Machine Learning and Multivariate Analysis for Chemical Determination of Suspicious Objects by Laser-Induced Breakdown and Raman Spectroscopies

By

Nathan Trevor Garner
B.Sc

Thesis
Submitted to Flinders University
for the degree of

Master of Science by research
College of science and engineering
11/07/2023

Table of contents

Abstract

Hidden explosives are commonly used as a tactic in modern warfare, it has been one of the largest sources of casualties in the most recent wars that Australia has been involved in. There is a need to find a way to work out if objects are explosives without getting close. Laser-Induced Breakdown Spectroscopy (LIBS) makes use of an optical spectrometer to collect atomic data; and Raman Spectroscopy (RS) makes use of an infra-red spectrometer to collect molecular-level-data about the composition of the target. Even separately, LIBS and RS data can be difficult to interpret quickly by hand. While trying to read both types of data, either combined or in quick succession, a human under pressure is likely to make mistakes. Machine learning (ML) can be used to interpret subtleties within the data of these two systems quickly and accurately with no concern about pressure leading to human error. Different forms of ML have various advantages and effective use cases. Several different ML techniques were considered including, Linear Discriminant Analysis (LDA), K-Nearest Neighbour (KNN), and forms of Artificial Neural Networks (ANNs). The most robust variant was found to be a form of KNN producing high accuracies in a range of conditions including high noise and fewer data points. The other models were found to lose accuracy much more quickly, LDA and ANNs losing accuracy in high noise conditions and ANNs losing accuracy when given low numbers of datapoints. However, LIBS and RS separately are unable to identify every possible type of sample and thus must be combined in a meaningful way. Different methods for data fusion were considered, including low-level concatenation, low-level addition, and high-level fusion of predictions. All of these methods proved nearly as effective at classifying spectra within their dataset as each other but mid-level fusion via Principal Component Analysis (PCA) was found to be most effective. All fusion methods were able to produce 100% accurate classification, however the mid-level fusion method was able to do so using significantly less data than the other methods, thus allowing for the fastest processing with minimal computing resources needed.

Declaration

I Nathan Garner hereby certify this thesis does not include any work previously submitted for any other degree or diploma in any university without acknowledgement. This work will not be submitted for a future degree or diploma without the permission of Flinders University. To the best of my knowledge and belief this thesis does not contain any material previously published or written by another person except where due reference it made in the text.

Signed

Nathan garner
22/12/2022

Acknowledgements

I give thanks first to my supervisors Jamie Quinton, Claire Lenehan, and Ula Alexander for encouraging me and pushing me forward when I felt I'd never be able to force myself finish the interminable writing. As well as more specific thanks for the endless help with that writing returning drafts with more speed than I deserved and more comments than I would have hoped. A further thanks to Ula for collecting and assisting with the curation of my datasets, I know asking for Raman of metals is a bit silly, but you still did it for me and so much else. I also want to thank QUT and DSTG for their support of the project.

On a more personal note, I offer thanks to my parents for supporting me through this process. I never would have gotten here without the support and even the nagging. Not even mentioning the help with grammar from mum or talking ideas and planning with dad. Finally, I thank my friends both those I know at Flinders University and those outside Flinders for support, faith and simply keeping me (mostly) sane while I worked my way through this degree.

Chapter 1 - **The Need to Test Unknown Samples for Energetic Material Identification**

1.1 Energetic Materials and Improvised Explosive Devices

A common tactic in modern warfare is the use of improvised explosive devices (IEDs). They presently make up almost half of the attacks with explosive weaponry in the world and do a great deal more harm to civilians than other explosive weapons [1, 2]. IEDs are often camouflaged as piles of debris or other similar innocuous and irregular objects and left by roadsides and in zones of conflict [1, 3, 4]. IEDs can be made from a wide range of explosive chemicals, collectively referred to as energetic materials that can be used in their construction. These can range from repurposed conventional explosives such as mines and mortar shells, through to common organic compounds such as ammonium nitrate and sugars or inorganic materials such as chlorates and aluminium powder [5, 6]. The varied composition and construction in addition to the camouflage make the detection and identification of IEDs difficult. Ideally, suspected IEDs need to be tested quickly, and from a safe distance in case they are in fact explosive. Current methods for detection of IEDs include the use animals (e.g. sniffer dogs, bees), though scent mimicking biosensors are being developed to replace or augment animals [4, 7]. Alternatively IEDs can be detected underground with the use of radar or metal detecting electromagnetic scanners [8]. A final option is to treat suspicious objects as if the object is an explosive and steps are taken to avoid or defuse it. These methods are typically slow, costly, complex to use or require approaching to an unsafe distance, leaving the operator in danger should the device be triggered [9].

1.2 Determination Requirements

Methods for the detection and identification of energetic materials for unknown and potentially dangerous objects have several requirements. Ideally a detection method should be:

- Able to work at tens to hundreds of meters of standoff range or be able to be performed remotely. Standoff range is preferable as this allows both personnel and equipment to stay clear of potential danger, minimising the risk of damage or death.
- Performable in situ without any preparation or unusual requirements such as an inert gas atmosphere.
- Capable of identifying a wide range of energetic materials
- Return results within 10 seconds and with high accuracy.
- Have low detection limits and thus be able to identify trace amounts of energetic compounds.
- Useable by a moderately skilled operator. While the operator could be trained in a complex procedure, an easy to use, difficult to misinterpret system would be advantageous.

1.3 Advanced Chemical Analysis Methods

A survey of literature reveals that some common analytical techniques that are typically used for identifying an unknown sample are,

- InfraRed (IR) spectroscopy [10, 11], which identifies compounds based on vibrational energy levels
- Mass Spectrometry (often but not always making use of chromatography) [12, 13], which identifies compounds based on the mass of molecular fragments
- Ion Mobility Spectrometry (IMS) [14], which also makes use of molecular fragments
- Gas sensors [15, 16] which monitor the atmosphere for energetic vapour in a variety of ways.
- Raman Spectroscopy [10, 17], another technique exploring vibrational energy levels
- Laser-Induced Breakdown Spectroscopy (LIBS) [18], which identifies the elements within a sample by plasma light and

Each of these techniques has advantages and disadvantages for this application, as follows.

IR spectroscopy can be performed with minimal sample preparation and at standoff ranges. Standoff IR involves the use of an illuminating heat source and absorbing the emitted radiation [19, 20]. IR has been used before for a similar standoff identification application [11] and gives fast accurate molecular information about the target samples. Matrix effects are more prevalent and detection limits are not as low as some other methods requiring more of an energetic to be present to produce a signal [21]. IR spectroscopy is potentially useful for energetic material detection applications but would likely require a second technique for inorganic samples.

Mass spectrometry generally requires samples to be separated so each type of molecule in the sample can be analysed individually. This separation is usually achieved with chromatography however it is not wholly necessary and samples can be directly vaporised for analysis. Samples are ionised and fragmented with the molecular masses of each fragment and ion collected and measured. This requires equipment and time for separation and some form of sample collection. With the information provided by mass spectrometry and a relevant library most compounds can be swiftly and accurately identified by an unskilled operator. There are some forms of mass spectrometry with lower sample preparation requirements including inductively coupled plasma mass spectrometry and ambient ionisation mass spectrometry [22-24]. However, the inability to perform standoff detection mass spectrometry limits the usefulness of the method to this project [12, 13].

IMS is a widely used technique for detecting trace amounts of energetic materials, particularly nitrates and is used commonly in airports [14, 25]. This is another ionisation technique and is designed for detection of trace amounts of a specific range of compounds. It makes use of ion time of flight to characterise an unknown sample. The technique has detection limits between 1ng and 1000ng for a selection of energetic compounds [26]. Portable models of the technique are also readily available for use. However, this technique cannot be used at standoff ranges.

Gas sensors have been developed for energetics detection as both passive sensors for monitoring local atmosphere [15] and also as close trace detection [16, 27]. While both forms of the detectors produce fast results and would be able to detect trace amounts of explosives [7] without sample preparation. The passive sensors only determines if there are explosives nearby but not where and the close trace detection cannot be performed at standoff ranges. Both are also somewhat limited in scope of materials they can detect.

Raman Spectroscopy (RS) operates on a similar principle to IR spectroscopy. However, it observes a different form of scattering to measure different vibrational modes. The method provides primarily molecular structural information. RS has no sample preparation requirements and can be performed at standoff ranges. RS has higher detection limits than most of the other techniques mentioned and is less able to perform trace detection [28]. The information that RS can gather from inorganic samples is limited, but it may be a useful tool in conjunction to another technique to detect energetic materials. RS has been used for energetics detection at close [17] and standoff [29] ranges.

LIBS is a laser technique that, making use of a high-powered laser pulse, ablates and ionises a small section of the sample surface into a plasma. An optical spectrometer collects the light emitted by the plasma as it cools. This light contains atomic information for all species present and can be used to discern the atomic composition of the sample. LIBS can be performed both at standoff ranges and with no sample preparation although this does reduce the signal to noise ratio of the resultant data spectrum and poses additional challenges [30]. Samples composed of low atomic-number elements such as those found in organic compounds can be difficult to generate a plasma. However LIBS is extremely effective on samples composed of heavier elements such as mineral samples [31]. Given the prevalence of organic energetics, LIBS would not be an effective technique if used alone and would require a complementary technique for use in this project.

Having considered the above techniques a combined LIBS/RS system was selected for this project. Both techniques are usable at standoff ranges and through some forms of containers as shown in Izake *et al.* [11] where Raman spectroscopy was used at a stand-off range of 15m to determine the content of highly fluorescent packages. González *et al*. [30] performed LIBS at a range of 30m though a transparent barrier. Both techniques can be performed without sample preparation [17, 31] and return results within the desired timeframe of less than 10 seconds. LIBS/RS were individually unable to classify some types of materials reliably, however the types of energetic materials that cannot be easily classified by one technique are relatively easily classified with the other [32]. IR would also be able to produce similarly complementary data, however standoff IR requires a different type of excitation source, where standoff LIBS and RS can both use the same laser. LIBS and RS techniques have gained a recent increase in tandem use for a range of uses including identification of plastics [32], glasses and in one notable case Martian minerals [33] as well as energetic materials [21, 33-35]. None of the methods considered were able to meet all the selection criteria. LIBS/RS were able to meet most of the criteria however the results are difficult to interpret quickly without training. To increase the ease of use, an alternate method to a human trained to read spectral results is thus in need of consideration.

1.4 Laser Induced Breakdown Spectroscopy

Laser induced breakdown spectroscopy (LIBS) is a chemical analysis technique that uses a nanosecond pulsed laser to illuminate and ablate a small section of the analyte which forms a micro plasma [31, 36]. This ablation is caused by a variety of factors including; molecules within the sample absorbing the photons from the laser and gaining energy, but also heat induced ionisation and other processes [37]. When the energy density at the target is high enough, as in the case of LIBS, this can cause the electrons to pass the maximum energy state of the molecule producing a free electron. When a free electron is produced the molecule breaks down into an ionised state and a plasma is produced. In addition to a broadband blackbody spectrum, the plasma releases light as it cools due to electrons releasing energy as they return to lower energy states. The light is collected by a spectrometer from UV through visible wavelengths to IR, to produce a light emission spectrum of the plasma [38]. The light an electron produces as it relaxes into lower and ground energy states is characteristic of the element it is orbiting. Thus, the spectrum generated by LIBS can be used to determine elemental information and occasionally some molecular fragments.

An example of a LIBS spectrum can be seen below in figure 1.1. This spectrum of stainless steel shows many peaks that are often common to a metallic LIBS spectrum. Each peak in this spectrum comes from a specific elemental source. For instance, the largest peak at 520nm is a chromium (I) peak. An alternative method of writing this peak would be Cr (I), Cr indicating the element and (I) indicating the ionisation state of the element. Other key peaks in this spectrum are iron (I) at 382nm, chromium(I) at 427nm and mixed iron (I) and chromium (I) peaks between 490-496nm [39]. This example shows how LIBS spectra are unique to the elements contained in the target sample. LIBS is most effective on heavier elements, more easily producing a clear spectrum with good signal to noise ratio.

