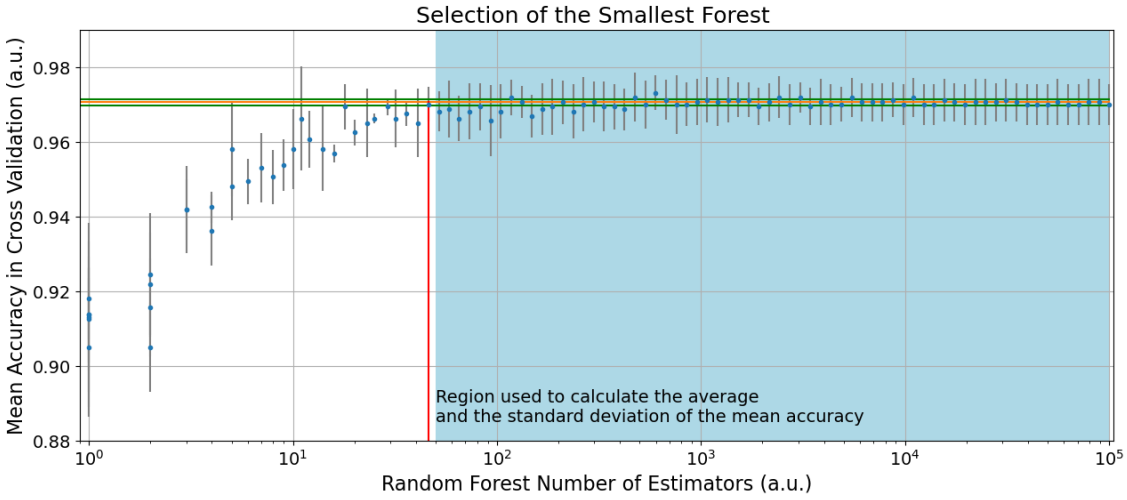


Figure 7.4: Selection of the smallest random forest model.

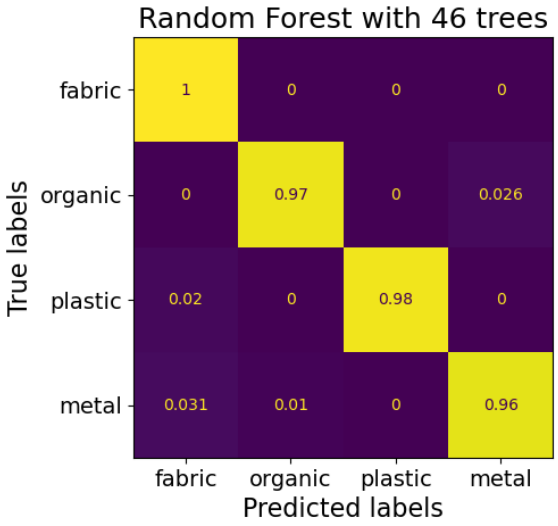


Figure 7.5: Normalized confusion matrix of the selected random forest model.

Similarly with the approach used for the analysis of the confusion matrix from the logistic regression model, the results were evaluated by the comparison of the spectral fingerprint of the samples. Hence, considering the average presented in Figure 6.6 it is possible to understand the confusion between the plastic and fabrics classes, due to the similarities presented in the spectrum of samples 2 and 3 of plastic compared to sample 4 of fabrics. However, only the visual analysis of the average of the spectrum does not clarify the confusion between the metals and organics classes. One possible reason for this confusion is the irregular surface of the organic samples that increase the standard deviation of the spectrum measured, and can result in fingerprints with similar shapes causing the algorithm to miss classify. A second

reason that may confuse the model is the similarity in the substances that compose the samples used in the creation of the dataset. A detailed analysis of the substances that compose the samples used and the performance of the classification of these substances can be a subject for further studies.

The overall analysis of the performance of the models developed from the entire data available has shown that random forest achieved a higher accuracy when classifying this dataset. The explanation relies on the capability of random forests to learn complex patterns when the trees grow indefinitely. By splitting the features and performing the maximum voting, this model presents more stable results than the one-vs-rest approach of the L1-regularized logistic regression method.

7.2 Feature Selection

Following the evaluation of the performance of the models when the entire dataset is available, they were evaluated considering a reduction in the amount of spectral information available. Hence, the number of wavelengths used as features was limited to address the capabilities of the models to classify data from multispectral system domains.

This section describes the results for the application of the feature selection techniques which aims to find a subset of features that maximizes the performance of the model to guide the development of a multispectral LiDAR system. It starts by the analysis of the intrinsic feature selection capabilities of the L1-regularized logistic regression, followed by the evaluation of the intrinsic feature importance metric of the random forest model. Finally, assessing the results of the recursive feature elimination model.

7.2.1 L1-Regularized Logistic Regression

The evaluation of the feature selection techniques started with the LASSO regularization, L1-norm, in the logistic regression model. Therefore, the experiment described in Section 6.2 was performed. It was observed that the relationship between the amount of features used by the model and its regularization strength is non-linear. Hence, a dynamic update of the regularization had to be performed. This allowed the comparison of models using a maximum of 100 wavelengths. The relation between the regularization strength parameter and the number of wavelengths selected by the model is shown in Figure 7.6.

The results exemplify the nonlinearity in the relation between the two variables. They also expose the intervals between the regularization strength values necessary to achieve a unitary decrease in the number of wavelengths selected, which occurred due to the dynamic update of the regularization strength parameter. The decreasing step size of this parameter was updated dynamically to fine-tune the regularization strength in order to achieve a specified number of wavelengths. Therefore, the distance in the measurements corresponds to the values where the step size was

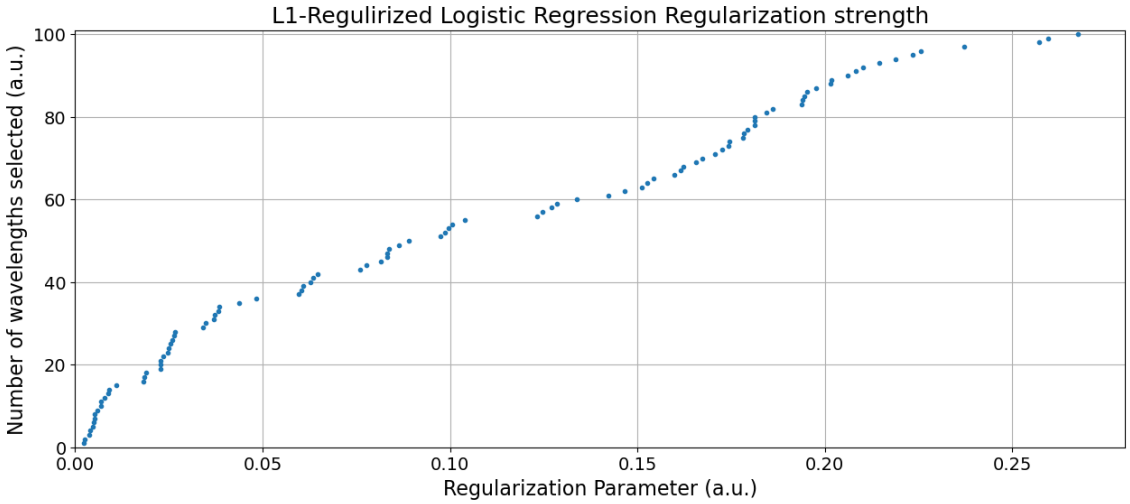


Figure 7.6: Relation between regularization strength and wavelength selection.

changed during the analysis. The possible explanation for the non-linearity comes from the theory behind the feature selection capabilities of the lasso model. Since the L1-norm is a penalty term added to the loss function of the logistic regression, the training procedure of the model aims to find the minimum of this non-linear regularized loss.

The minimum of this loss occurs in the interception point of both terms of the Equation 2.8. This point of interception lies where some of the weights of the features are equal to zero. Therefore, when the number of selected features is fixed to a certain value, and the regularization strength is modified, the training procedure updates the weights of each feature generating models with different coefficients.

A specific goal of this project is to evaluate the relation between the number of wavelengths used by the model and its accuracy. Hence, a graph showing the relation of both variables was created. The results are presented in Figure 7.7. The blue dots represent the average and the gray lines the standard deviation of the accuracy for each model during 5-fold cross-validation. The graph shows the results starting from the model using 2 wavelengths, since a monochromatic model can not be used for material classification in practical applications, given the insufficient amount of spectral information contained in only one wavelength.

As can be seen in the graph there is a strong decay in the accuracy for the versions of the model using less than 5 wavelengths to perform the classification. The reason for this decay lies in the characteristics of the model to create one classifier for each class in the one-vs-rest approach. Therefore, when the number of features selected is lower than the number of classes, there is a huge increase in the miss classification rate of the model. In addition, the results also show that evaluating the amplitude of 96 wavelengths is enough for the model to achieve an accuracy higher than 90%. The non-linear relation between the number of features and the accuracy of the model is also related to the fact that the regularization strength generates models

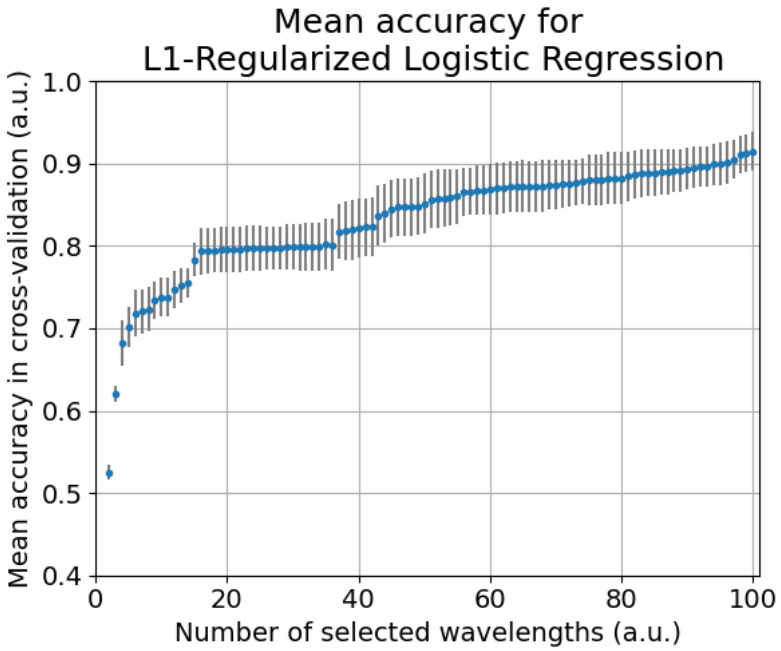


Figure 7.7: Relation between accuracy and number of selected wavelengths for the logistic regression model.

with different coefficients to achieve the specified number of wavelengths.

Afterwards, the wavelengths selected from the models were evaluated to identify the regions of the spectrum that contain most of the information necessary to classify the materials. For this reason, the wavelengths selected by each of the 100 models created were analyzed through the histogram of the spectrum presented in Figure 7.8.

The histogram counts the number of times that a specific wavelength was selected across all the 100 models. From the histogram, it is possible to observe that some of the wavelengths were repeatedly selected by diverse versions, while other regions of the spectrum were not used by any of them. This confirms the hypothesis that not all the spectral data is necessary to perform the classification of materials.

The evaluation procedure required the selection of one version of the models developed. This selection took into consideration the performance of the models in terms of accuracy and selection of wavelengths. Since the goal of this work was to select an optimized model to be tested in a multispectral LiDAR system the practical capabilities of the demonstrator had to be considered. The development of a system using discrete monochromatic laser sources requires the fulfillment of the physical limitations of the hardware. One of them is the selection of the lowest amount of wavelengths. Since each wavelength selected represents a laser source on the discrete system and its respective optical components, the selection of the lowest amount results in a reduction of hardware costs. In addition, each discrete laser system emits light in a defined bandwidth, as described in Chapter 2.2, that

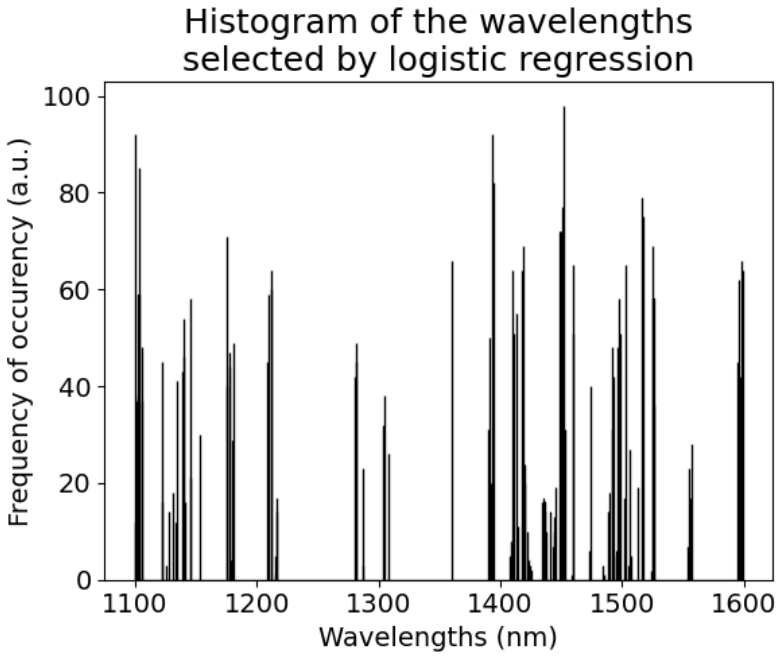


Figure 7.8: Histogram of the wavelength selected by 100 models of L1-regularized logistic regression.

is normally higher than the resolution of the FT-IR spectrometer. Hence, to select a model that meets these requirements the first 10 versions with the lowest number of wavelengths used were analyzed in detail. The model with the highest spectral distance between the wavelengths selected was chosen for further evaluation. Since the model using a single wavelength is not suitable for practical application, it was removed from the analysis. Hence, the spectral position of the wavelengths selected for each of the resulting 9 models is presented in Figure 7.9. The vertical axis presents the number of wavelengths selected by the different versions of the model, and the horizontal axis presents the position of these wavelengths in the spectrum.

As can be seen from the results in Figure 7.9, the model with 6 wavelengths presented the highest spectral distance between the wavelengths selected while maintaining the highest accuracy. Even though models using more wavelengths achieved higher accuracy, the results show that these models selected neighbor wavelengths, which can not be measured in practical applications using laser systems given the bandwidth of the emitted light. Therefore, the model with 6 wavelengths was selected for further evaluation. This model achieved accuracy in the test set of 65.25%. The memory size usage of the model was 0.02 MB and the time needed to predict one sample of material is 1.04 ms. To increase the comparability of these results, the feature selection capabilities of random forest and recursive feature elimination were similarly analyzed.

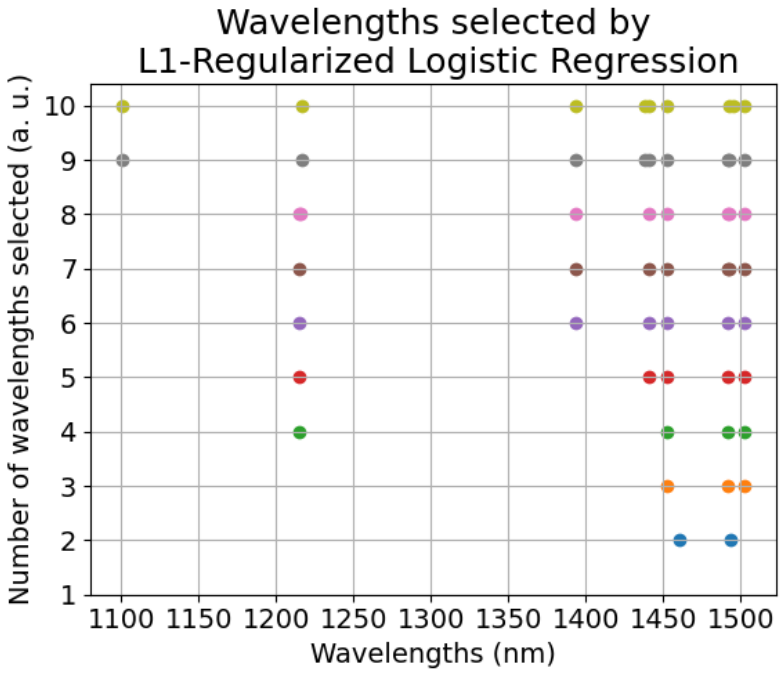


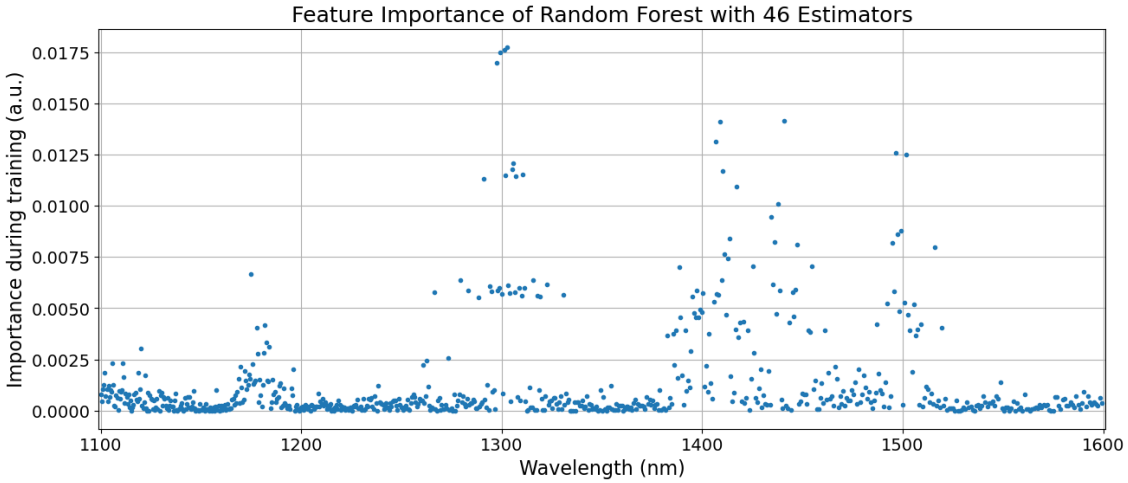
Figure 7.9: Positions of wavelengths selected by 9 versions of L1-regularized logistic regression.

7.2.2 Random Forest

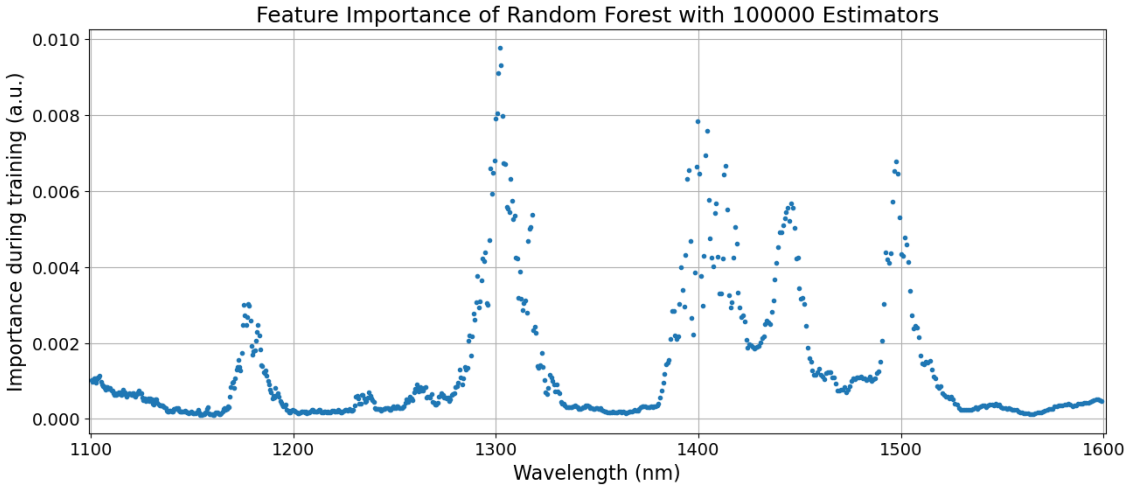
One characteristic of the random forest is the capability to assign an importance metric for each feature. This metric was analyzed in detail to evaluate the potential of the model in performing wavelength selection. Therefore, the feature importance of the model selected during the optimization of the random forest, with 46 estimators, was computed and is presented in Figure 7.10(a). However, the author in [8] states that larger forests have higher stability during the feature importance calculation. Hence, the results were compared with the calculated importance of the largest forest developed, which consisted of 100 thousand estimators, as described in Section 6.3. This comparison is presented in Figure 7.10.

The graphs reinforce the hypothesis that the stability during the calculation of feature importance is higher using models with more estimators. In addition, the results presented in Figure 7.10(b) reveal that the model with the largest forest attributed high importance to 5 distinct regions of the spectrum in the shape of peaks. Even though the peaks achieved different amplitudes, it is possible to observe that wavelengths with similar importance rely on the same peak with close spectral positions. This graph shows that setting an importance threshold would lead to a selection of adjacent wavelengths. For instance, selecting all the wavelengths with an importance higher than 0.0085 would result in the three points located around 1300 nm.

Since spectral data are highly correlated, it is expected that the information



(a) Feature importance of the optimized random forest.



(b) Feature importance of the largest random forest.

Figure 7.10: Comparison of feature importance from two forest sizes.

present in neighbor spectral regions has the same influence on the classification results. For instance, the regions of interest identified by the algorithm were also used in [61] for the distinction of plastic in water. The water, which might be present in the organic samples, shows a characteristic absorption band in various wavelengths starting from 1400 nm. Hence, the algorithm uses samples of these spectral regions to perform the classification. However, the absorption characteristics of the spectrum are highly dependent on the substances that compose the materials samples and the relation between the absorption band and the type of substance is still subject to further investigation.

Therefore, a more informative approach to give to the algorithm samples on different absorption regions is selecting the wavelengths at the five peaks of the graph.

One possibility to achieve this goal would be manually selecting the central wavelength of each peak, however, this approach does not allow a detailed analysis of the relation between the amount of features and the accuracy of the model. An alternative approach that allows the selection of the wavelengths automatically, is through the application of wrapper methods. Therefore, the random forest model was placed inside the recursive feature elimination wrapper to eliminate features with lower relevance and evaluate the model's performance in different system domains.

7.2.3 Recursive Feature Elimination - RFE

The evaluation of the recursive feature elimination method was performed as described in Section 6.4. This allowed the comparison of the relation between number of wavelengths and the accuracy of the random forest with the results obtained by the L1-regularized logistic regression. Therefore, the random forest model with 46 trees selected from the optimization of the forest size was placed inside the recursive feature elimination (RF-RFE) wrapper. Afterwards, the wrapper was evaluated for a maximum of 100 wavelengths aiming to compare its results with the analysis performed for the L1-regularized logistic regression. Hence, the elimination of one feature at each iteration was applied. The algorithm used 5-fold cross-validation to increase the comparability and the last 100 versions of the model were saved.

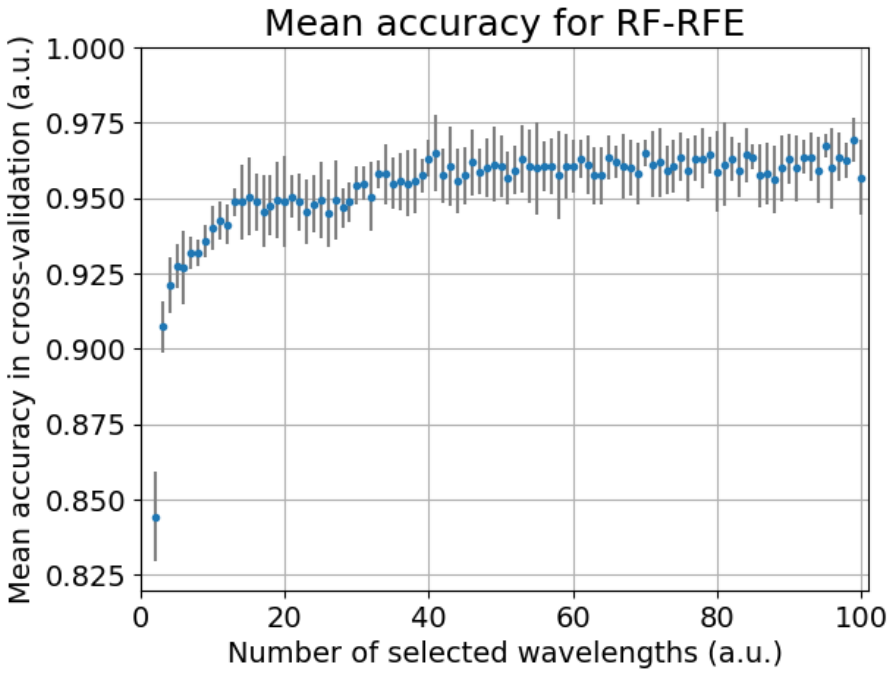


Figure 7.11: Relation between accuracy and number of selected wavelengths after RFE.

The relation for the mean accuracy during the cross-validation and the amount

of wavelengths selected by the wrapper are shown in Figure 7.11. Similarly with the results presented for logistic regression, the blue points represent the mean and the gray lines represent the standard deviation of the accuracy during 5-fold cross-validation. In addition, the model using a single wavelength was removed from the analysis due to the irrelevance of the result for practical applications, since the spectral measurements with a single wavelength can not be used for classifying materials.

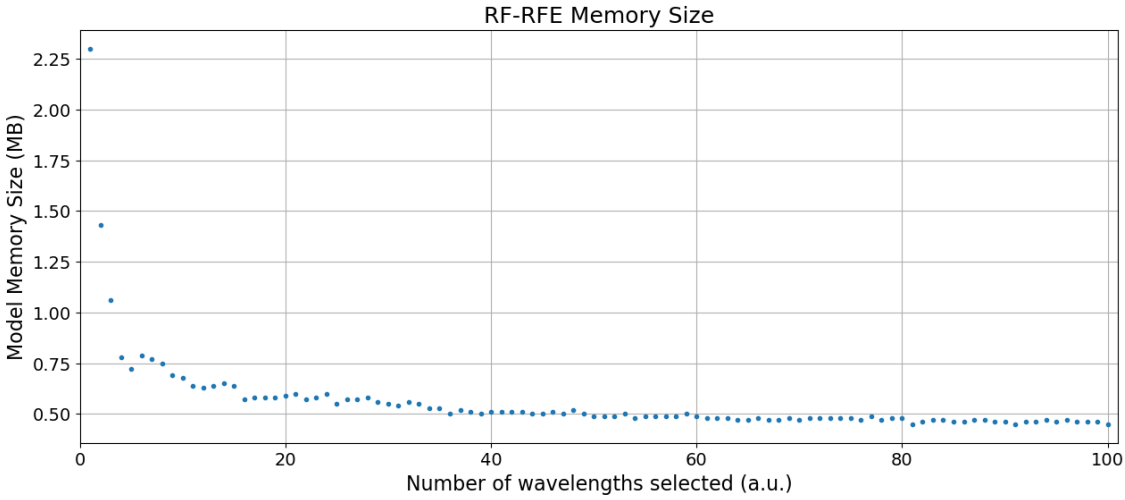
From the results presented in Figure 7.11 it is possible to observe a logarithmic relation between the accuracy of the model and the number of wavelengths used during the classification. They indicate that a lower number of wavelengths is enough to achieve high accuracy levels and the use of the entire spectrum can result in a waste of resources. Moreover, the graph shows that a random forest model using the intensity data of two wavelengths have enough information to achieve a mean accuracy of almost 84.44%. This accuracy increases drastically to 90.75% with the addition of a third wavelength.