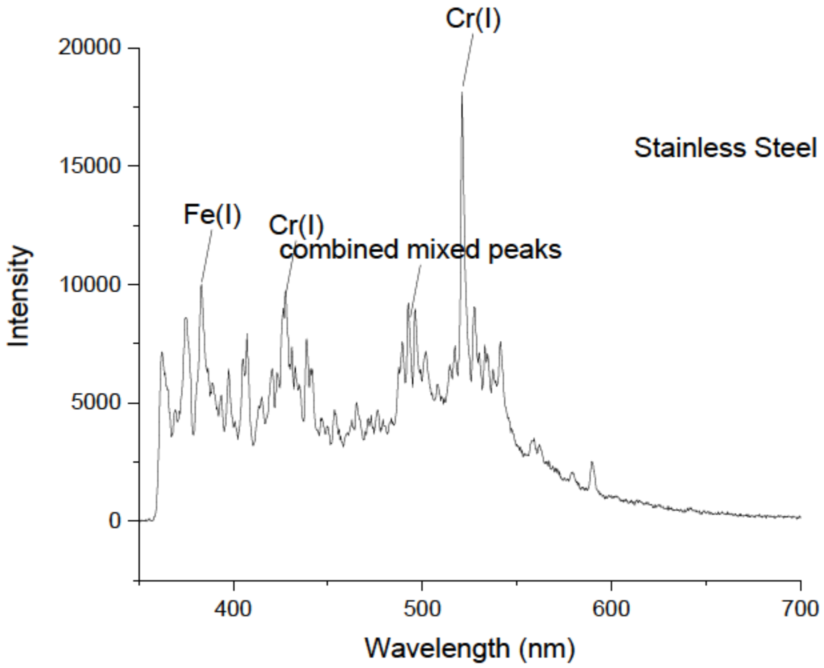


Figure 1.1 An example LIBS spectrum of a stainless-steel surface. This spectrum shows mean peaks from iron and chromium. The data shows the wavelength and intensity of light emitted by the cooling sample.

LIBS is prone to shot-to-shot variation in the resultant spectrum that is obtained each time the laser is fired [30]. Shot-to-shot variation is caused by several factors including

(i) Local elemental inhomogeneity in the sample, only the small area illuminated by the laser is vaporised and sampled thus any local variation will be evident in the spectra;

(ii) the surface conditions of the sample, local roughness altering the ablation plume and how light travels to the detector and possibly favouring light of a given wavelength; and

(iii) the local atmospheric conditions, small changes to these conditions can change the laser focus altering the plasma. In a lab setting shot-to-shot variation is often controlled with the use of an inert gas environment and careful sample preparation but that is not possible in the intended application [40].

LIBS is useful in this research for much the same reasons as RS, being fast, effective at standoff ranges of as much as 30m [30], having no need for sample preparation, or required atmospheric controls. LIBS has been shown to have high sensitivity, having been used to detect trace amounts of energetic material even from confounding backgrounds [18]. However, interpretation of LIBS data does require special consideration when used on organic energetics as the similarities in elemental composition between many common safe and unsafe compounds make their spectral features harder to discern [41]. LIBS is also fast and focused enough that, while it does introduce enough energy to turn a section of the sample into plasma, it does not cause any but the most sensitive, and thus unsafe to use, energetic materials to detonate.

1.5 Raman Spectroscopy

Raman Spectroscopy (RS) is a technique that has been used in fields such as criminalistics, to determine the species of origin for a given blood sample is [42], analytical chemistry to differentiate carbohydrates amount many other compounds [43] and physical chemistry where it can be used to observe the mechanics of an ongoing reaction [44]. RS interrogates vibrational modes in molecules which change polarizability. It does this by collecting light that has inelastically scattered from a target molecule. The light observed post scattering gains or loses a specific amount of energy, in either an anti-stokes or stokes scattering event respectively as shown in figure 1.2. These two types of scattering are both rare however anti-stokes events are even less common and thus generally only stokes events are presented [45]. While RS relies on inelastic scattering elastic scattering is still far more common and thus a filter is required to reduce the amount of unshifted light that is collected. The amount of energy lost or gained is quantised but characteristic of the type of bond it interacted with by way of Hooke's law [46]. The difference in energy between the incident and recorded photons can be plotted into a spectrum [47]. This spectrum can be used to determine information about chemical bonds including which elements are bonded, the strength of the bonds and some structural information. An example spectrum where bonds can be differentiated by strain is shown below in figure 1.2. The example shows that even very similar molecules produce different characteristic Raman spectra.

Removed due to copyright restriction

Figure 1.2 showing (a) the energy change of a Raman scattering, either a gain or loss of energy is induced in a photon by a change of vibrational state in the molecule hit [45]. (b) showing an illustrated Raman spectrum with a notable separation between two bonds of the same type, the only difference being in bond strain [47].

RS is ideal in the current research applications of energetic materials detection due to being complementary and compatible with LIBS. RS provides different information, using the same laser, quickly, is non-destructive and capable of use at standoff ranges [28, 35]. The ability to determine subtle differences in a sample such as differentiating between types of sugars [43] is also vital to the detection of energetic materials which can involve equally subtle variations of organic compounds.

1.6 Combining Complementary Techniques

The complementary information obtained from each of the two techniques of LIBS and RS, when combined together, affords an incredibly powerful data set. The data set could be used to discriminate and identify different classes of chemicals according to their explosive potential [35]. Pairing these techniques has allowed for the discrimination of similar compounds that neither technique could easily determine on its own, such as discerning between different plastics when they have been coloured with dyes or pigments [32]. As an example, RS can determine the type of plastic but is often incapable of determining the colour causing additives, making it ineffective on some black plastics. LIBS on the other hand cannot differentiate types of plastics as easily as RS but as it is not dependant on scattering, LIBS is able to isolate additives within plastics, including black plastics. Together LIBS and RS can provide correlative data on composition and molecular structure for a sample.

There are challenges when trying to use the two techniques in one system. These challenges have already mostly been resolved by prior research in the overall project performed by Queensland University of Technology, Flinders University and Defence Science and Technology Group (DSTG). The primary difficulty in using the two techniques together was that of timing. While both can be collected from the same laser pulse, the timing of data collection to record a clean LIBS and Raman spectrum separately took some experimentation. The solution for the timing issue is fortunately possible as figure 1.3 shows Raman and LIBS emissions occur over different timescales. For a given laser pulse, RS data is generated over nanosecond timescales and can be collected first before ablation occurs and the plasma forms. Thus, RS describes the sample surface in its initial state and is not influenced by the breakdown products and fragments created by ablation. LIBS data can be collected as the ablation plasma cools over several microseconds and thus LIBS provides information on the sample composition in the state it used to be in. With careful detector timing both types of data can be collected without interference. Both techniques use the same kinds of spectrometer laser source and detector configuration [48].

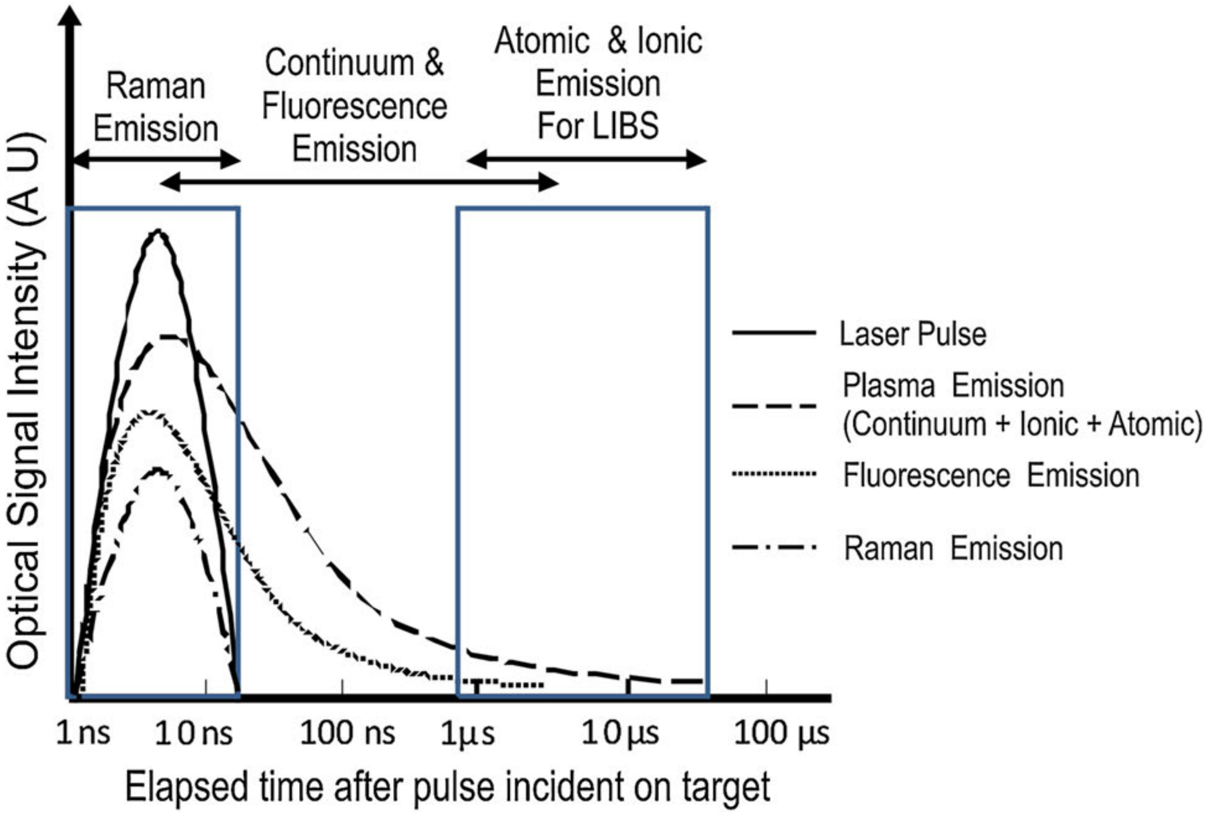


Figure 1.3, showing the timescale LIBS and Raman emissions occur when a laser pulse hits a sample. Fluorescence is included though many samples do not fluoresce strongly enough to be noticeable. Note the logarithmic timescale on the x axis, showing a significant temporal separation between the two different types of emission [48]. Reprinted with permission from Springer Nature.

1.7 The Need for Automated Data Interpretation

As mentioned both systems require some time to understand the complex data output and interpret the data to a sample class, even for a skilled operator [46]. The time and skills to manually read spectra are not ideal for military operations, simpler and less personnel intensive methods are preferred and thus an alternative, autonomous method for data interpretation is required.

An option to reduce the need for a skilled operator would be the implementation of digital hardware that performs a set of algorithms to perform the analysis. This algorithm could then provide the class labels with confidence levels so that decisions can be quickly made by the user. An advantage of such an approach would be that it is able to perform the analysis far faster than a human operator when presented with a complex spectrum.

Chapter 2 - **Algorithms for Autonomous of Data Interpretation**

2.1 Interpreting Complex Chemometric Data

In Chapter 1 the spectroscopic techniques of LIBS and RS were discussed and determined to be the most applicable for this project. However, it was also established that without specialist training the data can be difficult and slow to interpret.

To demonstrate the typical data features that need interpretation in the application of this work, the Raman spectrum of xylitol is presented in figure 2.1. This spectrum has key peaks at 395nm due to a CH_2 stretch and a peak at 402nm due to OH vibration and several other peaks.

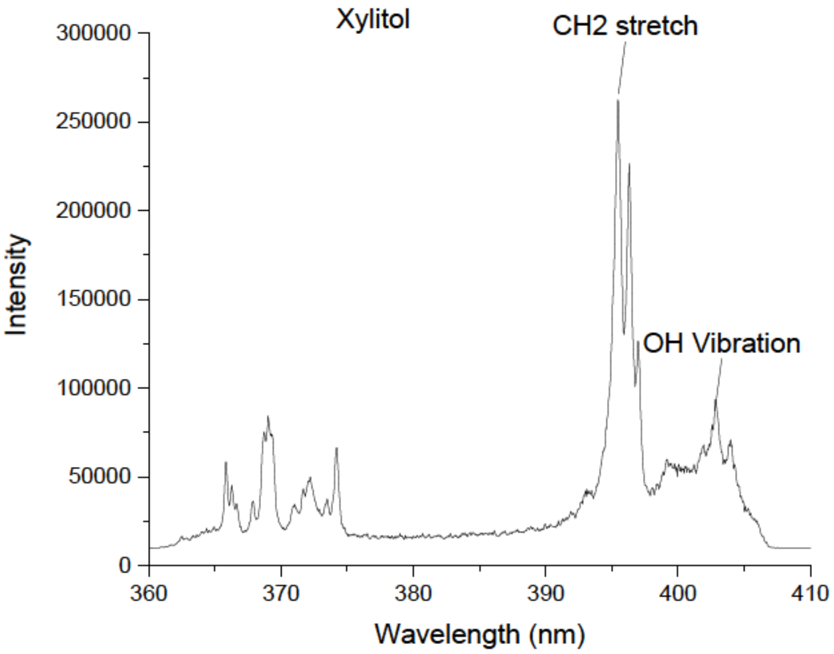


Figure 2.1 Raman spectrum of xylitol.

The xylitol spectrum has more complex features than the most notable peaks labelled above. These peak structures indicate molecular level information about the sample surface. Xylitol is an excellent sample to illustrate the typical features of a Raman spectrum of an energetic material. Xylitol is an organic sugar and precursor to the hazardous material xlylitol pentanitrate.

In contrast the LIBS spectrum often has more features to discern than the Raman spectrum. An example of a LIBS spectrum of a sparkler, a well -known, commonly available pyrotechnic device, is presented in figure 2.2. Some of the key peaks in this sample are Al(I) at 394nm and 396nm Mg(I) at 383nm and Ba(I) at 455nm and 494nm.