The results serve as a guide for the development of multispectral LiDAR systems with focus on material classification, since they show that increasing the amount of spectral data not necessarily increases the amount of information required for the material classification. Therefore, LiDAR systems may not depend on the use of hyperspectral resolution to be able to achieve high classification accuracy. This can decrease the cost of development of their new generations, by the possibility of using discrete wavelengths to acquire informative spectral data.

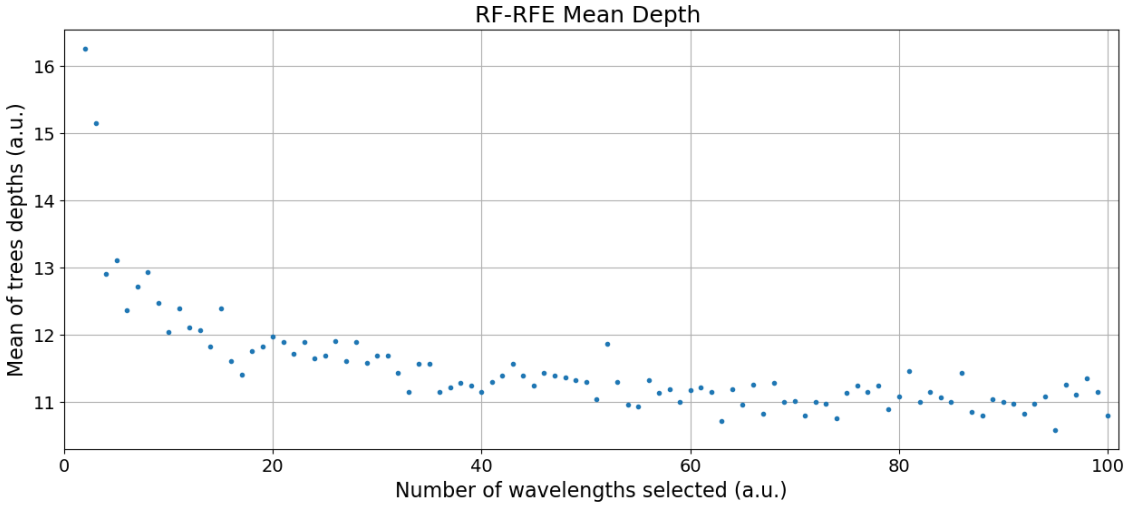
The highest average accuracy was achieved by the model with 99 wavelengths, and its value was 96.94%. However, the results show a fluctuation in the averages introduced by the randomness of the bagging technique. Hence, to achieve a more detailed analysis of the performance of these models, their memory usage were also addressed and the results are shown in Figure 7.12(a).

The outcome shows that the memory increases for models with lower number of wavelengths, which opposes the results obtained by the raw random forest model where the number of trees directly influenced the memory storage size. Since the number of trees inside of the wrapped random forest was defined to be 46, the results indicate that the model trains deeper trees to be able to perform the classification of the materials with lower spectral resolution. Consequently, deeper random forests are created occupying more memory space. To verify this hypothesis the depth of the trees were evaluated. One way to calculate the depth of the model is by summing the amount of edges inside a tree, as deeper trees have more edges and nodes. Scikit-learn counts with an implementation that calculates the biggest distance from the root to the leaves of each tree inside the random forest model [41]. Therefore, the average of the depth of the trees was computed as a metric to address the overall depth of the model, given that all models had the same amount of estimators. The average depth of the trees for each model is presented in Figure 7.12(b).

As can be seen in the Figure 7.12, the depth of the random forests are directly related to the memory usage of the model, confirming the hypothesis that models with lower spectral resolution need to train deeper trees to be able to perform a



(a) Relation between model's memory size and number of features selected.



(b) Average depth of the trees inside each model.

Figure 7.12: Memory and depth comparison of the RF-RFE models.

high accurate classification.

Finally, the wavelengths selected by each RF-RFE model were evaluated similarly to the evaluation procedure presented in the L1-regularized logistic regression. Therefore, the histogram of the wavelengths selected by the 100 models was created and is presented in Figure 7.13. The histogram shows that some regions of the spectrum were not used by the algorithm during the classification tasks. On the other hand, some of the wavelengths were used by distinct versions, demonstrating the high interest of the algorithm in these wavelengths.

Considering the same analysis performed for the L1-regularized logistic regression model, the wavelength selection capabilities were evaluated for a maximum of 10

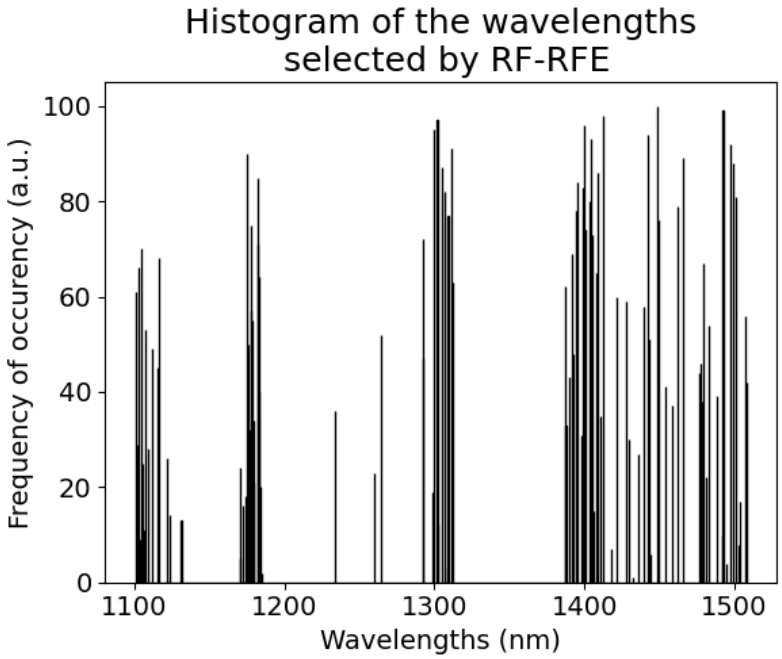


Figure 7.13: Histogram of the wavelength selected by 100 models of RF-RFE.

wavelengths. Hence, the values selected by these 10 models were plotted in a graph to allow the visualization of their position into the spectrum. Finally, the requirements of the hardware were considered for the selection of one model. The resulting graph is shown in Figure 7.14. The vertical axis presents the number of features selected by each version of the model, while the horizontal axis presents the position of these features in the spectrum.

It can be observed in the results that some of the models selected wavelengths in adjacent spectral positions. These wavelengths are difficult to be measured by a LiDAR system given the necessity of the use of laser emission in very narrow bandwidths, which increases the costs of development. Therefore, while selecting the model, the physical characteristics of the multispectral LiDAR hardware and its limitations during the light emission needed to be considered. Consequently, the model with 5 wavelengths was selected, since it maintained a high spectral distance between the wavelengths utilized. This model achieved an accuracy of 93.75% in the test set, occupied 0.72 MB in memory, and took 62.49 ms to classify one spectral sample.

7.3 Classification Model Comparison

To evaluate the performance of the models examined in the previous sections, they were compared based on several criteria, including the number of wavelengths utilized, the accuracy on the test set, time required to predict one spectral sample, and

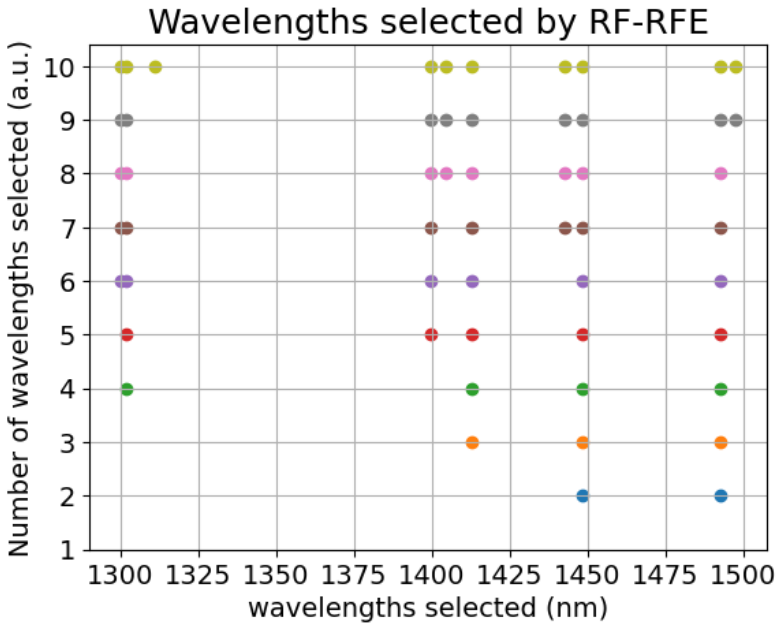


Figure 7.14: Spectral position of the wavelengths selected by 9 versions of RF-RFE.

size occupied in memory. The summary of these results for the four models are presented in Table 7.1. The first two rows of the table show the models that achieved the highest accuracy when all the spectral resolution was available for classification. These models are the optimized versions of L1-regularized logistic regression and the random forest with 46 estimators. Since they were allowed to utilize the entire spectral resolution of the dataset, their performance was considered the benchmark for the comparison.

The last two rows of the table expose the results of the models selected after the application of the feature selection techniques. These models were chosen considering their performance in accuracy while bearing the requirements for the development of a practical multispectral LiDAR hardware.

The table gives an overview about the potential of the feature selection techniques in the classification of spectral data. It shows that a drastic reduction in the amount of features does not decrease linearly the accuracy of the model. In addition, it reinforces the conclusions of the literature regarding the application dependency in the choice of the model. Furthermore, it shows that each version of the model has different characteristics in accuracy, prediction time and memory size that must be taken into consideration during model's selection.

	Model	Number of Features (unit)	Accuracy in Test (%)	Prediction Time (ms)	Memory (MB)
Before Wavelength Selection	L1-Regularized Logistic Regression	736	95.75	1.12	11.29
	Optimized Random Forest	694	97.75	46.85	0.48
After Wavelength Selection	L1-Regularized Logistic Regression	6	65.25	1.04	0.02
	Random Forest with RFE	5	93.75	62.49	0.72

Table 7.1: Comparison of the performance of the selected algorithms.

Comparing the two versions of the L1-regularized logistic regression, it is possible to observe that reducing the amount of features in 99.2% results in a decrease in accuracy of 30.5%. In addition, the prediction time was reduced in 0.08 ms due to the lower amount of calculations needed, and the memory storage size of the model was decreased in 11.27 MB. These reductions in memory size and processing speed support the importance of the application of feature selection for model optimization.

Similarly, comparing both versions of the random forest algorithm results in a reduction of 99.28% in the amount of features which only degraded 4% of the accuracy of the model. Yet, the memory size of the model increased in 0.24 MB and the prediction took 16.09 ms longer, due to the model's necessity to create deeper trees to compensate the performance in accuracy when a lower amount of data is available.

When comparing the versions of the models presented in the first two rows of the table, it is possible to observe that the optimized version of the random forest had an advantage of 2% in accuracy over the logistic regression. Moreover, the random forest did not used all the wavelengths to achieve 97.75% accuracy, even if no limitations in the amount of spectral data were imposed. However, the L1-regularized logistic regression performed a prediction 41.83 times faster than the optimized random forest model. This characteristic can take a crucial role for safety critical applications, where the classification must be performed within a strict deadline.

The results are reinforced by the comparison of the models presented in the last two rows of the table. The random forest achieved an accuracy of 30.4% higher than the accuracy of the L1-regularized logistic regression using one feature less. On the other hand, it also occupied extra 0.7 MB and took 60 times longer to perform the prediction of a single spectral sample than the logistic regression model.

To select one model for the evaluation in the LiDAR demonstrator, their perfor-

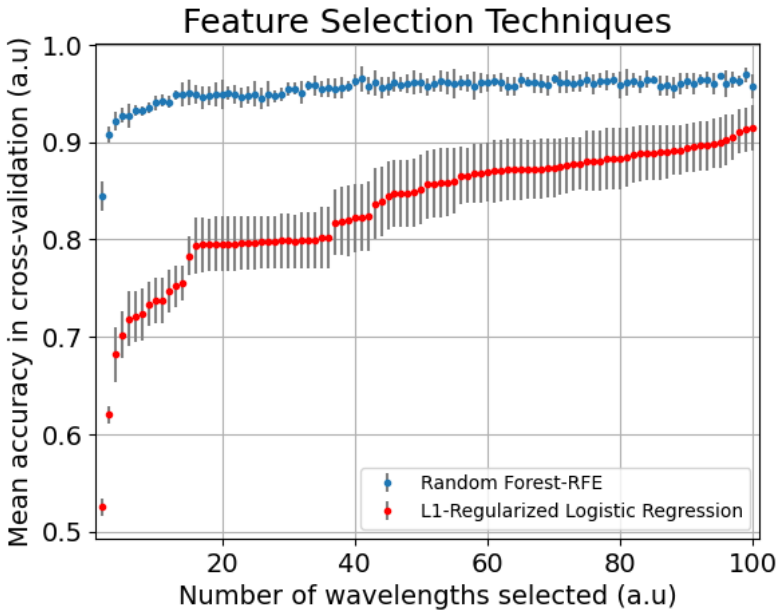


Figure 7.15: Comparison between random forest and L1-regularized logistic regression.

mance in feature selection while maintaining a high accuracy were considered. It is perceptible in Table 7.1 that the random forest achieved higher accuracy than the L1-regularized logistic regression in both scenarios. This behavior was observed again during the comparison of all the models created, which is presented in Figure 7.15.

The results evidence the superior performance of the random forest model over the L1-regularized logistic regression, achieving a higher accuracy in test independently of the number of wavelengths used by the model. This occurs given the possibility to create models with deeper trees in random forest, which overcomes the capabilities of the logistic regression when the amount of features is limited. However, the high accuracy is maintained by the cost of higher memory consumption and lower prediction speed.

Since the goal of this work is to choose one version of the model with high accuracy for a practical application, the model presented in the last row of Table 7.1 was selected to be tested in the multispectral LiDAR demonstrator.

7.4 Multispectral LiDAR Demonstrator

This section discusses the results obtained by the evaluation of the classification model in measurements performed with the multispectral LiDAR demonstrator presented in Section 5.3. The experiments were structured into three primary phases. First, a high resolution spectrum of a single point from a target sample surface was measured to analyze the comparability of the entire spectrum with the data of

the dataset. Afterwards, a grid with 5x5 points were measured with the selected wavelengths to evaluate the repeatability of the spectrum at different positions on the same sample's surface. In the end, the point cloud of a scenario created with different materials samples was acquired, and the performance of the classification was discussed.

7.4.1 Single Point Spectrum

The evaluation of the capabilities of the classification model in the multispectral LiDAR demonstrator started with the analysis of the differences in the spectrum obtained by the prototype in relation to the data acquired by the FT-IR spectrometer. Therefore, the reference sample presented in Figure 6.2 was placed in front of the moving mirror, and 1000 equidistant spectral points were measured. Afterwards, the reference was changed to the organic sample 1 presented in Figure 6.1, and the procedure was repeated. To decrease the influence of the effects of the light source in the result, the spectrum acquired for the organic sample was divided by the spectrum of the reference sample. This resulted in the a reflectance for the sample measured that was comparable with the reflectance spectrum obtained by the FT-IR spectrometer.

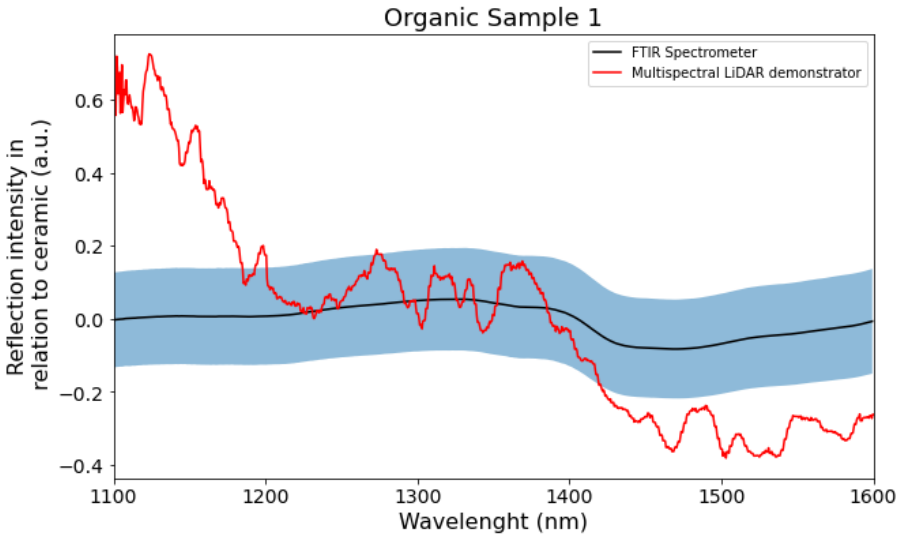


Figure 7.16: Spectral reflectance comparison between the multispectral LiDAR demonstrator and the FT-IR spectrometer for an organic sample centered at zero.

Figure 7.16 shows a comparison between the reflectance obtained by the multispectral LiDAR demonstrator, in red, and the average, in black, and the standard deviation, in blue, of 100 measurements performed by the FT-IR spectrometer. To facilitate the comparison, the mean of the spectral fingerprints was removed. Even though their measurements have different resolutions, this analysis gave a general

overview of the behavior of the spectral fingerprints with amplitudes around zero. This removed the effect of the different amplitudes and exposed the spikes present in the data measured by the demonstrator.

Since the classification model was trained in the calibrated commercially available FT-IR spectrometer, the spectrum acquired by this equipment was considered a benchmark to be achieved by the measurements of the demonstrator. In this case, similar performance in accuracy in the data acquired by the demonstrator could be obtained. Hence, to better understand the influences of the system that were leading to the deviation in the spectral measurements, different parameters of the prototype were analyzed.

During the measurement, it was observed an elevated heating of the laser system. So, the analysis started with the influence of the heating of the laser during the emission process. Therefore, the equipment was configured to measure a specific point at the reference sample's surface at 1450 nm during 1000 cycles. Each cycle corresponds to the time between a trigger sent to the emission of the laser pulse and the acquisition of the reflected light by the Andor spectrometer, which took 200 ms in total. The results are presented in Figure 7.17.

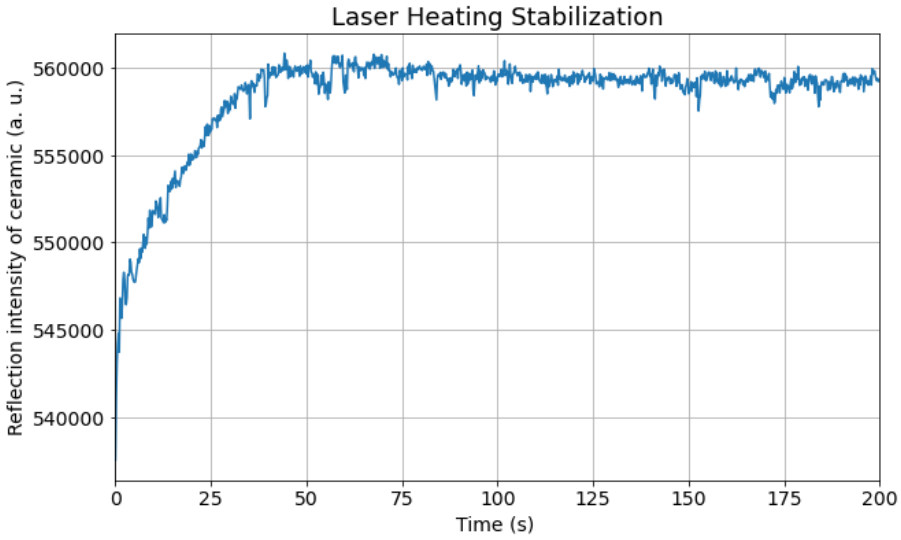


Figure 7.17: Time for stabilization of the amplitude measurement of a single wavelength.

From the results, it is possible to observe that the system took approximately 75 seconds to stabilize, and a previous heating of the laser was necessary to obtain more reliable data. In addition, the results also have shown a noise around the stabilized amplitude. This noise can mislead the algorithm during the classification, however, the amplitude of the noise was not enough to cause the fluctuations presented in Figure 7.16. A further investigation was performed on the exposure time of the Andor spectrometer. The spectrometer allows the configuration of different parameters which can optimize the performance of the prototype. The exposure

time corresponds to the period that the spectrometer sensor counts the amount of photons received. A higher exposure time represents a higher amplitude, however the amount of noise acquired by the sensor is also higher.

In addition, the exposure time is directly related to the amount of time needed to perform the amplitude measurement of a single wavelength. Hence, it strongly influences the period necessary for the acquisition of the entire multispectral point cloud. For instance, to acquire a point cloud with 101 x 101 points, measuring 5 wavelengths at each point using an exposure time of 1 second, would require 51005 seconds, or approximately 14 hours to be completed, without considering the time needed for moving the moving mirror and triggering the acquisition equipment. Therefore, the exposure time of 100 ms was used in the analysis of the experiments performed in this work, given the balance between lower noise, and faster acquisition. An optimization of the exposure time and its relation to the accuracy of the models is subject to further investigation.

Another important aspect that is characteristic of practical laser systems is the emission bandwidth. As stated in Chapter 2.2, a practical laser emits light that is collimated and in phase in a narrow bandwidth centered in a specific wavelength. This emission pattern is different from the light emitted by the tungsten light source of the FT-IR spectrometer, which is dispersed in all directions and has a broad bandwidth. The possible effects of this difference in emission were analyzed through the simulation of the emission patterns of the laser. Therefore, the subset of wavelengths selected by the model chosen in Section 7.3 were considered. This model used 5 wavelengths at 1301.91 nm, 1399.67 nm, 1412.65 nm, 1448.18 nm, and 1492.4 nm. In a practical multispectral system, these wavelengths should correspond to the central emission of the spectral waveform of each monochromatic laser.

Since the spectral fingerprint of a laser system can be approximated by a Gaussian function, with a central wavelength and an emission bandwidth, corresponding to the FWHM of the system [31]. These emission patterns were simulated considering the central wavelengths of the selected model and 10 nm of FWHM of the laser system, which is an approximation of the value measured in the multispectral LiDAR demonstrator. The simulated emission pattern of the multispectral LiDAR with the 5 wavelengths selected is shown in Figure 7.18 .

The measurement principle of the amplitude of the light reflected by the target in a LiDAR system is also different from the FT-IR spectrometer. The first measures the light reflected in the entire narrow bandwidth of the waveform emitted, by integrating the spectrum of the reflected light, to compute the amplitude of the central wavelength. On the other hand, the FT-IR spectrometer measures the spectrum by changing the length of the light path and calculating the Fourier Transform of the interferogram [55], resulting in a more precise spectral measurement given its higher resolution.

Since the resolution of the two spectral acquisition techniques is different, the simulation was performed to analyze the influence of the bandwidth, using a perfect gaussian shaped light emission of a laser source, in the spectral fingerprint of the sample under analysis. The reflected light acquired by a LiDAR system corresponds

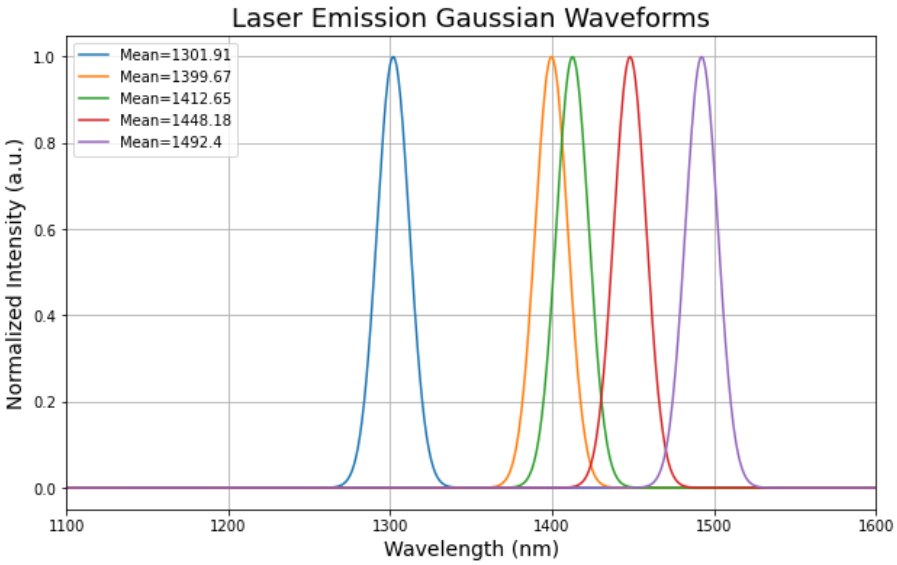


Figure 7.18: Simplified simulated spectral emission of a multispectral LiDAR.

to the waveform characteristics of the laser emission convoluted with the spectral characteristics of the target response with the addition of the system noise [17].

For the sake of simplicity, the simulation considered the target response as the spectral fingerprint of the sample acquired using the high resolution measurement of the spectrometer. In a real scenario, there is the influence of other factors and physical phenomena that are challenging to simulate, such as the width of the laser beam, the optical aperture of the detector, the shape and distance of the scatters present in the sample's surface area illuminated by the laser beam, the noise of the system and atmospheric transmission factors [17]. The experiment conducted aimed to evaluate the influence of a narrow bandwidth in the shape of the spectrum in the ideal case without considering the influence of these parameters.

It started with the selection of one spectral sample, which was convoluted with the emission spectrum waveform of one laser source centralized at the selected wavelengths. Since the spectrum is in the frequency domain, the convolution of both spectral waves is calculated through the multiplication of their spectrum. Afterwards, the convoluted signal is integrated. The value of the integration represents the intensity of the light reflected to the LiDAR system at the central wavelength of the laser emission. An example of the steps performed during the simulation of the spectral LiDAR measurements at 1301.91 nm for an organic sample is detailed in Figure 7.19.

The steps are repeated for all the 5 wavelengths selected, using the simulated waveforms presented in Figure 7.18. Finally, the resulting spectral fingerprint is normalized for zero mean and standard deviation, and a comparison with the normalized spectrum measured by the spectrometer is performed. The results are shown in Figure 7.20.