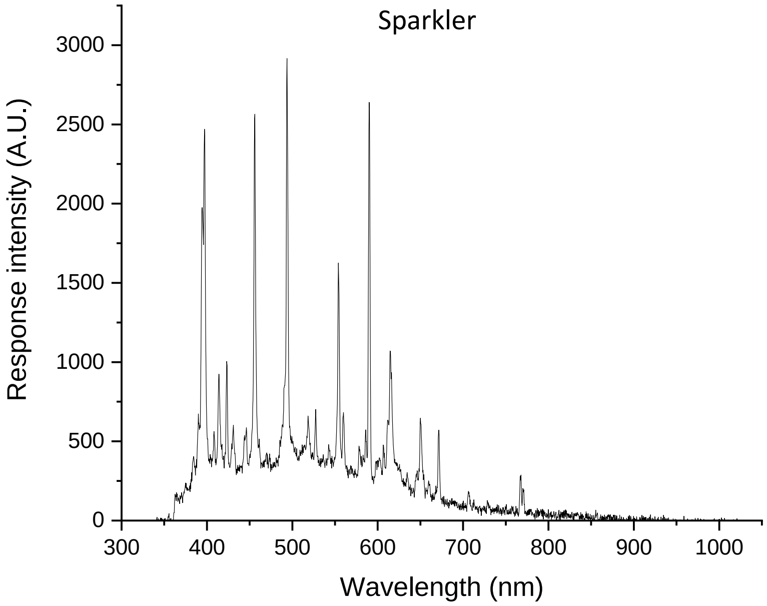


Figure 2.2 LIBS spectrum from a sparkler sample.

Sparklers are a complex mixture of compounds including aluminium and other metal oxides, a nitrate and a binding agent. This mixture is a relatively safe energetic compound but is an example that well illustrates how a complex mixture can present a challenging sample for a data analyst to interpret.

As illustrated with the peak assignments for the spectra in figures 2.1 and 2.2, data types are difficult to interpret without specialist training. Even with proper training and reference spectra the process of determining the identity of an unknown sample by hand is often slow and potentially inexact. This is highly undesirable in a high-stakes qualification of an unknown sample, as extra measures to confirm accuracy are required and are time consuming. The need for expedience with minimal possible human error leads to automation with an algorithmic approach as the best way forward.

To counteract this need for extended time periods and specialist training to operate a LIBS/RS system an alternative method of interpretation is preferable. To that end machine learning algorithms have been reported as suitable approaches for interpretation of complex data sets. Various methods utilising machine learning have been applied on both LIBS [49, 50] and RS [51, 52] data and similar chemometric techniques are beginning to be applied to combined systems [32]. The aim of this research project is to determine the most suitable algorithm to combine data from RS and LIBS to classify an unknown object as being either hazardous and comprised of energetic compounds, or safe.

2.2 Machine Learning – Reducing Human Error

Machine learning (ML) is the use of algorithmic program development to generate software that improves its effectiveness at a selected task without direct user intervention. The algorithms used to hone a program are referred to as machine learning algorithms. These algorithms make use of some system for rating a program performance at its task and a means to change the way the program works.

This project focuses on machine learning algorithms used to produce classification models, programs that take some unknown data and assign a property to it. The classifiers in this instance will be used to generate sample type labels for unlabelled spectra. Generally, programs produced with machine learning are too complex for a human to produce in a reasonable timeframe and include useful factors and patterns in the data humans would be unaware of. The created algorithms are often more accurate than what a human could feasibly create unaided, taking features a human would not consider into account [53, 54]. Another advantage of machine learning algorithms is that a trained ML model can be quickly and easily used to answer questions by a system by moderately skilled operators. This is an ideal property for this project as the intended use for a completed model is for relatively unskilled operators to be able to identify unknown objects.

This project fits into a wider research program, a part of an Australian Defence Grand Challenge program on Counter-Improvised Threats, to create a system for rapid detection of energetic materials deployment by military operators. This system is being designed to quickly identify hazardous objects under far from ideal laboratory conditions. As this is the case, machine learning models will provide the most consistent and fastest option to reliably determine the identity of unknown compounds with maximum confidence. While highly skilled and experienced human operators may be able to outperform the final program, they will be slower and may make errors in judgement, missing portions of the signal in an unpredictable fashion. As the situation of the detector's operation require both speed and accuracy, a computer-based platform is ideal [54]. Such a platform is possible with traditional chemometric analysis but faster and more effective if machine learning is utilised.

Machine learning algorithms use known "training" data to produce a model and use "testing" data to ensure the model is accurate [53-55]. Outliers within the data will often lead to errors in classification. While such errors are unavoidable, they can (when present in the training data) cause what is known as overfitting. If the model is trained too thoroughly, or on training data including outliers, the model can take these outliers into account and the model that is produced becomes less accurate for new data.

An example of overfitting can be seen in figure 2.3, the training data shown includes an outlier in the green data, which it is clearly placed and classified away from the other green data points. The model has been trained to the point of overfitting due to this data point. This is shown by the blue "decision boundary" zones, imaginary lines in the dataspace marking the transition between classification. When the testing (black) data is added, they are incorrectly classified by the model to be within the "green area", these points are in truth orange and the green outlier in the training set has resulted in incorrect classifications and a less accurate model.

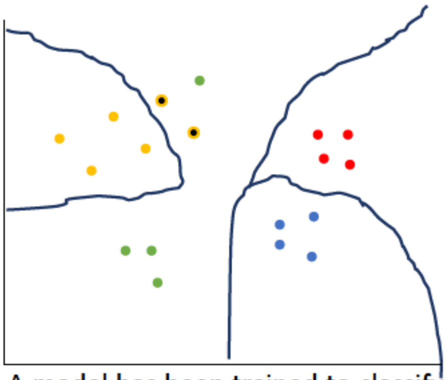


Figure 2.3 showing sample data. A model has been trained to classify each zone to the colour (class) of data in that zone. The black data points added by the testing set, true classification orange, show this to be a clear case of overfitting.

Preventing overfitting varies by the type of machine learning algorithm used to train the model. With some algorithms, simple parameters and removal of outliers from the training data is sufficient to prevent most forms of overfitting. In others, ensuring training is stopped after a "good enough" fit can be necessary to prevent overfitting. This is dependent on the type of machine learning algorithm.

There are two major types of machine learning algorithm for classification. These two types called supervised and unsupervised learning have different strengths and uses [54].

Supervised Machine Learning

Supervised algorithms make use of training data with known classes to train a model to sort data into those same known classes. Supervised algorithms can be used to make models for both classification and regression analysis. This project considers a problem of classification and thus will focus only on classification models, regression models will not be considered further. Classification models created through supervised machine learning are used, as the name suggests, to sort input data into a series of classes [53, 54].

Unsupervised Machine Learning

Unsupervised machine learning algorithms are designed to be used on data that was not classified at collection or subsequently. These algorithms produce models that find patterns in data and cannot generally assign data to classes in the same way as supervised learning. Due to these difficulties in using unsupervised learning to sort data into known classes it will not be used in this project and thus will not be discussed further [53, 54].

2.3 Machine Learning Algorithms

There are a number of different approaches that can be used for machine learning, each with its own strengths and limitations. This project requires models that can quickly and accurately determine the identity of an unknown compound from a large variety of material classes with an emphasis on accuracy. Once trained, most ML models are able to determine the identity of a single sample relatively rapidly. During training processing speed can become an issue with large multidimensional datasets due to the large number of calculations and high degree polynomials calculations required.

The No-Free-Lunch theorem of optimisation in reference to supervised learning as summarised by Wolpert *et al.* [56] is as follows:

For any two learning algorithms, there are as many situations (appropriately weighted) in which one is superior to the other as vice versa, according to any measure of superiority.

It is difficult to prove this property, as shown in a subsequent article by Wolpert *et al.* [57] and testing multiple learning algorithms is preferred. A number of algorithms will be considered for use in this project and the resultant models will be compared in measures of robustness and accuracy.

Decision Tree

A decision tree algorithm produces models that are functionally a branching series of questions, called nodes, with each possible answer leading to either a new question or a final outcome. In this application such an answer would be a classification and some possible questions would be whether a peak at a particular point was present or absent or the ratio between two points in the spectrum. The question asked at each node is determined via an information gain ratio. When deployed in a sorted order from greatest to smallest information gain ratio, the decision tree leading to the most efficient and quickest overall determination is selected.

Models produced with this algorithm have been used in various medical applications [58, 59] and are easy to understand by humans while maintaining accuracy. Decision tree models are not best suited to large multidimensional datasets and those with continuous features, they're also prone to over fitting unless the tree growth is carefully controlled [60]. A simple diagram of a decision tree is shown below in figure 2.4. This model shows initial and branching questions in the Root and Internal nodes and classifications stored in leaf nodes [61].

Removed due to copyright restriction

Figure 2.4 Example diagram of a decision tree structure. Figure source [61].

Support Vector Machines

A Support Vector Machine (SVM) algorithm calculates a mathematical hyperplane (by adding an extra dimension) that produces the most separation between data with different labels. New data introduced is positioned in the feature space and the hyperplane boundaries are used to determine the label of this new data. This classifier has been used for a LIBS identification of plastics [49], and Image analysis [62]. The main advantages of SVM are that it can be accurate with small training sets, can handle data of high dimensionality with minimal slowdown and is mathematically interpretable. However, SVM classifiers, while not suffering too heavily from the slowdown due to high dimensional data, slow down exponentially with the number of entries within the dataset and number of classes [49, 54, 63].

K-Nearest Neighbour

The K-Nearest Neighbour (KNN) algorithm is a mathematically simple algorithm taking a distance function and determining the “closest” labelled data to a new data point. The new data point is then assigned the label of the most common known data label within the nearest “K” data points. These classifiers have been used for such things as predicting the yield of shale oil [63], and ocean carbon content [64]. This algorithm is quick and simple to implement and iterate upon and with a moderate value of K will ignore outliers in the data. However, the choice of distance function is critical to its success. Depending on the distance function it can be simple to intuit or more difficult to understand why any given data point is considered “closer” to another. The biggest flaw of the KNN algorithm is the need to keep the whole dataset in memory. Where datasets are small, this is a nonissue, but with larger higher dimensional datasets, the prediction speed slows down until it is too slow to be pragmatically usable in this application and the large memory requirements can become overwhelming [63].

Artificial Neural Networks

Artificial Neural Networks (ANN) operate via a series of mathematical objects referred to as neurons. Each neuron takes the input data and utilising it in a calculation, produces an output from that neuron. The output can then be passed either onto another layer of neurons or into an output layer which can then identify the input. These classifiers have been used for detecting cancers [52], flight control [65], power flow control [66] and many other applications. ANN classifiers are very accurate and handle high dimensionality well, however, they are extremely mathematically complex being nearly impossible to decipher by hand. They are also they are prone to heavy overfitting[55].

Naïve Bayse

Naïve Baysian methods assign samples to a class based on a probability distribution built from training data. They have been used, for example, to work out landslide probability [67]. Naïve Bayes classifiers assume the predictors are independent in a dataset [68]. Preliminary testing found this method to be comparatively slow to train and less accurate than other methods, thus it was not optimised further.

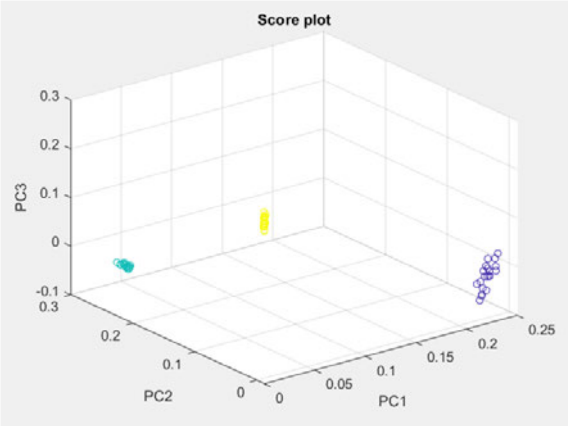
Methods Selected

As this project is designing a system for use with a large number of classes, speed, while not critical, is a discerning factor. SVMs will likely struggle under the full-sized dataset and are not considered viable. Decision trees while fast and accurate are unlikely to be the best fit for spectral data as it is both continuous and has high dimensionality. Thus, this project will examine KNN and ANN classifiers. Due to KNN’s relative simplicity fast prototyping and optimisation can be achieved. There are limitations in the nature of the dataset which can be contained in a KNN model, but, with a sufficiently processed dataset, it may be light and fast enough for the intended application. ANNs were selected for their accuracy and ability to work quickly with data even of high dimensionality, though their propensity to overfitting and low interpretability may be serious drawbacks. KNN and ANN will be used alongside a statistical chemometric approach utilising Principal Component Analysis and Linear Discriminant Analysis.

2.4 Principal Component Analysis

Principal Component Analysis (PCA) is a multivariate statistical technique, that by re-expressing the data in terms of a series of orthogonal basis sets allows for the key variations in large datasets to be easily visualised. PCA allows data to be compressed and removes potential data misinterpretation through de-emphasis of large amounts of irrelevant information. Each basis set generated in a PCA is used to describe the variation in the data. The amount of variation each basis set explains can be expressed in a scree plot, shown in figure 2.6. These basis sets are referred to as Principal Components (PC) and are numerically ranked by the amount of variance each component describes [54]. Data can be excluded with the use of PCA by simply using the PCs that explain the most variance [2]. Low numbered PCs are used to create a coarse categorisation with the high number PCs providing refinement, for example, in this project, the exact nitrate from a set of similar nitrates.

Another useful function of PCA is the identification of what data is useful and what is irrelevant. Influence plots can be generated showing how the identified variable affected the PCs. These plots can be used to determine if data is useful or not. Data proposed not to be useful is removed and the plots are compared before and after removal [36]. If no change is caused to the influence plot by the removal of data, then it can be confirmed that the data contributed little useful information [32]. Additionally scores plots can be used to help identify outliers highlighting when a sample is significantly different to others of its class.



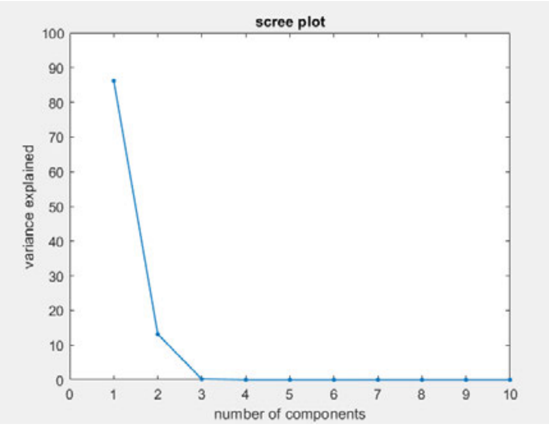


Figure 2.5 (left) a PCA scores plot showing the relationship between three classes of sample in the first three PCs. Figure 2.6 (right) a scree plot showing the amount of data variance explained by each PC.

2.5 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a form of simple supervised machine learning that builds a model for classifying new data into supplied classes. This classification relies on Bayes formula of contingent probabilities. Based on the assumption that all probability distributions are known and all possibilities are represented in the data it is possible to use the data to produce the probability of new data belonging to each of the classes [69]. To perform this task a Linear Discriminant Function (LDF) is performed. Two major types of LDFs are the class-dependant and class-independent functions. The two different functions emphasize different types of variance. Class-dependant functions better emphasise the variance between different classes allowing for clearer separation but require known classes. A class-independent LDF generalise better than class-dependant functions and show the variance between all data points more strongly. The choice of which function type to use is data dependant and in this application a class-dependant function seems initially to be more appropriate [70]. An LDF is used to produce an eigenvector or a series of eigenvectors as seen in figure 2.7 below. New data is transformed by the same LDF and a simple distance measure is used to determine which class is the most appropriate. This algorithm has no in-built way to determine if data does not belong to a known class, although a probability threshold can be used to ensure that the data is not assigned to a class in such instances [71].

Removed due to copyright restriction

Figure 2.7 (left) and 2.8 (right), showing the transformed and original data for two different LDFs, one using the class-dependant method the other using the class-independent method. Note the eigen vectors (points in pink and black) are aligned in the class-independent method but are not in the class-dependant method [70].

2.6 K-Nearest Neighbour

The KNN algorithm requires clustered data to be accurate. It stores a large library of known data points and notes the classifications of the K nearest points. Figure 2.9 below shows both a 3-nearest neighbour and a 5-nearest neighbour example. From these nearest points it assigns the most common classification to the new input. Nearest points are calculated by generating an n-dimensional hypersphere, n being the dimensionality of the data, with a radius set to include only the desired number of points [72]. This is a very simple algorithm that defers most calculation until a new input is introduced.

Two common ways to tune the accuracy of the algorithm are setting value of K and weighting the checked neighbours based on their distance from the new point. Distance in this context however is not a single concept with a single method of calculation. Several different forms of distance, distance metrics, are used for KNN studies. Here five different metrics were used, Euclidean, City block, Minkowski, Spearman, and cosine [73].

The most commonly referred to distance measure when "distance" is considered is the Euclidean distance metric,

$$d(q,p) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \qquad \text{EQ 2.1}$$

where q and p are two points with d(q,p) being the distance between them, q_I and p_I are elements of q and p in a given dimension i. The Euclidean measure is the most commonly thought of distance being effectively a straight line between two points in two-dimensional space. Two of the other distance metrics to be used are similar to Euclidean; those of City block and Minkowski. City block is defined in EQ 2.2 as

$$d(q,p) = \sum_{i=1}^{n}(q_i - p_i) \qquad \text{EQ 2.2}$$

$$d(q,p) = \sqrt[n]{\sum_{i=1}^{n}(q_i - p_i)^n} \qquad \text{EQ 2.3}$$

The Minkowski metric, EQ 2.3, to be used in this analysis is the cubic metric, ie n=3. These metrics were chosen to determine the effectiveness of a variety of more direct distance measures to contrast the other major type of metric.

The other metrics considered are correlation metrics, these are designed not to look at the direct magnitude of the distance between two points but rather determine a method to compare the shape of a pair of vectors. The two metrics employed to this end herein are the cosine distance metric and the Spearman dissimilarity metric. The cosine metric, defined as

$$sim(x,y) = \frac{x.y}{\|x\|\|y\|} \qquad \text{EQ 2.4}$$

where x and y are two vectors and ||a|| is the Euclidean norm $\sqrt{\sum_{i=1}^{n} a_i^2}$ of some vector "a". This metric calculates the cosine of the angle between two vectors, the lower the angle the greater the similarity between the shapes of the vectors. The value of this metric ranges between 0 and 1, a cosine correlation of 0 indicates the angle between the two vectors is 90 degrees and they share no similarities, while a cosine correlation of 1 indicates the angle is 0 and the vectors both have the same shape. The other correlation metric to be used is the Spearman's rank order correlation metric, this metric ranks all data for each variable from highest to lowest and uses those ranks to calculate how similar the shapes of the vectors are. The Spearman metric is here defined as

$$d(q,p) = \frac{(r_q - \bar{r}_q)(r_p - \bar{r}_p)}{\sqrt{(r_q - \bar{r}_q)^2}\sqrt{(r_p - \bar{r}_p)^2}} \qquad \text{EQ 2.5}$$

where r_q is the ranks of point q and $\bar{r}_q$ denotes the average rank of point q, as this is the average of the ranks $\bar{r}_q = \frac{n+1}{2}$ where n is the number of variables in vector q. This metric results in a value between 1 and -1, 1 showing complete correlation and -1 showing complete negative correlation. These correlation metrics are useful particularly for LIBS data as the magnitude is variable due to shot-to-shot variation being common. These metrics are resistant changes in intensity due to their focus on the shape of a vector rather than the size.

To weight the classification based on proximity to the new data point each data point's "vote" is numerically expressed and divided by the vector magnitude to the relevant point to give a vote strength. The exact degree of biasing can be controlled with cut off values, both maximums and minimums, for the vote strength, this allows a distance in which all points are treated equally and a point after which points will not be considered. Using the example of the right point in figure 2.9 below if a strong distance bias is used the vote strengths of each point could be; blue1 (B1) 1, R1 0.8, B2 0.1, R2 0.1, R3 0.1, thus the vote sums would be blue 1.1, red 1 and would thus class the point as blue. A weaker distance bias could produce vote strengths of B1 1, R1 0.9, B2 0.2, R2 0.2, R3 0.2, and thus have a vote sum of blue 1.2 red 1.3 and instead be classed as red.

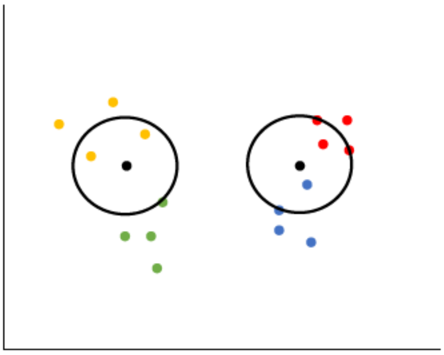


Figure 2.9 showing a pair of points being classified by a 3NN and 5NN 2 dimensional KNN model. The left point has two yellows as the closest of its three nearest points and thus would be classed as yellow while the right point has a blue as its closest point a red further away then two equidistant reds and a blue.

2.7 Artificial Neural Networks

An Artificial Neural Network (ANN) is an algorithmic system built of a network of interconnected nodes. When a node accepts an input, the value is multiplied by a set "weight" value. The weight given to an input is set on the connections between nodes and is constant. The weighted values are then compared to a bias function set by the individual node and the output of the node is determined by the transfer function. Most common neural networks are built such that all connections proceed in one direction and are called feedforward ANNs [55, 66].

2.7.1 Artificial Neural Network Structure

The simplest node type used in a neural network is the perceptron, as shown in figure 2.10 below. The perceptron is a node that can accept any number of inputs, apply weights to each input, sums the inputs and determines if the summed weighted inputs are less than or equal to or greater than the bias value. The perceptron will then fire either a 1 or a 0 depending on if the bias value was exceeded by the weighted sum of all inputs IE if $(\sum_j w_j x_j) - b > 0$ then a 1 would be the output [55]. These simple perceptron networks are of limited use as any change in the network substantial enough to alter the result of a perceptron can have large repercussions on the rest of the network making them exceedingly difficult to train. The solution to this problem with perceptron lies in the transfer function, that is the element of the node that determines what is sent when the node fires. Another, more useful, transfer function is the sigmoid function. Instead of being a simple step function like the perceptron the sigmoid function allows the output to be any number between 0 and 1. Mathematically this is achieved with the equation

$$O = \frac{1}{1 + e^{(-(\Sigma_j wjxj)-b)}} \qquad \text{EQ 2.6}$$

where O is the output w_j and x_j are the weights and inputs of a given connected neuron j and b is the bias value of the neuron. With this function a neuron receiving a sum of weighted values much higher than its bias will produce an output close to 1 and one with low inputs would produce a value close to 0 [55]. This transfer function allows for much smaller changes to alter the output of an ANN. Another potentially useful transfer function is the radial bias function. These can make use of several functions such as the multiquadric function or the Gaussian function. The Gaussian function is the most commonly used:

$$\phi(r) = e^{\frac{-r^2}{\sigma^2}} \qquad \text{EQ 2.7}$$

where ϕ(r) the output is proportional to σ a real variable, set by the user and r is the Euclidean distance between the input data d and a centre point u such that r=||d-u||. This produces a function in which when the value of d is much larger than u, a high negative output is produced and when d is much lower than u, a similar output is produced. This causes both high and low input values to produce no response from the neuron [65, 66]. These transfer functions make it simpler to express the certainty of a sample belonging to a class as a percentage, but the nature of the architecture of the network may make this impossible.

Removed due to copyright restriction

Figure 2.10 a basic perceptron ANN node (left), the x values are inputs, the circle area represents where the mathematical operations are performed and finally the output shows that only one output is produced even if this single output is sent multiple other nodes. And 2.11 (right) an example ANN architecture showing an input layer, two hidden layers and an output layer [55].

Any neural network will have at least three "layers"; an input layer, a number of "hidden layers" and an output layer as shown in figure 2.11. It is possible to have only a single hidden layer, a "shallow" ANN, however more complex networks utilise more hidden layers, "deep" ANNs. Deep ANNs are growing more common as training algorithms become more adept at training them. It is more difficult to train a deep ANN but they're able to more thoroughly break down a categorisation into simple yes or no questions answerable from the input and can lead to improved accuracy [55].

2.7.2 Stochastic Gradient Descent

In the creation of an ANN a training algorithm is used repeatedly to set weights and biases such that accurate assessment of input data into classes is made. The most common of these algorithms, the so-called gradient methods create a cost function, as do many others, defined as

$$C(w,b) = \frac{1}{2n}\sum_x ||y(x) - a||^2 \qquad \text{EQ 2.8}$$

where w is the collection of all the weights in the network, b is the collection of all biases in the network n is the total number of training inputs “a” is the output vector from the network for the given input x [55]. These gradient algorithms then attempt to minimise this cost function as it is far easier to measure changes to the cost function due to a change in weights and biases as opposed to measuring the effects of the changes on the outputs. To minimise the value of C a function is developed such that

$$\Delta C \approx -\eta||\nabla C^2|| \qquad \text{EQ2.9}$$

where ∇C is the gradient of the function C and η is the learning rate, a constant chosen in order to determine how large the individual changes in C are in each application of the gradient. A large η can lead to errors in finding a minimum, while a small η leads to extremely slow training [55].