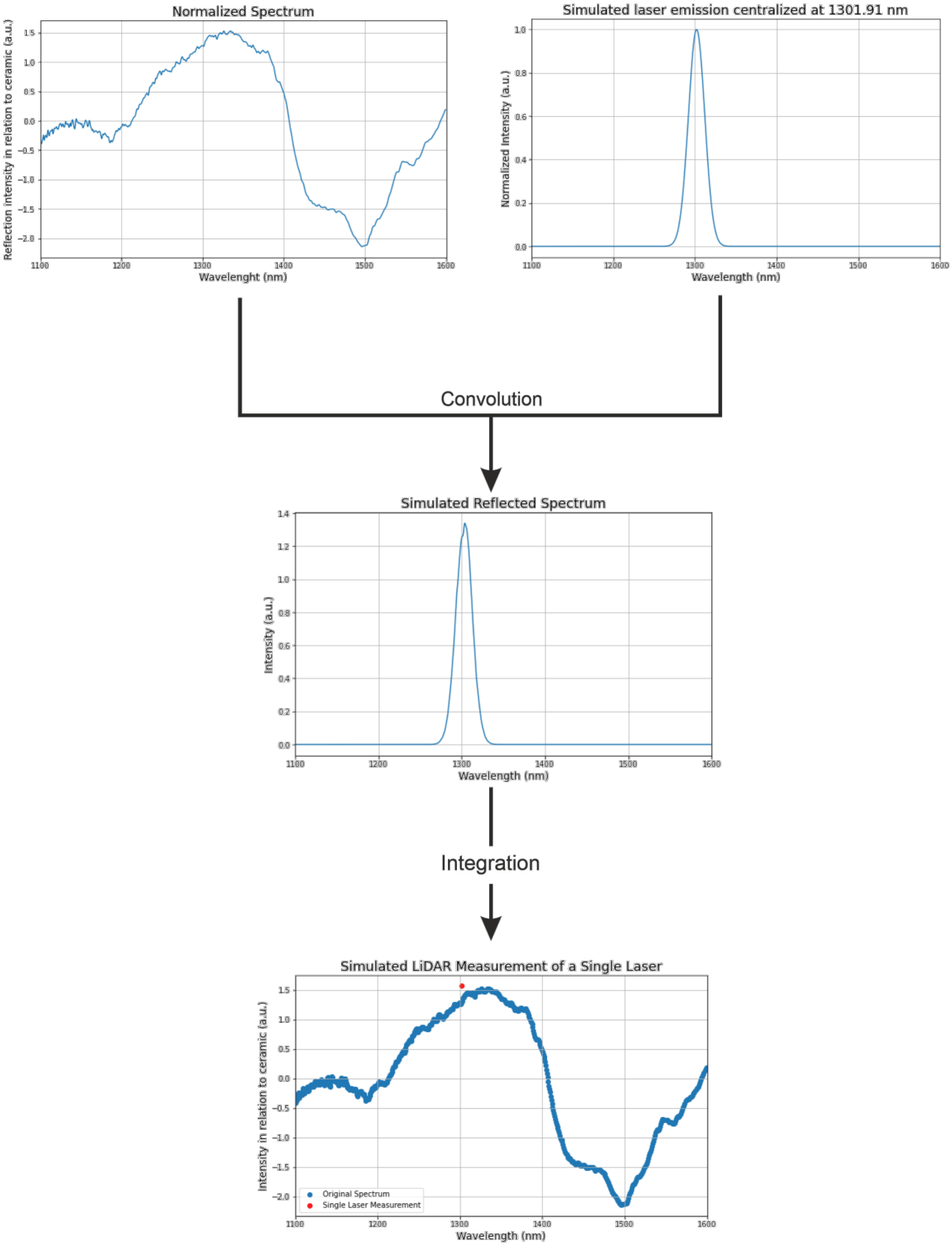


Figure 7.19: Schematic diagram with an example of the simulation of one LiDAR measurement.

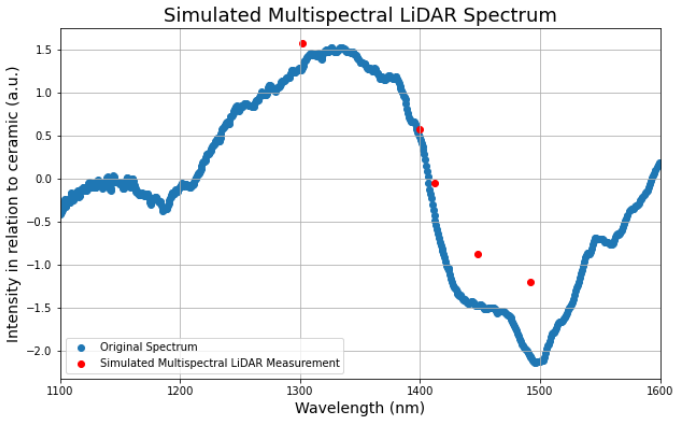


Figure 7.20: Comparison between the simulated multispectral LiDAR measurement and the spectrometer measurement.

Even though the graphs do not correspond to an accurate representation of the real spectrum measured by the prototype, and the normalization comparison is not fair given the differences in spectral resolution. These results showed a visual comprehension of the influence of the bandwidth in the spectral data.

From the graph a slight difference between the spectrum measured by the spectrometer and the results simulated considering the bandwidth of the laser emission can be observed. The reason for the difference lies on the convolution of the sample spectrum with the spectral waveform of the laser. This convolution filters the spectrum, smoothing the waveform. Hence, it modifies the original spectral fingerprint of the sample and the intensity values expected as input by the algorithm. This can cause higher classification errors.

During the experiments in the multispectral LiDAR prototype, the influence of the physical characteristics of the system were observed. The detector used by the demonstrator measures the intensity of light acquired in a row of pixels. Therefore, the signal measured by this detector represents the intensity distribution of the light in the beam. For a laser beam, the ideal intensity distribution at the beam should have the shape of a Gaussian function [31]. Hence, a mirror was positioned in front of the moving mirror to redirect the laser beam to the detector. The result for the intensity distribution at the laser beam during the emission at 1301.9 nm is shown in Figure 7.21.

As can be seen in the graph, the shape of the intensity of light distribution at the laser beam is distorted from the ideal Gaussian waveform. This distortion comes from the imperfections of the optical components of the system, such as the optical fibers, mirrors, and beam splitters, that introduce losses and reflections to the entire optical path. Even though the intensity distribution of light was distorted, the overall waveform is very similar to the optimal Gaussian shape. Therefore, the analysis was repeated with a sample of ceramic to observe the behavior of the light intensity pattern in the reflected signal.

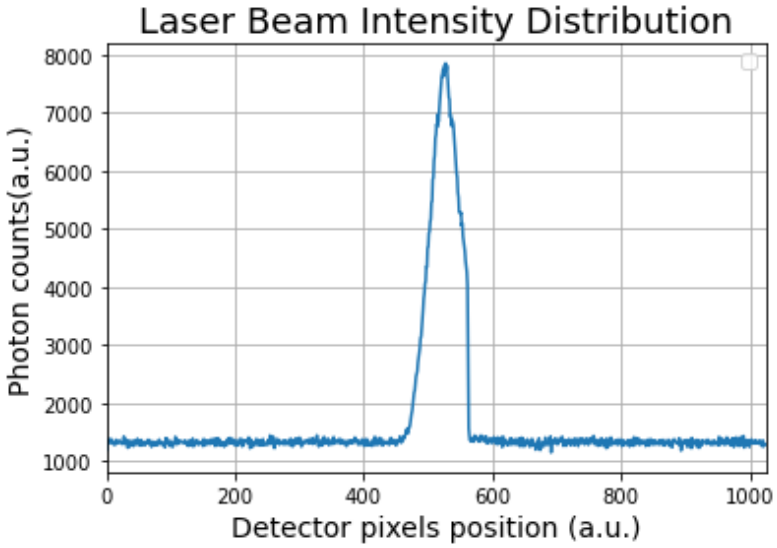


Figure 7.21: Laser beam intensity distribution acquired by the iDUS InGaAS detector.

For this analysis, the mirror was removed, and the ceramic reference sample was placed 10 cm in front of the moving mirror. Afterwards, a single point was measured in the 5 wavelengths selected. The pattern of the intensity distribution of the reflected beam for two wavelengths is shown in Figure 7.22.

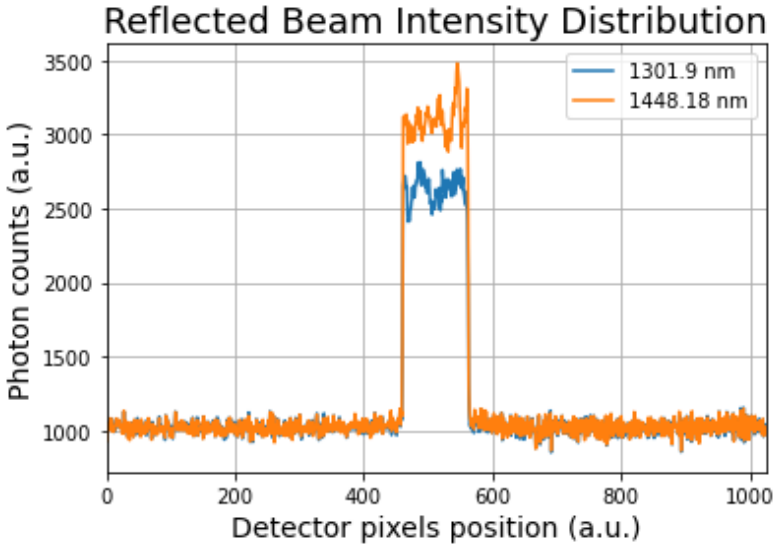


Figure 7.22: Reflected beam intensity distribution acquired by the iDUS InGaAS detector.

In the graph, strong variations in the peak of the signal acquired by the detector can be noted. These variations were observed to be mainly caused by interference

patterns coming from the sample's surface added to the noise present in the system. This assumption was performed given the repeatability of the results, which vary depending on the wavelength. Since the laser beam illuminates an area of the sample's surface, the roughness of the surface and the distribution of the substances in the material may cause spatial interference in the light reflected. An effect known to cause interference in LiDAR measurements is called speckle. The main cause of this effect comes from the roughness of the sample surface that changes the phase of the coherent light emitted by the laser, causing a disturbance in the reflected light [53]. This may have been the main cause of the variations shown in the comparison of Figure 7.16, since there are no disturbances in the spectrum acquired by the FT-IR spectrometer, which uses a tungsten light source that emits incoherent light. However, a detailed investigation of the sources of the variations presented is subject to further studies.

Following the evaluation of the factors that contribute to the influences observed in the spectral measurement, the first experiment, in which the results are described in Figure 7.16, was repeated considering the parameters analyzed in the presented results. Therefore the reference sample was positioned 10 cm in front of the moving mirror, and a measurement was carried out by pre heating the laser system, and using an exposure time of 100 ms. In addition, it was observed that averaging multiple spectral measurements decreased the noise of the spectrum. Hence, one point of the sample was measured 10 times using a high resolution spectral acquisition of 1000 wavelengths. This procedure was repeated for the organic sample and the resulting reflectance was compared with the results obtained by the FT-IR spectrometer. The comparison is shown without the mean of the waveforms in Figure 7.23.

The results show that the heating of the laser system before the acquisition process and the use of averaging increase the comparability of the spectral data. In addition, the variations are still present in the spectrum acquired, which reinforces the hypothesis that they come from the spatial characteristics of the sample. Furthermore, the experiment has shown that the spectrum can benefit from fine-tuning the parameters of the system, such as the exposure time, the laser power, and the aperture of the detector.

7.4.2 Grid Spectrum

A following evaluation was performed to verify the influence of the spatial position of the sample's surface on the spectral data. Therefore, the ceramic sample was placed in front of the moving mirror at a distance of 10 cm. The five wavelengths selected by the model elected in Section 7.3 were measured 300 times, at a fixed position, to generate an averaged reference spectrum. Subsequently, the reference sample was changed for the organic sample 1, and a grid pattern of 5 x 5 points was measured on the surface of the sample using the same wavelengths previously described. These wavelengths were measured 3 times to allow the average of the spectrum at each measured point. Hence, each position at the surface of the sample was measured 3 times with the 5 wavelengths selected, generating 3 similar multispectral point

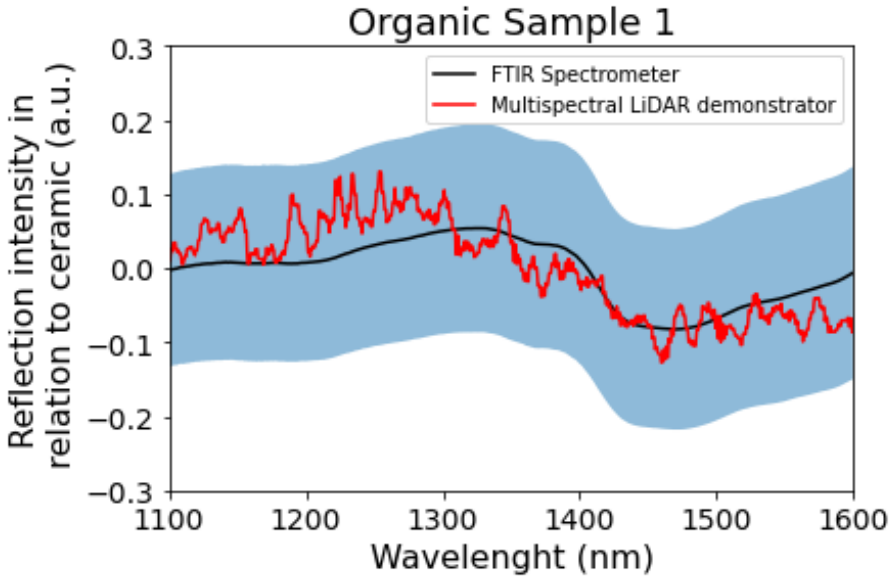


Figure 7.23: Spectral reflectance comparison between the multispectral LiDAR demonstrator and the FT-IR spectrometer for an organic sample centered at zero.

clouds using the spectral acquisition mode of the prototype. The experiments were performed by heating the laser system before the spectral acquisition of each sample and using an exposure time of 100 ms.

The reflectance was calculated by dividing the spectrum of the organic sample by the reference sample. To evaluate the influence of the average in the spectral results, the 3 measurements of the spectrum at a single point at the same position on the sample's surface was compared. The results are shown in Figure 7.24

The results have shown a high repeatability of the spectrum measured at the same position of the sample. Hence, the 3 measurements of the spectrum were averaged for each point. A following analysis of the variations of the measurements over the surface of the sample was performed. Therefore, the reflectance of 3 neighbor points of the acquired point cloud were compared, the results are shown in Figure 7.25.

The experiments have exposed a strong dependence of the reflectance measurements on the surface of the sample. Comparing the spectrum acquired at different points, the results were varied not only in amplitude but also in the waveform of the spectral fingerprint. These results reinforce the hypothesis that interference effects coming from the roughness of the sample's surface, such as the speckle and the variations observed in Figures 7.16 and 7.22, affect directly the spectrum measurements. In addition, the angle of incidence of the laser beam on the surface of the sample can also contribute to the variations observed.

One technique that can reduce the effects of these fluctuations is the average of the spectrum from neighboring points. This assumption was evaluated by averaging the amplitude of each wavelength from all the points of the 5 x 5 grid measured.

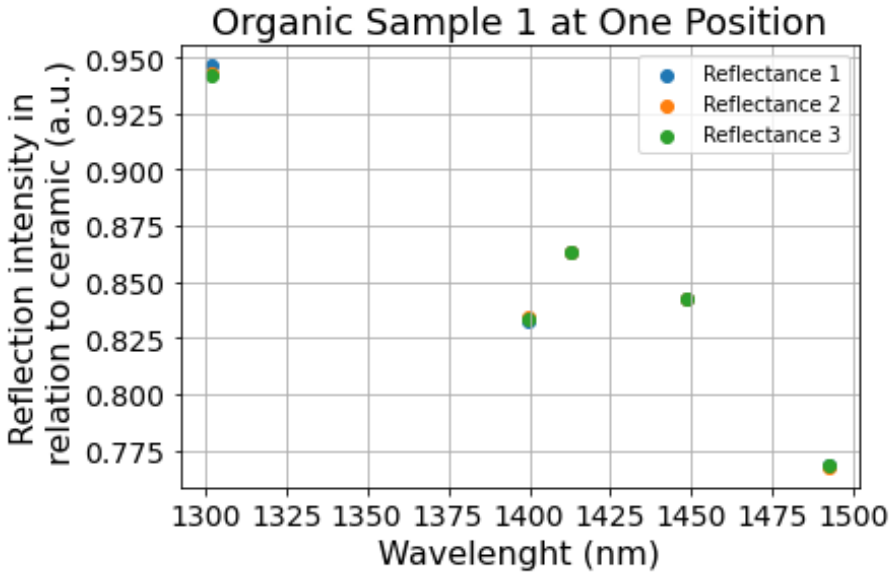


Figure 7.24: Reflectance at a fixed position on the surface of the organic sample for 3 repeated measurements.

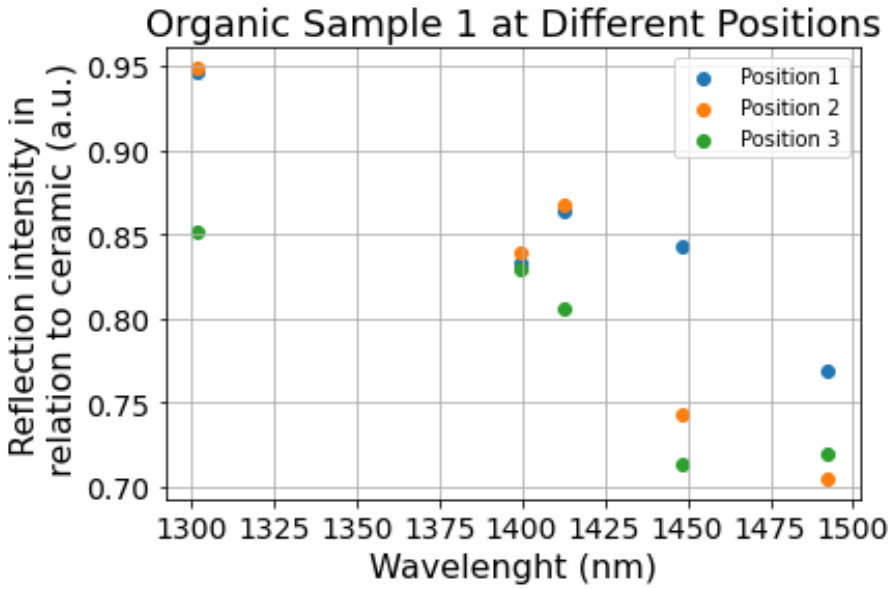


Figure 7.25: Reflectance at different positions on the surface of the organic sample.

The resulting averaged reflectance was compared to the spectrum measured by the FT-IR microscope for the sample of organic. The final comparison is shown without the mean in Figure 7.26. The black line represents the mean and the blue cloud the standard deviation of the spectrum measured with the FT-IR spectrometer centered at zero. The red dots represent the mean, and the vertical gray lines the standard deviation of the measurements of the spectrum in the grid of 5x5 points by the

multispectral LiDAR demonstrator, centered at zero for comparison.

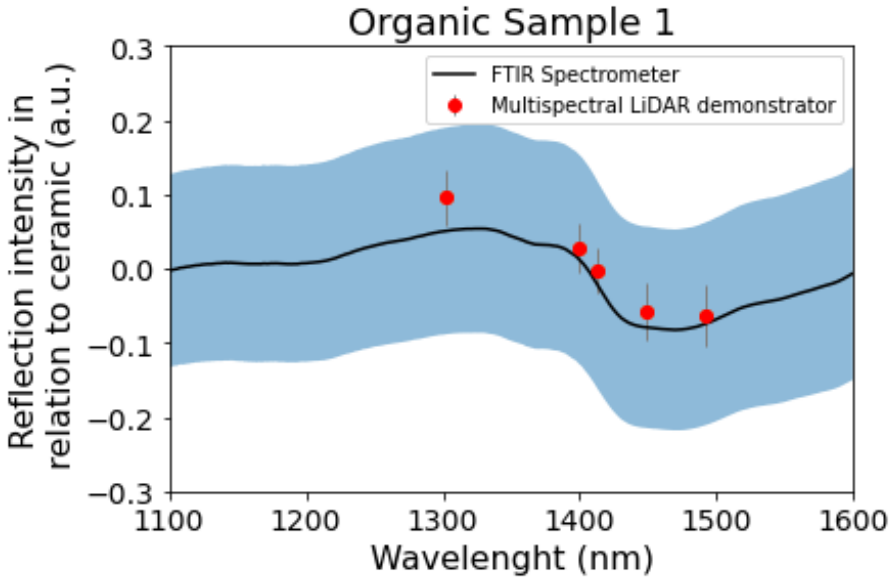


Figure 7.26: Comparison between the spectrum measured by the FT-IR spectrometer and the averaged spectrum of 5x5 points measured by the prototype centered at zero.

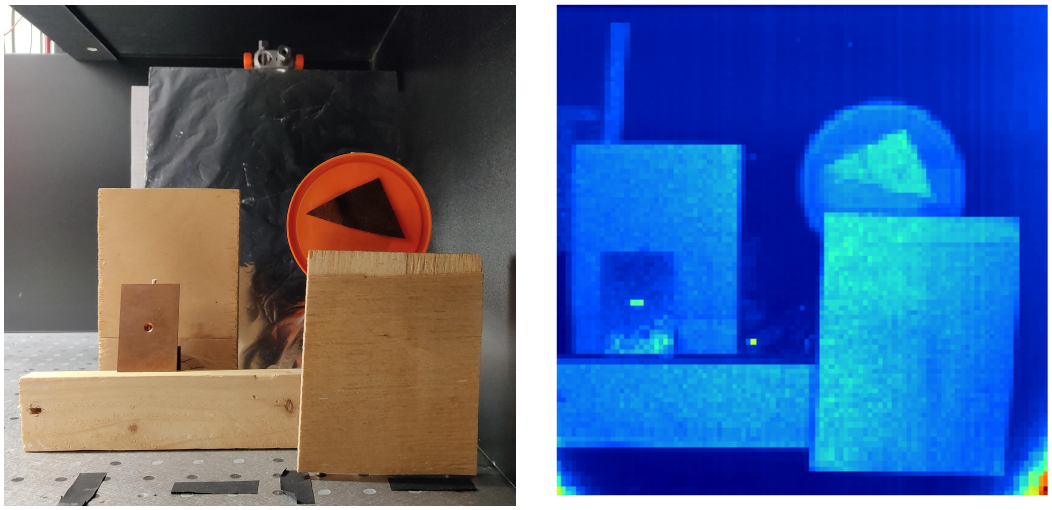
The graph has shown that averaging the spectrum between neighbor points is a promising approach to decrease the effects of the interference caused by the surface of the sample. However, the improvement obtained in the spectral waveform by the evaluations presented in this Section was analyzed in an organic sample with a flat surface, optically dense, and with high reflectivity in the SWIR region of the spectrum. During the experiments, it was observed that the spectrum of other materials that have lower reflectivity in this region or are not optically dense suffer strong deformations that modifies entirely their fingerprint waveform, such as the fabrics that are composed of intertwined fibers.

7.4.3 Multispectral Point Cloud

After the analysis of the behavior of the spectral data acquired by the multispectral LiDAR demonstrator, the scenario described in Section 6.5 was created at 45 cm of the moving mirror. Before the acquisition of the point cloud of the scene, the laser system was heated and the demonstrator was configured to use an exposure time of 100 ms and 60% of the maximum laser power. Hence, the reference sample was positioned in front of the scenario at 40 cm of the moving mirror, and a reference spectrum for the subset of the 5 wavelengths was measured 4 times at a single position to generate an averaged reference.

In a second step, the reference sample was removed from the scenario, and the laser system was heated again using the same previous configuration. Afterwards,

an image of 101 x 101 points was generated, with the subset of 5 wavelengths measured at each point. Following the acquisition of the spectral data, the spectrum of each spatial point measured was divided by the reference spectrum to obtain the reflectance data of the scenario. Figure 7.27 shows the comparison between the scenario created in the visible light, and the reflectance data measured by the multispectral LiDAR demonstrator in the SWIR at 1301.91 nm. For the visualization, an intensity color scheme was applied.



(a) Visible image of the scenario.

(b) 2D SWIR multispectral point cloud measured.

Figure 7.27: Comparison between the scenario created and the 2-dimensions of the SWIR multispectral point cloud acquired at 1301.91 nm.

Following the acquisition of the spectral data, the distance mode of the prototype was configured and the ToF measurements were acquired for the generation of the 3D multispectral point cloud. Afterwards, a copy of the results presented in Figure 7.27(b) was modified using a standard image editing software, and a color mapping was created to represent each material of the scenario. This annotated image was used as ground truth for the classification analysis. Then, the spectrum of each spatial point of the point cloud was normalized and classified using the algorithm selected in Section 7.3. The classification results at each point were compared with the material of the corresponding pixel at the annotated image to compute the miss classification rate. Subsequently, the accuracy of the model in the entire image was computed, by dividing the number of correct predictions by the total number of predictions on the image, and a color mapping was created to represent the classes of materials in 4 different colors.

The classified 3D point cloud of the scenario is shown in Figure 7.28, where the class predicted for each point of the multispectral point cloud is represented by its respective color. The accuracy computed for this scenario, dividing the number of

correct predicted points by the total number of predicted points, was 78.43%. This result was obtained without the utilization of the average of neighbor points.

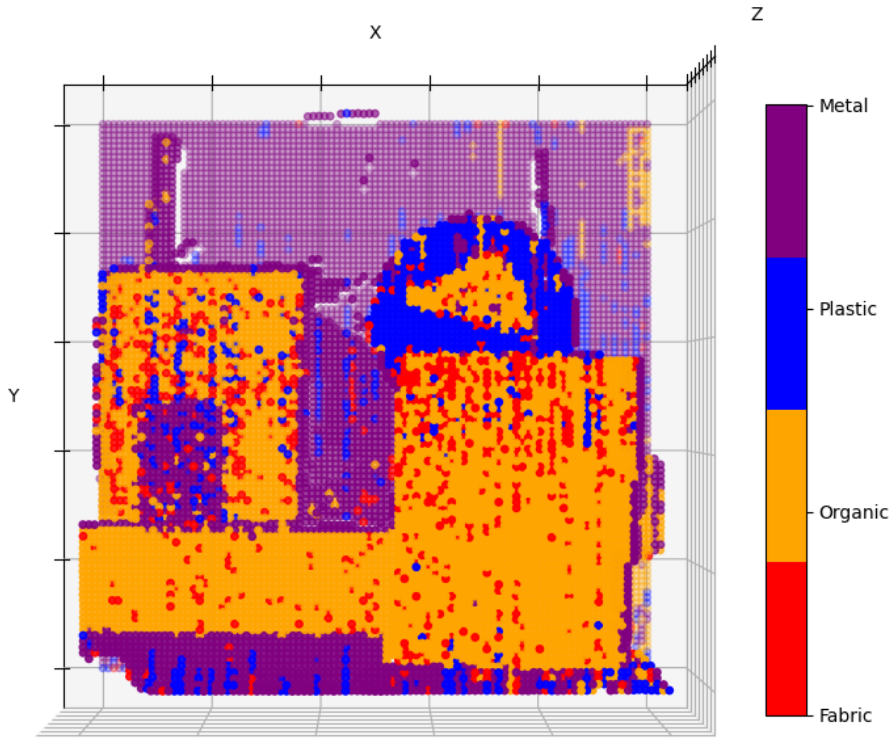


Figure 7.28: Front view of the classified multispectral 3D point cloud of the scenario.