These functions lead us to an update rule for changing the weights of the connections between each node.

$$\theta = \theta - \eta \cdot \nabla\theta J \qquad \text{EQ2.10}$$

In which θ represents the position of the optimiser on the cost function, $\Delta_\theta J$ is the gradient of the cost function at the current point θ and η is the learning rate. The learning rate generally set to between 0.09 and 0.001. This variance in learning rate is used to control how quickly a neural network learns and different data sets have different requirements. The learning rate is normally so small is in order to keep changes to the neural network gradual and avoid moving over a minimum. An optimiser with a learning rate either too small or too large generally will not converge in a reasonable timeframe.

Performing the function in equation 2.10 over an entire dataset at once is generally a slow method to optimise a network. As many entries in a data set will be extremely similar, determining the gradient over the entire dataset performs almost the same calculation multiple times each step which does not improve performance. To reduce this inefficiency networks are trained on batches of samples, randomly selected from the overall dataset, splitting training into two measures, iterations, each time a single batch is processed, and epochs, each time the entire dataset is processed. No single sample will appear twice in an epoch, each iteration using a different “minibatch” group of samples [55]. The size of a minibatch is set by the user based on memory and dataset considerations, it can be as low as one or into the thousands depending on the amount of data available but is generally kept between 50 and 256. The simplest ANN optimisation algorithm making use of equations 2.8 2.9 and 2.10 is the Stochastic Gradient Descent (SGD) algorithm. This algorithm has some limitations; moving a set distance each iteration and being unable to pass through a local minimum in the cost space. SDG’s inability to escape local minima in the cost space mean that it can reach a point in training where the accuracy is not at its maximum possible value but it cannot be further improved.

ANN optimisation is a random process and thus the exact number of epochs required to produce a viable categoriser is only approximately knowable. It is therefore very difficult to differentiate between the end of training and the beginning of overfitting. This difficultly makes ANNs particularly prone to overfitting. As the weights and biases are initialised randomly and then changed by the learning algorithm, it can be extremely difficult, if not impossible to determine the method by which an ANN produces decision boundaries, though they can be visualised as in figure 2.12 below. This difficulty can lead to an ANN model working in ways that are not expected, such as making classifications by background as opposed to the intended data and other possibly worse outcomes reducing the accuracy of the ANN [72, 74].

This project will analyse the effectiveness of three different optimisers, Stochastic Gradient Descent with Momentum (SGDM), Root Mean Squared Propagation (RMSProp) and Adaptive Moment Estimation (*Adam)*. Each of the three analysed are gradient decent based classifier optimisers. These are assessed in part due to their applicability to deep networks and their robust nature.

2.7.3 Stochastic Gradient Descent with Momentum

Stochastic Gradient Descent with Momentum (SGDM) functions much like SGD with one key difference, while SGM uses a fixed step and only considers the current state of the cost function SGDM also considers the prior steps made in the optimiser. Mathematically this is defined as

$$vt = \gamma vt - 1 + \eta \Delta_{\theta} J \qquad \theta = \theta - vt \qquad \text{EQ2.11[74]}$$

Compared to equation 2.10 the above is very similar the v_t terms and γ being the only additions. The v_{t-1} term is the v_t term from the prior step while the γ term is a factor this is multiplied by to scale the amount of momentum to be simulated. The γ term is usually set below one, with a standard value of 0.9 as this leads to the momentum from prior steps decaying and optimisation coming to an end when a minimum is reached but allows for the optimiser to pass through local minima. A γ value over one would lead to each step being further than the last and optimisation being impossible. This optimiser tends to overshoot the minimum which, while a feature of its ability to avoid local minima, does tend to make this optimiser slightly slower to find the global minimum of the cost space.

2.7.4 AdaGrad

AdaGrad is a more advanced gradient descent optimisation algorithm. The design goal of this algorithm was to create an algorithm that could adapt the learning rate of each feature based on how often it occurred. The general update rule for this algorithm is seen below in equation 2.12.

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \varepsilon}} \Delta_{\theta} J(\theta_{t,i}) \qquad \text{EQ 2.12[74]}$$

In this algorithm's update rule the ε term is a small constant value, normally set to 10^{-8}, intended to prevent division by zero in the rare event where $G_{t,ii} \approx 0$ while having a minimal effect on the results in other circumstances. The $G_{t,ii}$ is a diagonal matrix where each element i, i is the sum of squares of all the gradients of feature θ_i up to time step t [74].

$$G_{t=} \begin{matrix} g_{t,[1]} g_{t.[1]}^{\top} & 0 & \cdots & 0 \\ 0 & g_{t,[2]} g_{t.[2]}^{\top} & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ \lfloor \; 0 & \dots & 0 & g_{t,[k]} g_{t.[k]}^{\top} \rfloor \end{matrix} \qquad \text{EQ2.13[75]}$$

Where g_t,[x] is the gradient at time(t)=x and k is the most recent completed gradient. The gradients are squared utilising a transpose operator.

This change to the learning rate in AdaGrad means that in most instances there is no need to tune the learning rate of AdaGrad allowing the use of the default learning rate of 0.01. The G_t term alters the learning rate in such a way that features less often seen have a larger impact on parameter updates and common ones are less impactful. However, at large values of t, AdaGrad's learning rate tends to reduce to zero due to the matrix G_t becoming too large. This quirk means there is a limit on how many updates an AdaGrad algorithm can perform. For this reason, though AdaGrad is faster and can avoid saddle points that would trap SGDM, AdaGrad itself will not be used in this project, however derivatives of the algorithm will be used that do not have this quirk of the learning rate reaching zero.

2.7.5 Root Mean Squared Propagation

The first of these algorithms to be used in the project is Root Mean Squared Propagation (RMSProp). This algorithm is designed to take a sample of prior gradients rather than every gradient. To this end it makes use of a root mean square term to reduce the accumulated gradients each step. The equations for this algorithm are shown below in equation 2.13 and 2.14 below.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]t+\epsilon}} gt \qquad \text{EQ 2.14}$$

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2 \qquad \text{EQ 2.15[66, 74]}$$

The root mean square term works much like the momentum term in SGDM. The γ term in EQ 2.15 functioning in a similar fashion determining how strongly the prior gradients are considered. The value recommended for γ with this algorithm is 0.9 and a learning rate (η) of 0.001. This allows for a version of AdaGrad, including its increased speed and advantage with sparse features, that doesn't have the flaw of a learning rate that decreases to zero after many updates.

2.7.6 *Adam*

Adaptive Moment Estimation (*Adam*) combines the approaches of both SDGM and RMSProp to create an algorithm that has an adaptive learning rate and a form of momentum.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \qquad \text{EQ 2.16}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \qquad \text{EQ 2.17[74, 76]}$$

Equations 2.16 And 2.17 are estimates of the mean and uncentered variance of the gradients, also known as the first and second moments. These estimations are biased towards zero as they are initialised as zero vectors. This problem is exacerbated both during early time steps and when the decay rates β_1 and β_2 are near 1.

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \qquad \text{EQ 2.18}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \qquad \text{EQ 2.19}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t+\epsilon}} \hat{m}_t \qquad \text{EQ 2.20[74, 76]}$$

To correct the zero-bias seen in the initial mean and variance estimations equations 2.18 and 2.19 were developed. These bias corrected equations are used in the update rule for *Adam* seen in equation 2.20. The recommended values for β_1 and β_2 are 0.9 and 0.999 respectively.

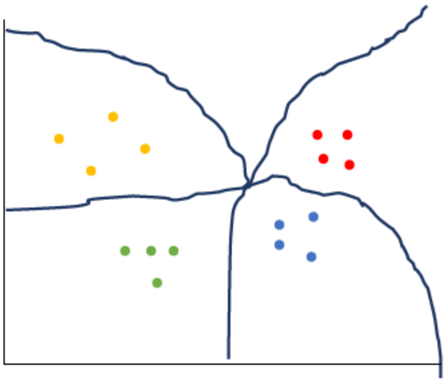


Figure 2.12, hypothetical decision boundaries for a given dataset that a neural network may produce. It is generally not simple to determine how such boundaries are reached and thus careful consideration is required for the use of an ANN.

2.8 Project Aim

After establishing the need for rapid detection of energetic materials and the theoretical background of Machine Learning for automated analysis and classification of the test or unknown sample data against a library or learned dataset, the necessary background for describing the aim of the project upon which this thesis is written has been established.

As noted earlier in sections 1.6 and 2.2 this project falls under a wider program of rapid detection of energetic materials for counter-improvised threats and forms a key role in determining the analysis and classification phase of the data collected through techniques developed in other areas of the project, namely LIBS and RS. The aim of this project and thesis is therefore summarised as follows

- If Machine Learning methods are to be deployed for rapid analysis and chemometric classification of data that are obtained through these techniques, what is the best set of ML-based data processing protocols for identifying whether an **unknown 'sample'** is an energetic material?

Chapter 3 - **Experimental and Computational Methods**

3.1 Data Collection

3.1.1 Instrumentation

To train LIBS/RS machine learning algorithms both LIBS and RS data were collected to use as training and testing data. This data was collected using a purpose-built LIBS/RS system designed to allow for laboratory standoff of up to 5m detection (that is the distance between the target and samples is up to 5m), as well as short range tests as close as 20cm. The data collection system is described in the schematic shown in figure 3.1.

Removed due to copyright restriction

Figure 3.1 The basic set up for combined LIBS-RS detection. Different datasets make use of varied components in this basic framework. Components changed include the laser, optics and the spectrometer. Image from [77].

Two laser systems were utilised in this work, either a Continuum Surelite III NdYAG Laser (operation conditions: 355 nm (UV), 5 ns pulse width, 10 Hz repetition rate, 0.9-4.2mJ/pulse) or a Continuum Minilite NdYAG Laser (operating conditions: 1064 nm (IR), 5 ns pulse width, 10 Hz repetition rate, 8.5-41.5mJ/pulse).

Different optical systems were used for standoff and close proximity. An adjustable four-lens system was used for stand-off conditions at 5 m target distances to focus light onto the target. A single lens (50.8mm diameter, focal length 200 mm) was used for close proximity focusing. For close proximity measurements, the light from the target sample was collected using a dual lens system (50.8mm diameter, focal lengths 200 mm, 100 mm) placed 0.2 m from the target. For stand-off measurements, light from the target was collected via a telescope (Celestron Advanced VX Schmidt-Cassegrain, 203.2mm diameter) and lens (50.8mm diameter, focal length 150 mm). In both collection set-ups, the light then passed through an edge filter (Semrock, 355 nm long-pass, BLP01-

355R-25) and into the optic fibre input of the spectrometer. For 1064 nm LIBS experiments, a short pass filter (Newport) was used.

The LIBS and RS signals were collected using one of, a Flame miniature spectrometer with a non-gated CCD detector (Ocean Optics Flame (Flame-S-UV-IR-ES), 200 μm slit, 2048 pixels) or a Czerny-Turner spectrometer (Andor Shamrock 303i, 1200 lines/mm grating, 100 μm slit) with gated intensified CCD detector (Andor iStar, DH340T-18U-E3). Data from these detectors was sent then to either the Flame OceanView program for the Flame Optics detector or Andor Solis for the Andor detector. Data from OceanView was averaged in program and exported as a series of .csv files. Data from Andor Solis was produced as .sif files and converted to .csv files.

3.1.2 Sample Preparation

Sample preparation and data collection was undertaken by Dr Ula Alexander, work on collected data was performed by Nathan Garner. Crystalline materials such as ammonium nitrate (AN), potassium chlorate ($KClO_3$), potassium perchlorate ($KClO_4$), calcium ammonium nitrate (CAN), hexamine and xylitol were pressed with a hydraulic press into 10 mm diameter discs at 300 bar. Discs comprised of mixed samples were also made to test the ability to collect a combined LIBS/RS signal from one target (e.g. ammonium nitrate with powdered aluminium) and multiple Raman spectra from one target (e.g. AN, $KClO_3$, $KClO_4$ and xylitol mixtures). Mineral samples (barite, bauxite, magnetite, pyrite, calcite and sulfur) were used in their naturally occurring form and were not pressed.

Pressed discs were mounted perpendicular to the laser pulses at the focal point of the beam, and unless otherwise specified, rotated to expose a new surface for each laser pulse. Mineral samples were placed at the focal point of the beam.

Samples were selected as examples of likely targets and sourced from several different organisations. Mineral samples from the rock collection were from a geology teaching set. These samples were used as they are representative examples of the most common chemistry of that mineral. The metals from the Flinders university workshop were used as examples of commercially available metals and representative of "normal" metals found in the field. Samples from DSTG were synthesised by DSTG and mostly consist of energetic compounds and close precursors. Chemicals purchased from sigma Aldrich and other chemical companies were commercially pure chemical samples. The sparkler sample was purchased commercially as an energetic blend analogue. Samples were taken from a single batch, but for pressed organic samples several discs were produced as laser ablation wore through these soft materials.

Table 3.1 Summary of collection conditions for data for machine learning. Courtesy of Dr. Ula Alexander.

Sample	Data Type	Laser λ (nm)	Data collected	Detector	Optics set-up	Accumulation time (gate width) and delay relative to laser pulse	Pulse energy (mJ/pulse)
Aluminium (sheet), Brass, Copper, Iron, Marble ($CaCO_3$), Plastic tray (polystyrene),	LIBS	355	100 sequential single shot spectra	Flame miniature spectrometer	Close proximity (0.2m)	1 ms with pre-triggered detector (100 spectra @ 10Hz = 10s total acquisition	8

Sparkler, Stainless steel (SS)						time)	
AN (and KClO4)	LIBS Raman	355	>20 single sequential spectra at 11 focusing lens positions	Flame miniature spectrometer	Close proximity (0.2m)	1 ms with pre-triggered detector	8
Aluminium, KClO$_3$, Iron, AN, calcium carbonate (CaCO$_3$, shell), AN and aluminium powder mixture, nylon	LIBS Raman	355	20 sequential spectra with 100 shot average per spectrum	Andor iStar iCCD with Shamrock 303i spectrometer, 600 l/mm, 100 μm	Stand-off (5m)	Raman: 30 ns gate width, 0 ns delay LIBS: 1 μs gate width, 150 ns gate delay	2.6 (eye-safe conditions)
AN	Raman	355	15 sequential single shot spectra	Flame miniature spectrometer	Close proximity (0.2m)	1 ms with pre-triggered detector	5
AN, CAN, KClO$_4$, KClO$_3$, AN_KClO$_3$, AN_KClO$_4$, AN_KClO$_4$_Xylitol, ANFO, Hexamine, Xylitol, Urea, RDX in ore (background material), plastic holder, nylon	Raman	355	20 sequential spectra with 20 shot average per spectrum	Andor iStar iCCD with Shamrock 303i spectrometer (1200 l/mm and 600 l/mm)	Close proximity (0.2m)	Raman: 30 ns gate width, 0 ns delay LIBS: 1 μs gate width, 150 ns gate delay	5
Minerals – Bauxite, Barite, Magnetite, Pyrite, Calcite, Sulfur	LIBS	1064	100 sequential single shot spectra	Flame miniature spectrometer	Close proximity (0.2m)	1ms accumulation with pre-triggered detector	41.5

3.2 Datasets

3.2.1 Dataset 1

Three main datasets were used in these experiments. Dataset 1 was the main LIBS testing set, which is larger and more widely varied than the other datasets. It contains 1,456 LIBS spectra from 16 sample types. The distribution of spectra from the different samples is shown below in figure 3.2.

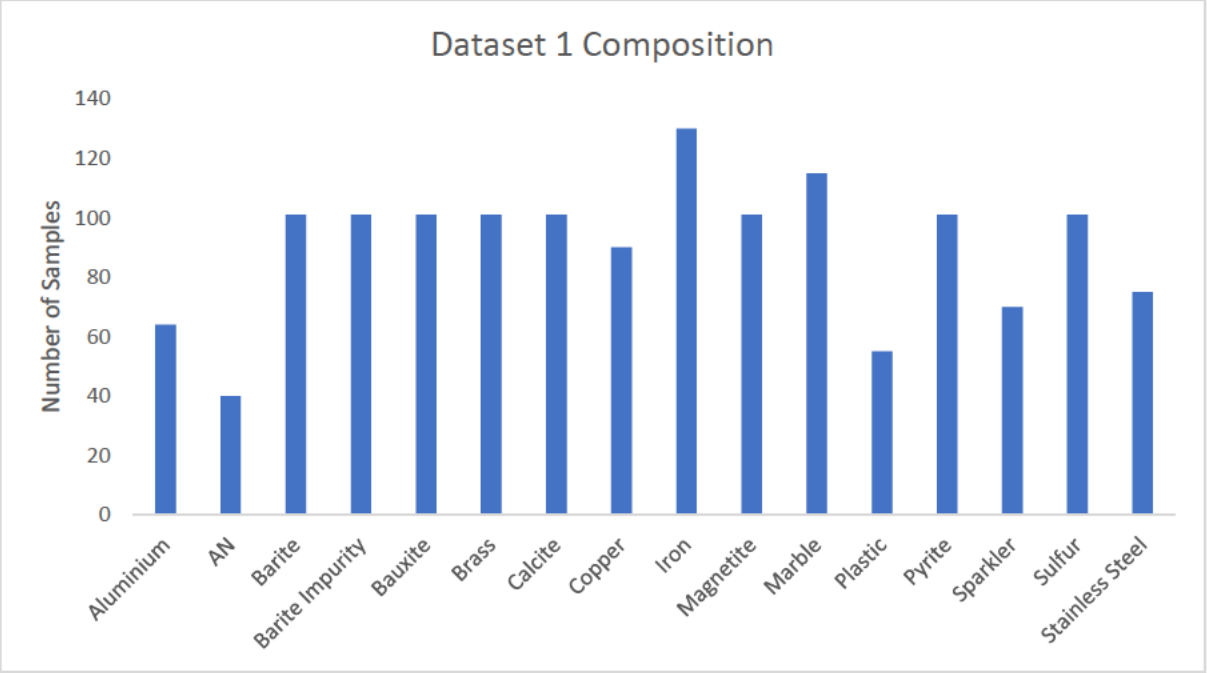


Figure 3.2, showing the number of samples of each class in the dataset.

All spectra were collected with the Flame spectrometer as shown in table 3.1. The molecular formulas (when known) are shown in table 3.2 and an example spectrum for each sample are shown in figure 3.3.

Table 3.2 list of samples and chemical formulas in this dataset [78].

Sample	Proposed molecular formula	Source
Aluminium (Al)	Al	Flinders university workshop
Ammonium Nitrate (AN)	NH_4NO_3	DSTG
Barite	$BaSO_4$	Flinders University rock collection
Barite impurities	$BaSO_4$ (observed pb)	Flinders University rock collection
Bauxite	$Al_2H_2O_4$	Flinders University rock collection
Brass	CuZn	Flinders university workshop
Calcite	$CaCO_3$	Flinders University rock collection
Copper	Cu	Flinders university workshop
Iron	Fe	Flinders university workshop
Magnetite	Fe_3SO_4	Flinders University rock collection
Marble	$CaCO_3$	Flinders University rock collection
Plastic	$(C_2H_4)_n$	Flinders university workshop
Pyrite	FeS_2	Flinders University rock collection
Sparkler	Variable, includes Al, nitrate	Korbond
Sulfur	S	Flinders University rock collection
Stainless steel	Variable, includes Fe Cr and others	Flinders university workshop

Dataset 1 was used primarily as a preliminary test of the classification method. This was the largest and broadest dataset designed to generate the most confusion for the classifier. This dataset included samples such as metals, minerals and a small number of organic and energetic samples. These organic and energetic samples were collected with a different laser to produce a clearer spectrum. These spectra are considered only as a test of the classifier and thus clearer similar spectra are more important than directly comparable spectra.

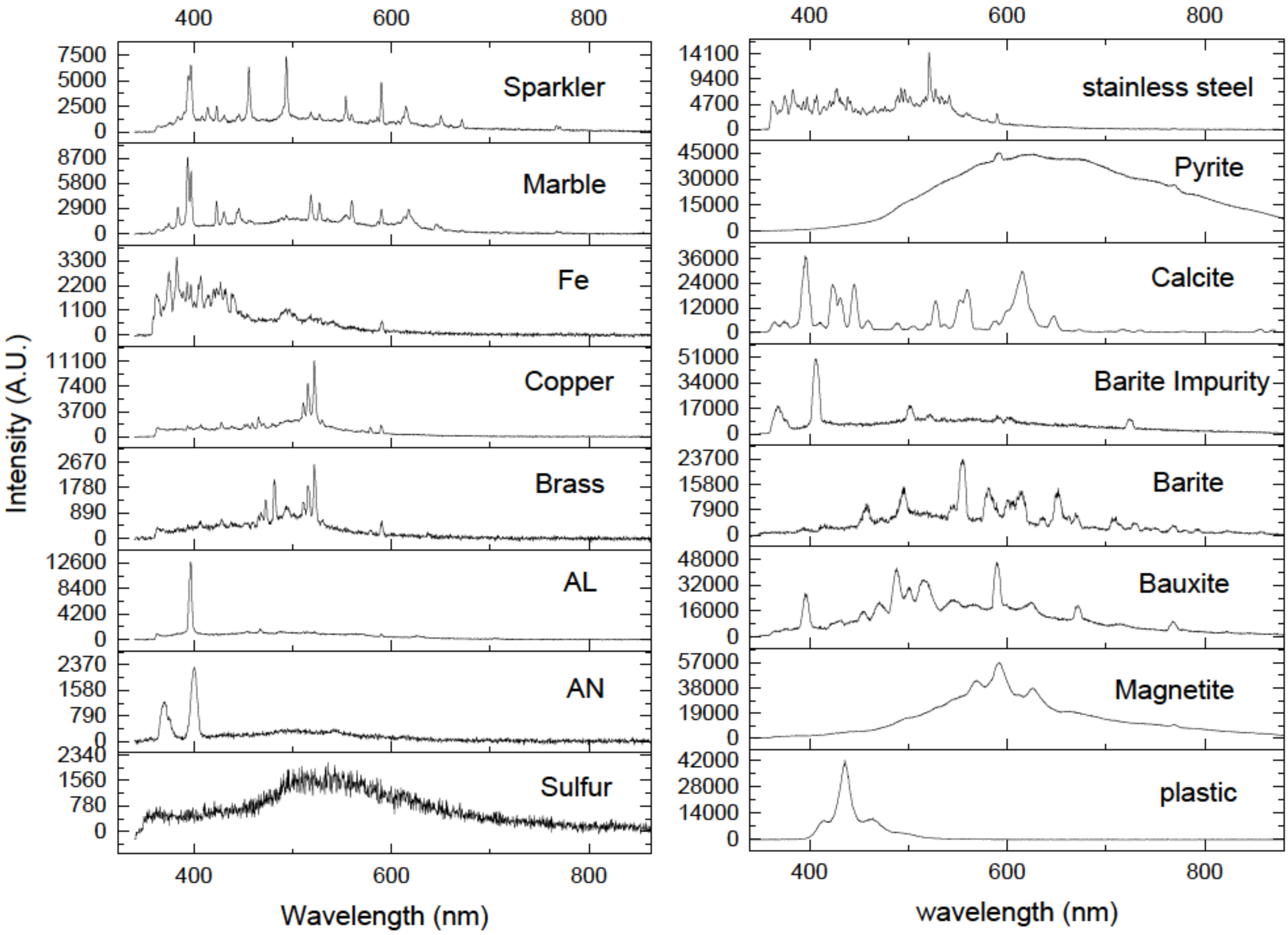


Figure 3.3 example spectrum from each sample type in Dataset 1.

As can be seen in Figure 3.3 the spectra from each of the sample types have a variety of features that should be considered when trying to create a classifier from this dataset. Particularly notable are plastic, calcite, sulfur, and pyrite as these samples had either inconsistent or otherwise notable features.

Table 3.3 Key peak assignments of Dataset 1.

Sample	Peak 1	Peak 2	Peak 3
Aluminium (Al)	396nm Al(I)		
Ammonium Nitrate (AN)	AN raman		
Barite	553nm Ba(I)	457nm Ba(II)	580nm Ba(I)
Barite impurities	405nm Pb(I)	365nm Pb(I)	
Bauxite	588nm Na(I)	486 unassigned	395Al(I)
Brass	522nm Cu(I)	481 unassigned	494nm Zn(II) Cu(II)
Calcite	395nm Ca(II)	616nm Ca(I)	423nm Ca(I)
Copper	522nm Cu(I)	515nm Cu(I)	510nm Cu(I)
Iron	373nm Fe(I)	383nm Fe(I)	406nm Fe(I)
Magnetite	592nm not assigned[1]	567nm	Broad black body
Marble	395nm Ca(II)	616nm Ca(I)	423nm Ca(I)
Plastic	435nm Appears to be fluorescence		
Pyrite	588nm Na(I)	Broad black body	
Sparkler	396nm Al(I)	455nm Ba(I)	383nm Mg(I)
Sulfur	None	None	None
Stainless steel	520nm Cr(I)	373nm Fe(I)	383nm Fe(I)

1) Likely not an atomic spectra, possibly a diatomic species.

The plastic sample is notable for its extremely high intensity variability, in some cases saturating the detector. An example of a normal plastic sample and a sample saturating the detector is shown in figure 3.4. This variability was caused by the target being moved in relation to the focus area of the laser to simulate a moving target. While the general shape of each spectrum was the same, barring detector saturation, spectra taken with better focus resulted in a significantly more intense emission. This spectrum is not the expected LIBS from a plastic sample. This emission in the 400-500nm range appears to be a fluorescence-based emission. Detector saturation can happen for a variety of reasons and thus a procedure for detecting and removing saturated data is required.