The results have shown that the approach adopted in this work is promising for the identification of materials in multispectral LiDAR point clouds. From the image presented in the Figure, it is possible to observe that the classification model using the amplitude of 5 selected wavelengths could visually distinguish objects with different spectral fingerprints. However, it is also possible to observe the influence of the spectral variations in the classification, which appears as noise in the image, and the confusion made by the algorithm during the classification of the fabrics and organics classes. The fabric sample used in this scenario was a segment of leather commonly found in clothes, enumerated with the number 1 in Figure 6.1(b). It was glued on top of a plastic sample, and the result demonstrated that the algorithm was capable of distinguishing between the spectrum fingerprints of the two samples. However, since leather is also an organic material, the fluctuations of the spectral data may have led the algorithm to misclassify the sample. In addition, the experiments have shown that the samples selected for the fabric class are challenging to distinguish, given their low reflectivity and optical density, which may have interaction properties with the collimated laser beam different from a highly reflective flat surface. An investigation of these interactions can be a subject for further studies. The performance in the classification of the model should also benefit from a more

distinctive choice of the spectral fingerprints of the samples for each class. For instance, using synthetic samples with defined substance compositions or evaluating the model in different regions of the spectrum.

A final evaluation was performed by averaging the spectrum acquired between the neighboring spatial points, given the possible contributions for the decrease in the fluctuations of the data as discussed in Section 7.4.2. Therefore a moving average was performed in each wavelength of the spectrum acquired using a grid of 4 x 4 neighbor points. Hence, the resulting average multispectral point cloud was classified again and the results are presented in Figure 7.29. For better visualization of the 3D measurements capabilities in distance measurements of the multispectral LiDAR demonstrator, Figure 7.30 shows a rotated view of the classified scenario, and the miss classification rate of the model is presented in Figure 7.31, where the correct classified points are shown in green and the misclassified points in red.

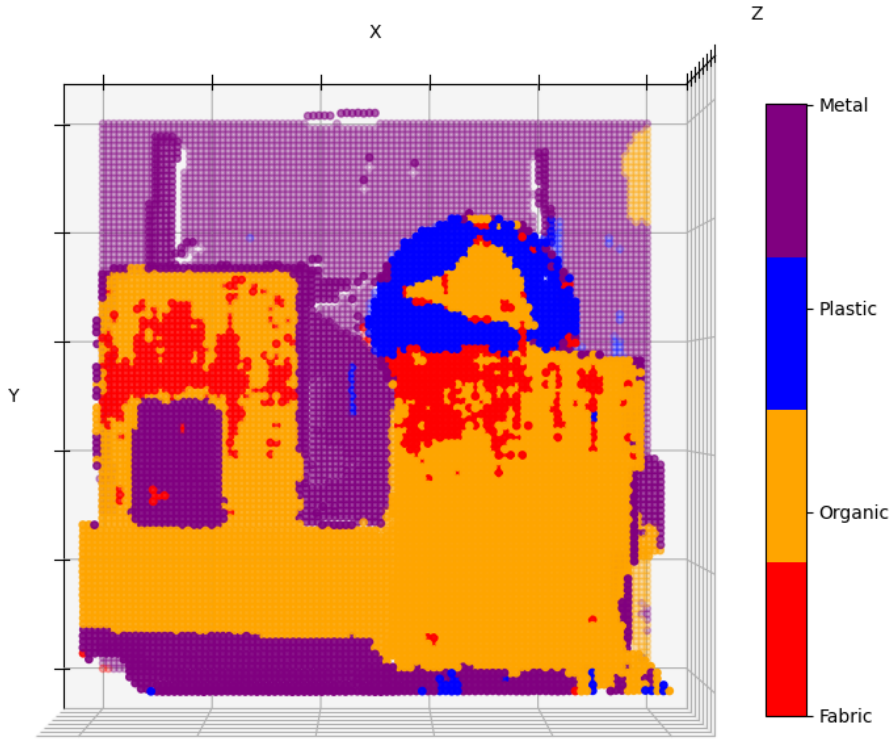


Figure 7.29: Front view of the classified multispectral 3D point cloud of the scenario after spectral average.

The accuracy of the classification in the scenario was increased to 84.77% using the moving average of the spectrum. These results show a promising combination of the use of image processing techniques for the classification of materials in multispectral LiDAR point clouds.

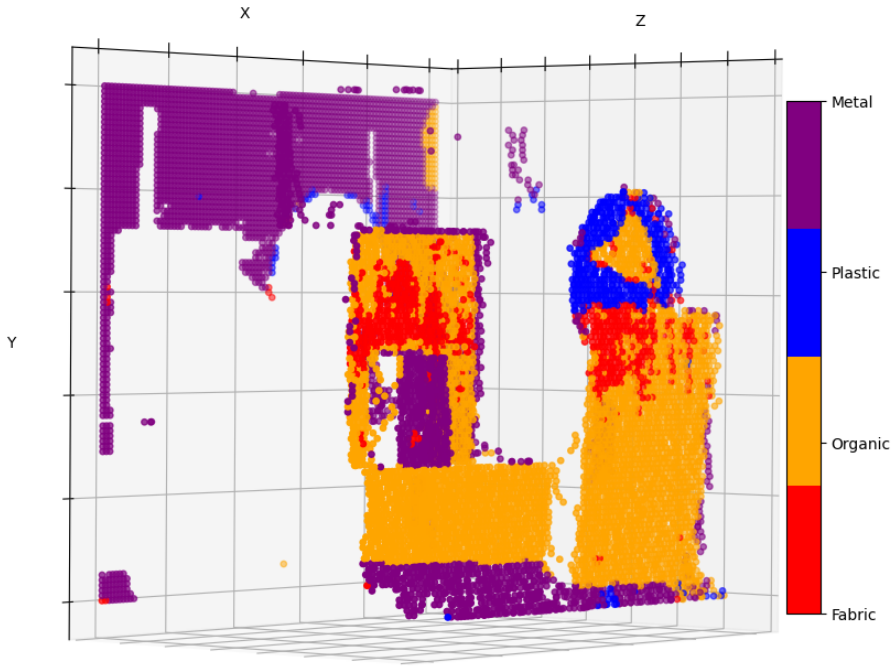


Figure 7.30: Rotated view of the classified multispectral 3D point cloud of the scenario after spectral average.

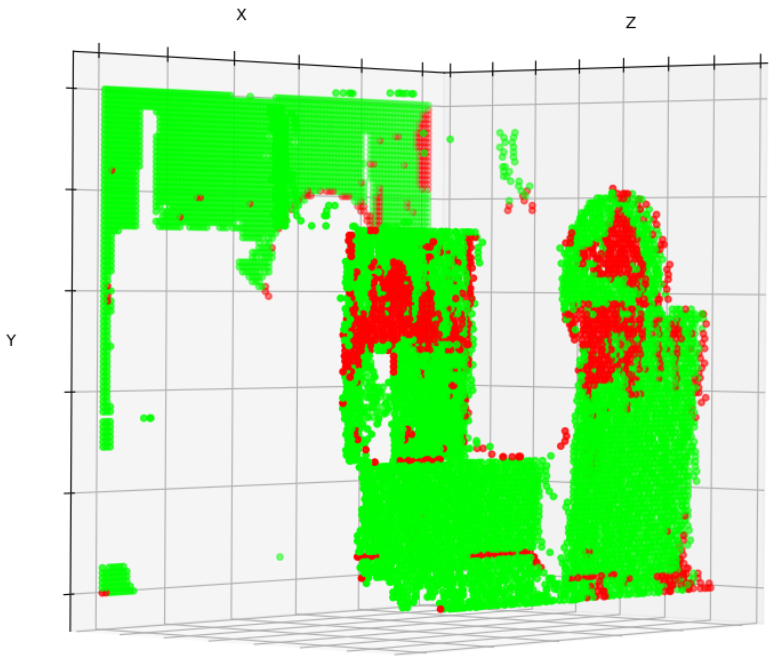


Figure 7.31: Rotated view of the miss classification of the classified 3D point cloud scenario.

8 Conclusion

8.1 Conclusion

The classification of materials using spectral information from multispectral LiDAR sensors is a complex and challenging task. Many factors interfere in the development of hardware and software for the large scale deployment of these systems, and ongoing and future research is still necessary for the entire understanding of the behavior of these sensors.

This work has addressed some of the challenges by evaluating the feasibility of using a commercially available FT-IR spectrometer to create a spectral dataset of target materials, and analyzing the performance of machine learning models before the availability of the multispectral LiDAR hardware. This approach has shown promising results in helping to isolate and demystify the behaviors of the system.

In this work, a dataset with the spectrum of four classes of materials commonly found in urban environments in the SWIR region was created, and a comparison of their spectral fingerprints was presented. In addition, the importance of normalizing the spectrum for zero mean and unity variance, before the training of the models, to avoid the influence of the offset was exposed. Then, a detailed comparison of L1-regularized logistic regression and random forest was performed without wavelength selection.

After, the comparison of the feature selection capabilities of both models was performed. Therefore, the random forest was first optimized for different forest sizes. The results have shown the instability of the feature importance of the random forest for different forest sizes and the increase in memory storage for models with large forests. Subsequently, the forest with the smallest size to achieve a high accuracy after comparison was placed inside the recursive feature elimination wrapper for the automatic feature selection. Hence, a comparison of the accuracy of 100 versions of the models using a maximum number of 100 wavelengths was performed. The random forest presented the highest accuracy independently of the number of wavelengths used, and one version of each model was selected for a detailed comparison considering the demonstrator characteristics. The results have shown that the selected model of the random forest achieved an accuracy 30.4% higher than the selected model of the L1-normalized logistic regression, with the drawback of performing a prediction 60 times slower and occupying extra an 0.7 MB in storage memory.

After the analysis, the optimized version of the random forest with 5 wavelengths was selected to be tested in the multispectral LiDAR demonstrator. The evaluation

started by comparing the spectral data measured by the demonstrator, in a single point, with the data measured with the FT-IR spectrometer. A variation pattern was observed in the prototype's data, and a simulation of the behavior the the laser bandwidth was performed to better understand the effects. The results have shown the influences of the parameters of the demonstrator on the quality of the spectral data, and reinforced that an optimization of the data acquired by the demonstrator improves the performance of the classification. In addition, an analysis of the spectrum at different points on the surface of the sample was performed, where the influence of the roughness of the sample in the spectral data was exposed. Furthermore, the averaging of the spectrum between close points has shown to be promising in reducing spectral variations.

Finally, an example scenario was created and the multispectral point cloud was acquired and classified by the selected model, which predicted 78.43% of the points correctly. The results were further improved to 84.77% by averaging the spectrum between neighboring points, proving that the use of image processing techniques can be a promising approach to improve the classification of materials. However, the results obtained in this study are valid for the materials selected and inside the spectral range considered. Further investigation is still necessary to understand the confusion of the system between organic and fabric samples and the effects of different regions of the spectrum that could not be measured with the demonstrator. Meanwhile, the results have shown that the system may benefit from materials with high reflectivity in the region analyzed and optically dense.

8.2 Future Work

There are many challenges that have to be solved for the practical implementation of multispectral LiDAR systems, mainly in the embedded systems domain. Future works may address the behavior of the interactions of the laser beam with materials that are not optically dense, such as fabrics with intertwined fibers, and their influence in the reflected light acquired by the detector. Others can evaluate the influence of materials with a mix of substances, such as rocks, in the balance of the dataset.

For real world scenarios, it is also important that the algorithms consider the background or utilize attention based mechanisms to avoid its classification. In addition, further studies may benefit from the use of materials with known concentrations of substances to address micro classes, such as identifying different types of plastic inside the class plastics for recycling applications.

Since this work only evaluated the classification performance using subsets of single wavelengths, further studies should investigate the behavior of the models using groups of wavelengths, and wavelets, or analyze the influence of single wavelengths inside the same bandwidth of the laser.

Another possible topic of evaluation is the influence of other image processing techniques to improve the quality of the spectrum. In this work the average of

neighboring points was performed without optimization. Further, studies can evaluate this influence for different numbers of points in the average and different filtering techniques.

The use of macro classes also opens the opportunity for studies in the field of reconfigurable systems, that use tunnable optical filters to optimize the classification during the interaction with different environments. In addition, a comparison with other machine learning techniques, such as neural networks with L1-regularization, can be performed. Therefore, an increase in the dataset size is necessary.