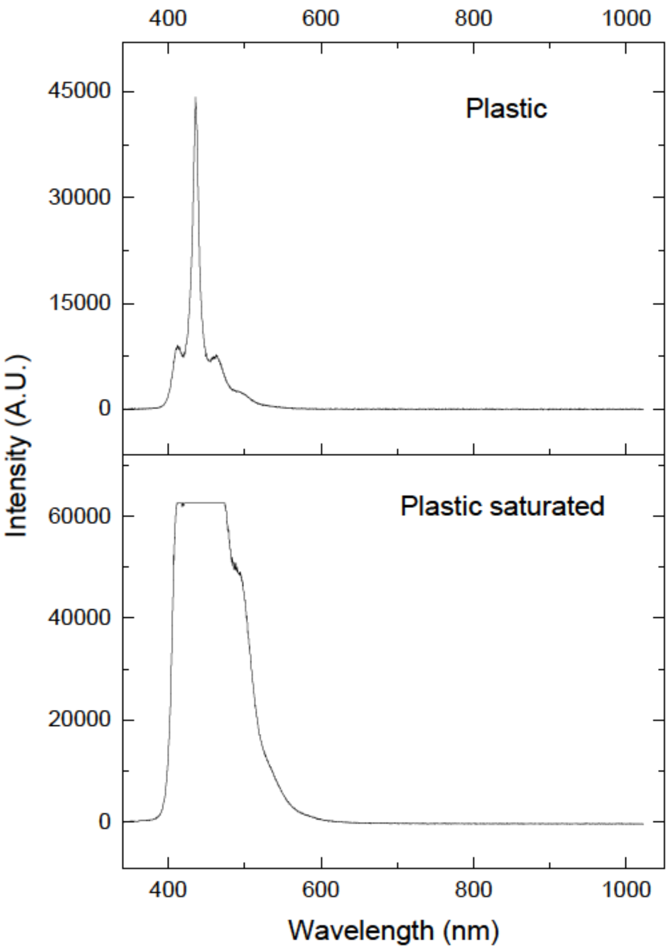


Figure 3.4 showing a high intensity plastic signal (top) and an oversaturated (bottom) LIBS spectra from plastic.

The saturation shown here indicates a clear need to set an upper limit to the value of data to be considered. In this case an upper limit response intensity of 60,000 was implemented to ensure no saturated or near saturated samples were considered. Saturated data, where the shape is distorted, was discarded as it cannot be confidently assigned to a material class. A simple check was performed for the maximum intensity value of each spectrum and spectra with a maximum value equal to or greater than 60,000 were deleted.

In more than 90% of calcite samples, plasma initiation did not occur. When plasma initiation did not occur, no clear spectrum was produced, conversely a clear LIBS spectrum was produced when plasma formed. Both a calcite signal and a non-plasma event spectrum are show in figure 3.5. The high rate of non-plasma events with this sample may have been due to surface morphology. Particular angles on the sample surface may prevent enough energy reaching a single point for plasma initiation. Surface morphology is more likely the cause in this situation as a plasma was able to form in a small percentage of LIBS pulses. This sample highlights the need to determine when plasma initiation does not occur to prevent incorrect assignment from data collected from a non-plasma event.

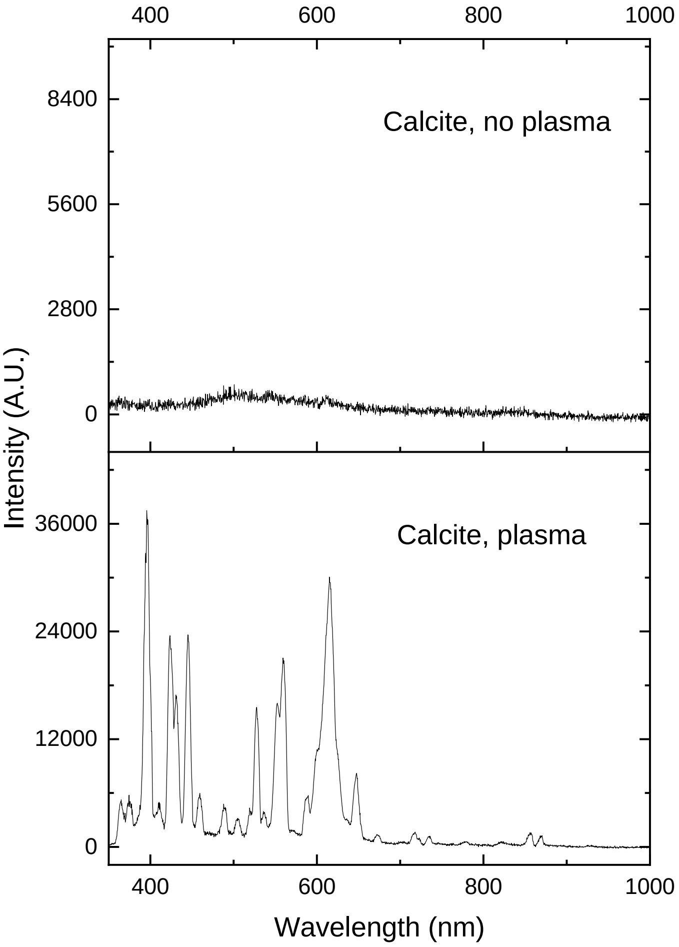


Figure 3.5 top, a calcite LIBS signal and bottom, a non-plasma event from a calcite sample.

Two methods were employed to resolve low signal data in such instances. First, for samples like calcite where a plasma forms intermittently a lower limit on the data was imposed. Any sample that did not reach a minimum value was removed from the dataset. In this case, the lower limit a sample must reach was a maximum intensity of over 800. A second solution was also developed for the sulfur class. This class had similarly not initiated LIBS plasma never producing a LIBS signal. In the case of sulfur this appears to be more closely related to laser energy as there was no spectrum produced from this sample. As various factors such as surface morphology and laser intensity can cause this to occur this class was kept as a "no plasma" class. This class is labelled as "sulfur" throughout this investigation but represents a class of samples in which plasma was absent, but some light was still observed. The values of upper and lower limit were determined empirically for the flame detector and will vary from detector to detector.

Pyrite samples produced a black body emission spectrum with a peak at 593nm that if characteristic, would be ascribed to sodium(I) and is possibly present as a consistent impurity. The spectrum lacks evidence of iron such as the large peaks expected at 404nm and 373nm or sulfur peaks expected at 675nm or 605nm as would be expected from pyrite. This signifies that while plasma did form it was not a plasma formed from pyrite but rather a sodium containing contaminate either on the surface or part of the natural mineral sample. This spectrum was consistent across all attempts and is not wholly unexpected as sulfur is difficult to ablate with LIBS [79]. This is not a clear LIBS spectrum however having this broad result in the database allows for this response to be categorised rather than it being applied to another class.

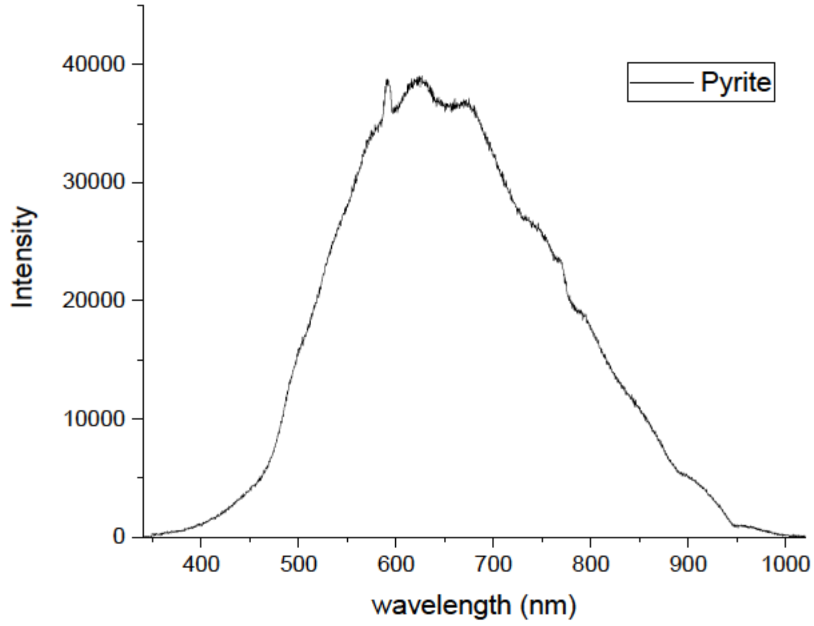


Figure 3.6 Pyrite sample, here showing a blackbody emission with a sodium(I) peak visible at 590nm

Broad responses such as those collected from pyrite perform a valuable role in the dataset. These samples represent real and common responses that may be encountered for a variety of reasons when using LIBS that are not necessarily diagnostic. Producing a system that can account for these responses rather than simply ignoring them will result in a more robust system, better able to inform the user as to the nature of an object.

This dataset also contains a large number of mixtures such as iron and stainless steel which visually share the iron spectrum with additives creating additional peaks from 500-550nm distinguishing them. Copper and brass similarly share the same clear copper peaks, though the additives for distinguishing brass appear between 450-500nm in this spectrum. Some of the mineral samples notably calcite and marble are both $CaCO_3$ based and as such have nearly identically shaped spectra, though marble's spectra is of lower intensity and thus peaks are more clearly defined. These spectra with similar shapes and features may lead to confusion in classification however being able to separate these classes will demonstrate a good separation ability by the classifier.

3.2.2 Dataset 2

The second and third datasets were smaller than the first and both contain LIBS and RS data fused at the point of detection. The second dataset consisted of 142 samples of 7 classes as shown in figure 3.7. The dataset was collected on the Andor iStar iCCD with Shamrock 303i spectrometer.

Table 3.4 Formulas and sources for samples in Dataset 2 [78].

sample	formula	Source
Ammonium nitrate (AN)	NH_4NO_3	DSTG
Ammonium nitrate aluminium (AN_Al)	NH_4NO_3/Al	DSTG/Flinders university workshop
Aluminium	Al	Flinders University workshop
Marble	Ca	Flinders University rock collection
Copper	Cu	Flinders University workshop
Iron	Fe	Flinders University workshop
Potassium chlorate	KClO3	Sigma Aldrich

[80]

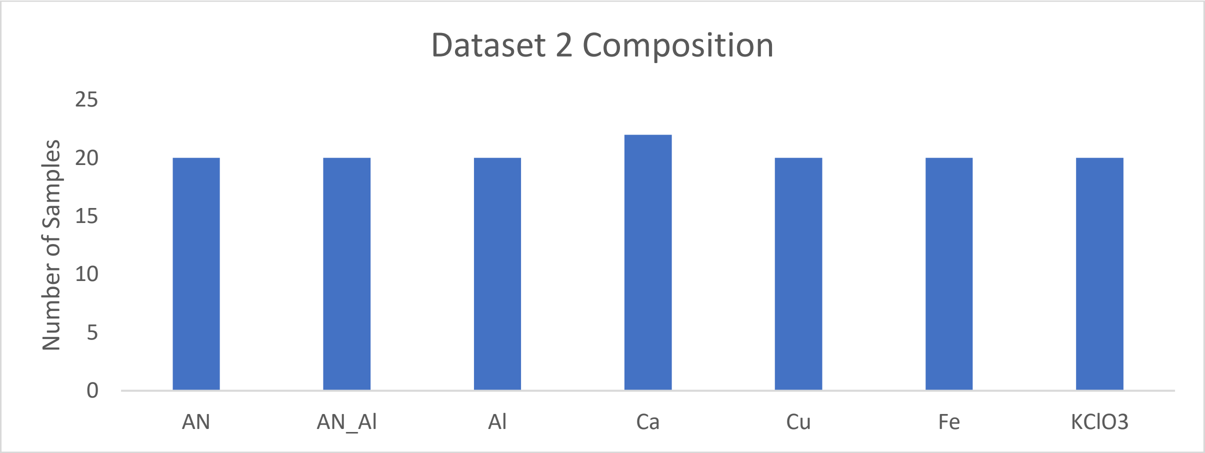


Figure 3.7 Composition of Dataset 2.

RS data for this investigation will be presented utilising the wavelength of collected light rather than Wavenumber or Raman shift wavelength. This is due to the direct comparison between RS and LIBS data and the need for consistency in units.

This dataset was designed as a check against the first dataset. The first dataset was intended as a preliminary test, this dataset is a more focused test of the same settings developed on the first dataset. This dataset consists of a smaller number of metal and energetic samples. Errors within this dataset indicate that the algorithm parameters are becoming over-specialised to the first dataset.