Bibliography

[1] Abdulrahman, F.H.: Hyperspectral and lidar data fusion in features based classification. Arabian Journal of Geosciences 14, 1–18 (2021)

[2] Anaconda: Anaconda software distribution (2020), `https://docs.anaconda.com/`, accessed on January 22, 2024

[3] Bishop, C.: Pattern recognition and machine learning. Springer google schola 2, 5–43 (2006)

[4] Bolón-Canedo, V., Sanchez-Marono, N., Alonso-Betanzos, A.: Feature Selection for High-Dimensional Data. Artificial Intelligence: Foundations, Theory, and Algorithms, Springer, Cham (2015), description based upon print version of record

[5] Bragato, G., Piccolo, G., Sattier, G., Sada, C.: Identification of spectral responses of different plastic materials by means of multispectral imaging. Environ. Sci.: Processes Impacts pp. – (2024)

[6] Breiman, L.: Bagging predictors. Machine learning 24, 123–140 (1996)

[7] Breiman, L.: Random forests. Machine learning 45, 5–32 (2001)

[8] Breiman, L.: Manual on setting up, using, and understanding random forests v3. 1. Statistics Department University of California Berkeley, CA, USA 1(58), 3–42 (2002)

[9] Bruker: Guide for infrared spectroscopy, `https://www.niu.edu/clas/chembio/_pdf/analytical-lab/ftir/Buker-FTIR-guide.pdf`, accessed on April 23, 2024

[10] Bruker: Vertex 70v ft-ir spectrometer, `https://www.bruker.com/en/products-and-solutions/infrared-and-raman/ft-ir-microscopes/hyperion-ii-ft-ir-and-qcl-microscope.html`, accessed on November 10, 2023

[11] Budei, B.C., St-Onge, B., Fournier, R.A., Kneeshaw, D.: Effects of viewing geometry on multispectral lidar-based needle-leaved tree species identification. Remote Sensing 14(24) (2022)

[12] Chen, B., Shi, S., Gong, W., Sun, J., Chen, B., Du, L., Yang, J., Guo, K., Zhao, X.: True-color three-dimensional imaging and target classification based on hyperspectral lidar. Remote Sensing 11(13), 1541 (2019)

[13] Chen, B., Shi, S., Gong, W., Xu, Q., Tang, X., Bi, S., Chen, B.: Wavelength selection of dual-mechanism lidar with reflection and fluorescence spectra for plant detection. Opt. Express 31(3), 3660–3675 (Jan 2023)

[14] Chen, B., Shi, S., Gong, W., Xu, Q., Tang, X., Bi, S., Chen, B.: Wavelength selection of dual-mechanism lidar with reflection and fluorescence spectra for plant detection. Opt. Express 31(3), 3660–3675 (Jan 2023)

[15] Chen, Y., Jiang, C., Hyyppä, J., Qiu, S., Wang, Z., Tian, M., Li, W., Puttonen, E., Zhou, H., Feng, Z., Bo, Y., Wen, Z.: Feasibility study of ore classification using active hyperspectral lidar. IEEE Geoscience and Remote Sensing Letters 15(11), 1785–1789 (2018)

[16] Erickson, Z., Xing, E., Srirangam, B., Chernova, S., Kemp, C.C.: Multimodal material classification for robots using spectroscopy and high resolution texture imaging. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10452–10459. IEEE (2020)

[17] Fang, J., Wang, Y., Zheng, J.: Gaussian convolution decomposition for non-gaussian shaped pulsed lidar waveform. Measurement Science and Technology 34(3), 035203 (dec 2022), https://dx.doi.org/10.1088/1361-6501/aca3c6

[18] Ge, Y., Atefi, A., Zhang, H., Miao, C., Ramamurthy, R.K., Sigmon, B., Yang, J., Schnable, J.C.: High-throughput analysis of leaf physiological and chemical traits with vis–nir–swir spectroscopy: a case study with a maize diversity panel. Plant Methods 15(1), 66 (Jun 2019)

[19] GetSpec: getspec std 5101 cin - reflectance standard, http://www.getspec.com/www/getspec.nsf/main.html?open&lang=EN&id=getRefleX_EN, accessed on March 12, 2024

[20] Gimmestad, G.G., Roberts, D.W.: Lidar Engineering: Introduction to Basic Principles. Cambridge University Press (2023)

[21] Grandini, M., Bagli, E., Visani, G.: Metrics for multi-class classification: an overview. arXiv preprint arXiv:2008.05756 (2020)

[22] Guo, Q., Su, Y., Hu, T.: LiDAR principles, processing and applications in forest ecology. Academic Press (2023)

[23] Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine learning 46, 389–422 (2002)

[24] Hirasawa, K., Ho Yeap, K.: Electromagnetic Fields and Waves. IntechOpen (2019)

[25] Hunter, J.D.: Matplotlib: A 2d graphics environment. Computing in Science & Engineering 9(3), 90–95 (2007)

[26] Islam, M.: Autonomous systems revolution: Exploring the future of self-driving technology. Journal of Artificial Intelligence General science (JAIGS) ISSN:3006-4023 3(1), 16–23 (Feb 2024)

[27] James, G., Witten, D., Hastie, T., Tibshirani, R., et al.: An introduction to statistical learning, vol. 112. Springer (2013)

[28] Jo, K., Lee, S., Kim, C., Sunwoo, M.: Rapid motion segmentation of lidar point cloud based on a combination of probabilistic and evidential approaches for intelligent vehicles. Sensors 19(19) (2019)

[29] Jung, A.: Machine learning: the basics. Machine Learning: Foundations, Methodologies, and Applications, Springer, Singapore (2022)

[30] Kaasalainen, S.: Multispectral terrestrial lidar: State of the art and challenges. Laser Scanning pp. 5–18 (2019)

[31] Kannatey-Asibu Jr, E.: Principles of Laser Materials Processing: Developments and Applications. John Wiley & Sons (2023)

[32] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C.: Jupyter notebooks – a publishing format for reproducible computational workflows. In: Loizides, F., Schmidt, B. (eds.) Positioning and Power in Academic Publishing: Players, Agents and Agendas. pp. 87 – 90. IOS Press (2016)

[33] Kuprowski, M., Drozda, P.: Feature selection for airbone lidar point cloud classification. Remote Sensing 15(3) (2023)

[34] Lee, S.I., Lee, H., Abbeel, P., Ng, A.Y.: Efficient l˜ 1 regularized logistic regression. In: Aaai. vol. 6, pp. 401–408 (2006)

[35] Li, X., Wang, L., Guan, H., Chen, K., Zang, Y., Yu, Y.: Urban tree species classification using uav-based multispectral images and lidar point clouds. Journal of Geovisualization and Spatial Analysis 8(1), 5 (2024)

[36] Manakkakudy, A., De Iacovo, A., Maiorana, E., Mitri, F., Colace, L.: Waste material classification: A short-wave infrared discrete-light-source approach based on light-emitting diodes. Sensors 24(3) (2024)

[37] McClarren, R.G.: Machine Learning for Engineers. Springer (2021)

[38] Wes McKinney: Data Structures for Statistical Computing in Python. In: Stéfan van der Walt, Jarrod Millman (eds.) Proceedings of the 9th Python in Science Conference. pp. 56 – 61 (2010)

[39] Murphy, K.P.: Probabilistic Machine Learning: An introduction. MIT Press (2022)

[40] Mutz, Y.S., do Rosario, D., Galvan, D., Schwan, R.F., Bernardes, P.C., Conte-Junior, C.A.: Feasibility of nir spectroscopy coupled with chemometrics for classification of brazilian specialty coffee. Food Control 149, 109696 (2023)

[41] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)

[42] Pérez, F., Granger, B.E.: IPython: a system for interactive scientific computing. Computing in Science and Engineering 9(3), 21–29 (May 2007)

[43] Puttonen, E., Hakala, T., Nevalainen, O., Kaasalainen, S., Krooks, A., Karjalainen, M., Anttila, K.: Artificial target detection with a hyperspectral LiDAR over 26-h measurement. Optical Engineering 54(1), 013105 (2015)

[44] Quinlan, J.R.: Induction of decision trees. Machine learning 1, 81–106 (1986)

[45] Raschka, S., Liu, Y.H., Mirjalili, V., Dzhulgakov, D.: Machine Learning with Pytorch and Scikit-Learn: Develop Machine Learning and Deep Learning Models with Python. Packt Publishing, Limited, Birmingham (2022)

[46] Ray, P., Salido-Monzú, D., Camenzind, S.L., Wieser, A.: Supercontinuum-based hyperspectral lidar for precision laser scanning. Opt. Express 31(20), 33486–33499 (Sep 2023)

[47] Reddy Cenkeramaddi, L., Bhatia, J., Jha, A., Kumar Vishkarma, S., Soumya, J.: A survey on sensors for autonomous systems. In: 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA). pp. 1182–1187 (2020)

[48] Rogers, M., Blanc-Talon, J., Urschler, M., Delmas, P.: Wavelength and texture feature selection for hyperspectral imaging: a systematic literature review. Journal of Food Measurement and Characterization 17(6), 6039–6064 (Dec 2023)

[49] Shan, J., Toth, C.K.: Topographic laser ranging and scanning: principles and processing. CRC Press, Boca Raton, second edition edn. (2018)

[50] Shao, H., Chen, Y., Li, W., Jiang, C., Wu, H., Chen, J., Pan, B., Hyyppä, J.: An investigation of spectral band selection for hyperspectral lidar technique. Electronics 9(1), 148 (2020)

[51] Shao, H., Chen, Y., Yang, Z., Jiang, C., Li, W., Wu, H., Wen, Z., Wang, S., Puttnon, E., Hyyppä, J.: A 91-channel hyperspectral lidar for coal/rock classification. IEEE Geoscience and Remote Sensing Letters 17(6), 1052–1056 (2020)

[52] Shi, S., Chen, B., Bi, S., Li, J., Gong, W., Sun, J., Chen, B., Du, L., Yang, J., Xu, Q., Wang, F., Song, S.: A spatial–spectral classification framework for multispectral lidar. Geo-spatial Information Science 0(0), 1–15 (2023)

[53] Shlyahtin, K., Shelkovnikov, E.Y.: Study of impact of noises of different nature on range profile formation in lidar system. In: AIP Conference Proceedings. vol. 2605. AIP Publishing (2023)

[54] Taher, J., Hakala, T., Jaakkola, A., Hyyti, H., Kukko, A., Manninen, P., Maanpää, J., Hyyppä, J.: Feasibility of hyperspectral single photon lidar for robust autonomous vehicle perception. Sensors 22(15), 5759 (2022)

[55] Theophanides, T.: Infrared Spectroscopy: Materials Science, Engineering and Technology. IntechOpen (2012)

[56] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) 58(1), 267–288 (1996)

[57] Yeong, D.J., Velasco-Hernandez, G., Barry, J., Walsh, J.: Sensor and sensor fusion technology in autonomous vehicles: A review. Sensors 21(6) (2021)

[58] Zhang, H., Wang, M.: Search for the smallest random forest. Statistics and its Interface 2(3), 381 (2009)

[59] Zhang, W., Kasun, L.C., Wang, Q.J., Zheng, Y., Lin, Z.: A review of machine learning for near-infrared spectroscopy. Sensors 22(24) (2022)

[60] Zhang, Y., Carballo, A., Yang, H., Takeda, K.: Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. ISPRS Journal of Photogrammetry and Remote Sensing 196, 146–177 (2023)

[61] Zhu, C., Kanaya, Y.: Eliminating the interference of water for direct sensing of submerged plastics using hyperspectral near-infrared imager. Scientific Reports 13(1), 15991 (Oct 2023)

[62] Zollanvari, A.: Machine learning with Python: theory and implementation. Springer International Publishing, Cham (2023)

Chemnitzer Informatik-Berichte

In der Reihe der Chemnitzer Informatik-Berichte sind folgende Berichte erschienen:

CSR-21-01 Marco Stephan, Batbayar Battseren, Wolfram Hardt, UAV Flight using a Monocular Camera, März 2021, Chemnitz

CSR-21-02 Hasan Aljzaere, Owes Khan, Wolfram Hardt, Adaptive User Interface for Automotive Demonstrator, Juli 2021, Chemnitz

CSR-21-03 Chibundu Ogbonnia, René Bergelt, Wolfram Hardt, Embedded System Optimization of Radar Post-processing in an ARM CPU Core, Dezember 2021, Chemnitz

CSR-21-04 Julius Lochbaum, René Bergelt, Wolfram Hardt, Entwicklung und Bewertung von Algorithmen zur Umfeldmodellierung mithilfe von Radarsensoren im Automotive Umfeld, Dezember 2021, Chemnitz

CSR-22-01 Henrik Zant, Reda Harradi, Wolfram Hardt, Expert System-based Embedded Software Module and Ruleset for Adaptive Flight Missions, September 2022, Chemnitz

CSR-23-01 Stephan Lede, René Schmidt, Wolfram Hardt, Analyse des Ressourcenverbrauchs von Deep Learning Methoden zur Einschlagslokalisierung auf eingebetteten Systemen, Januar 2023, Chemnitz

CSR-23-02 André Böhle, René Schmidt, Wolfram Hardt, Schnittstelle zur Datenakquise von Daten des Lernmanagementsystems unter Berücksichtigung bestehender Datenschutzrichtlinien, Januar 2023, Chemnitz

CSR-23-03 Falk Zaumseil, Sabrina, Bräuer, Thomas L. Milani, Guido Brunnett, Gender Dissimilarities in Body Gait Kinematics at Different Speeds, März 2023, Chemnitz

CSR-23-04 Tom Uhlmann, Sabrina Bräuer, Falk Zaumseil, Guido Brunnett, A Novel Inexpensive Camera-based Photoelectric Barrier System for Accurate Flying Sprint Time Measurement, März 2023, Chemnitz

CSR-23-05 Samer Salamah, Guido Brunnett, Sabrina Bräuer, Tom Uhlmann, Oliver Rehren, Katharina Jahn, Thomas L. Milani, Güunter Daniel Rey, NaturalWalk: An Anatomy-based Synthesizer for Human Walking Motions, März 2023, Chemnitz

CSR-24-01 Seyhmus Akaslan, Ariane Heller, Wolfram Hardt, Hardware-Supported Test Environment Analysis for CAN Message Communication, Juni 2024, Chemnitz

Chemnitzer Informatik-Berichte

CSR-24-02 S. M. Rizwanur Rahman, Wolfram Hardt, Image Classification for Drone Propeller Inspection using Deep Learning, August 2024, Chemnitz

CSR-24-03 Sebastian Pettke, Wolfram Hardt, Ariane Heller, Comparison of maximum weight clique algorithms, August 2024, Chemnitz

CSR-24-04 Md Shoriful Islam, Ummay Ubaida Shegupta, Wolfram Hardt, Design and Development of a Predictive Learning Analytics System, August 2024, Chemnitz

CSR-24-05 Sopuluchukwu Divine Obi, Ummay Ubaida Shegupta, Wolfram Hardt, Development of a Frontend for Agents in a Virtual Tutoring System, August 2024, Chemnitz

CSR-24-06 Saddaf Afrin Khan, Ummay Ubaida Shegupta, Wolfram Hardt, Design and Development of a Diagnostic Learning Analytics System, August 2024, Chemnitz

CSR-24-07 Túlio Gomes Pereira, Wolfram Hardt, Ariane Heller, Development of a Material Classification Model for Multispectral LiDAR Data, August 2024, Chemnitz

Chemnitzer Informatik-Berichte

ISSN 0947-5125

Herausgeber: Fakultät für Informatik, TU Chemnitz
Straße der Nationen 62, D-09111 Chemnitz