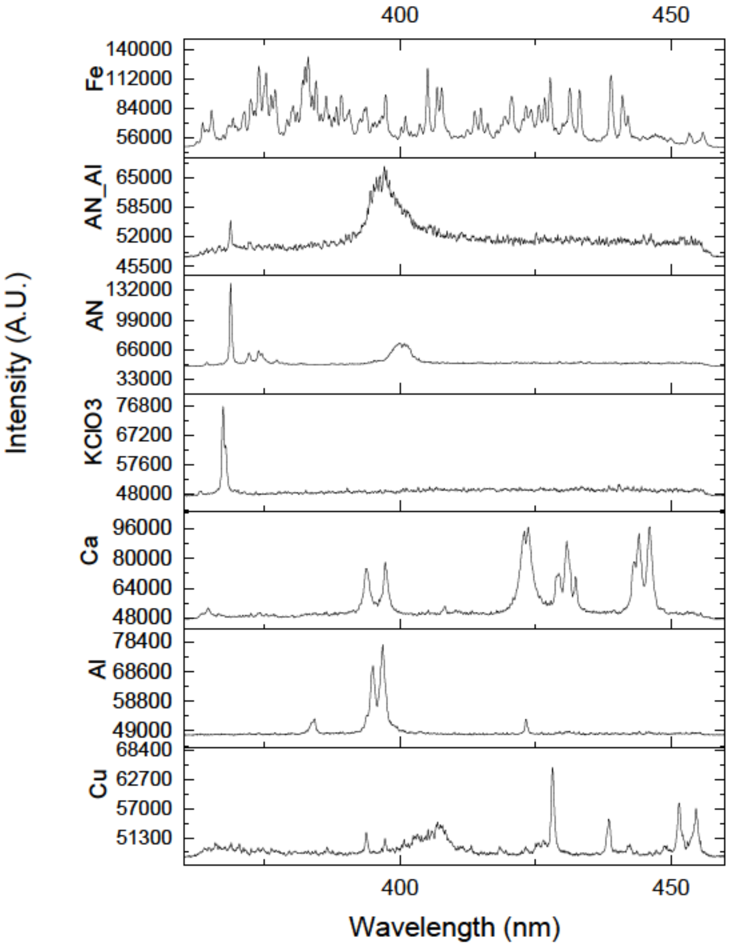


Figure 3.8 Example spectrum from each sample in Dataset 2.

Table 3.5 Key peaks of LIBS and RS spectra in Dataset 2

Sample	Peak 1	Peak 2	Peak 3
Ammonium nitrate (AN)	368nm NO_3(v1)	400nm	372nm NO_3
Ammonium nitrate aluminium (AN_Al)	368nm NO_3(v1)	396nm Al(I)	
Aluminium	396nm Al(I)	394nm Al(I)	
Marble	422nm Ca(I)	445nm Ca(I)	396nm Ca(II)
Copper	427nm cu(I)	451nm cu(I)	453nm cu(I)
Iron	373nm Fe(I)	382nm Fe(I)	438nm Fe(I)
Potassium chlorate	367.5nm ClO_3		

[80]

The peaks of the data in Dataset 2 are well separated between samples. The only pair of samples where this is not the case is ammonium nitrate (AN) and the ammonium nitrate/Aluminium (AN_Al) mixture. As this dataset is well separated, errors are not expected within this dataset other than between AN and its mixture. Errors in this dataset indicate a major error within the classifier that needs to be examined more closely as a dataset with such clear peaks should not cause confusion. This dataset also has a large flat background of approximately 48,000 counts. This background is due to the number of individual spectra averaged into each spectrum, in this case 100 spectra averaged into each spectrum. As this background is consistent, the data is directly comparable without background removal.

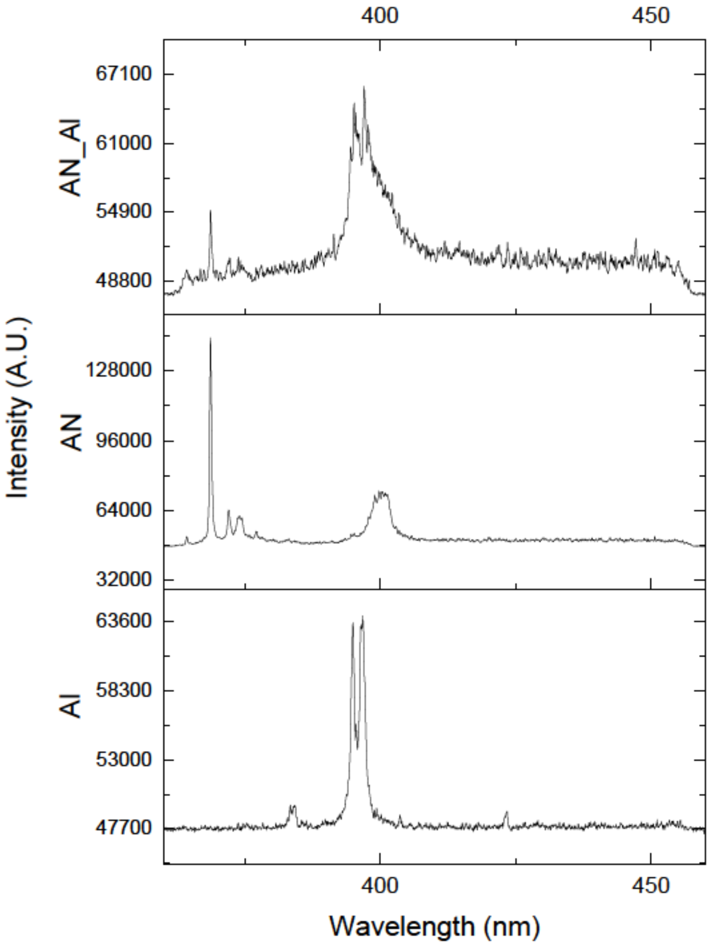


Figure 3.9 The AN, Al, and mixture's spectra, the mixture produces a weak signal for both AN and AL. The combined sample showing both peaks with a significantly lower response.

3.2.3 Dataset 3

Dataset 3 is primarily focused on organic compounds and RS and was again produced with the Andor iStar iCCD and Shamrock spectrometer. It contains 280 spectra from 14 classes, each class having 20 examples as shown below in figure 3.10.

Dataset 3 contained a large number of energetic and energetic like mixtures as well as other organic compounds. This dataset is designed to examine the classifiers' ability to distinguish similar RS spectra and ensure it does not become too specific to LIBS.

Table 3.6 Formulas and sources for the samples in Dataset 3 [78].

Sample	Proposed formula	Source
Ammonium nitrate (AN)	NH_4NO_3	DSTG
Ammonium nitrate fuel oxide (ANFO)	NH_4NO_3 and fuel oxide	Mixture, fuel oxide is variable
Ammonium nitrate/potassium chlorate (AN_$KClO_3$)	NH_4NO_3 and $KClO_3$	DSTG/Sigma Aldrich
Ammonium nitrate/potassium perchlorate (NA_$KClO_4$)	NH_4NO_3 and $KClO_4$	DSTG/Sigma Aldrich
Ammonium nitrate/potassium perchlorate/xylitol (AN_$KClO_4$_xylitol)	NH_4NO_3, $KClO_4$ and $C_5H_{12}O_5$	DSTG/Sigma Aldrich/Nirvana Organics
Calcium ammonium nitrate (CAN)	$Ca\ H_4N_2O_3$	DSTG
Hexamine	$C_6H_{12}N_4$	Tokyo Chemical Industry
Potassium perchlorate	$KClO_4$	Sigma Aldrich
RDX in ore	$C_3H_6N_6O_6$	Mixture prepared from RDX (DSTG), and mine tailings (gold mine, Kalgoorlie region)
Nylon	Nylon block	Flinders University workshop
Urea	CH_4N_2O	Sigma Aldrich
Xylitol	$C_5H_{12}O_5$	Nirvana Organics
Ore	Dirt	Mine tailings (gold mine, Kalgoorlie region)
Plastic holder (polyethylene)	$(CH_2)_n$	Polyethylene

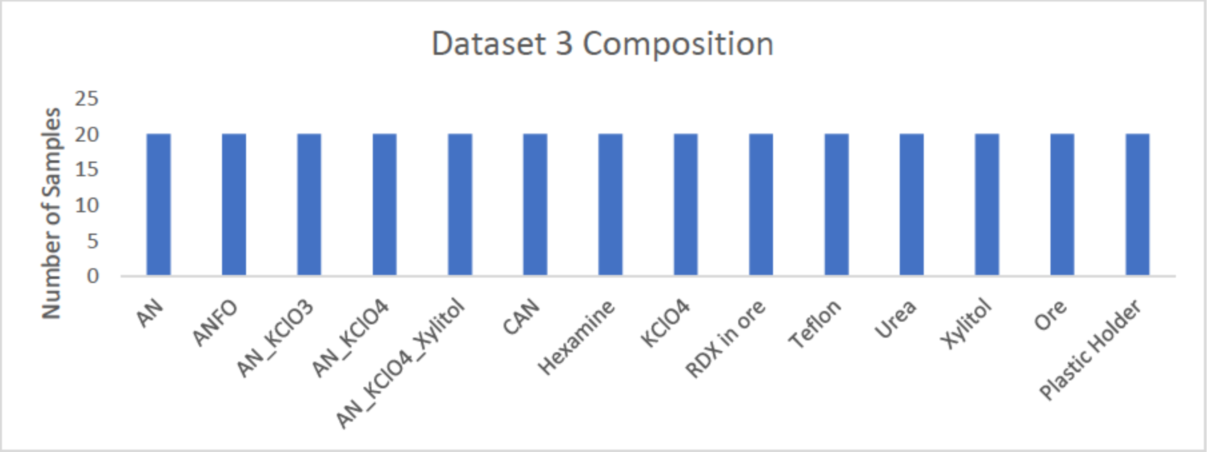


Figure 3.10 Composition of Dataset 3

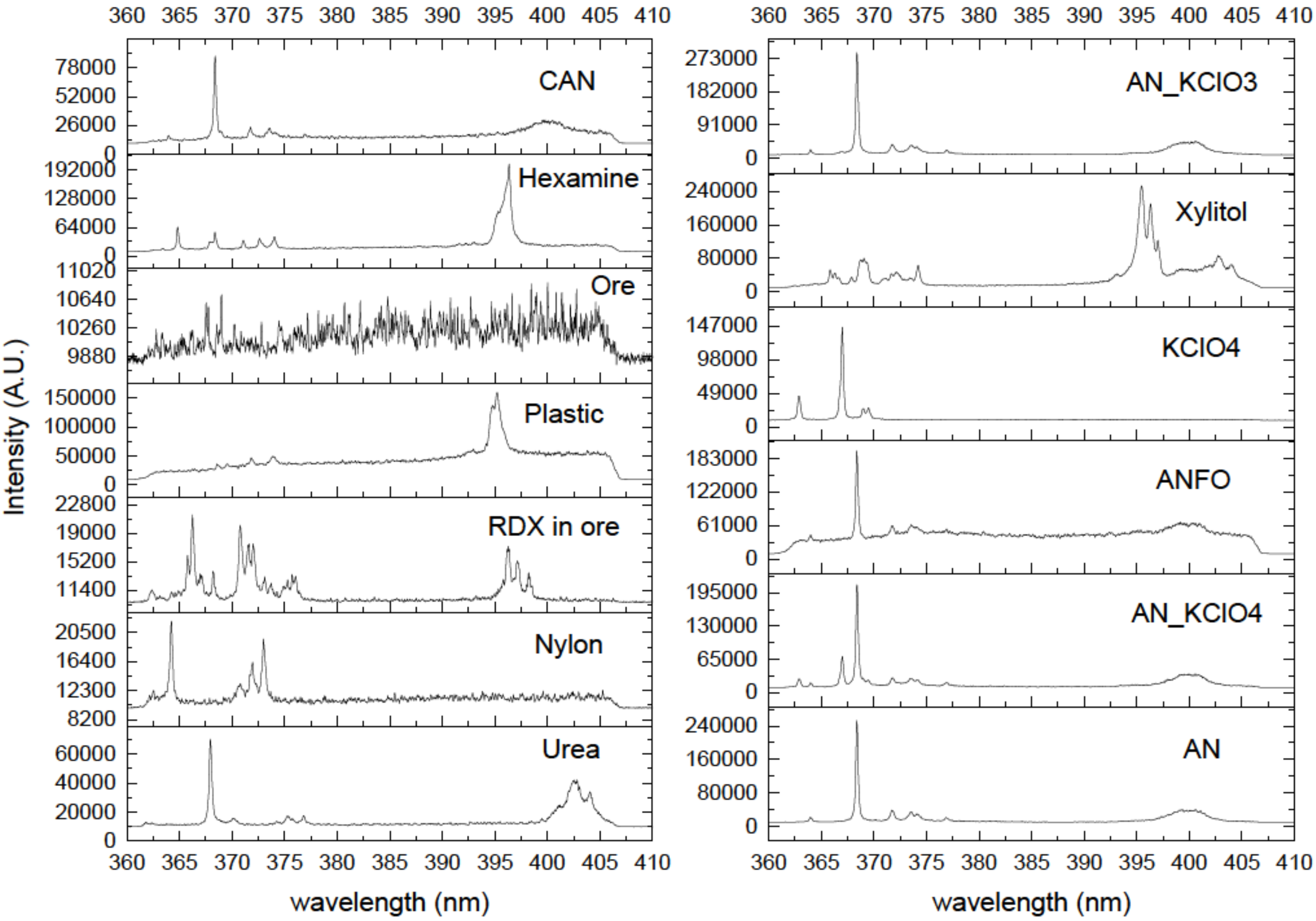


Figure 3.11 Example spectrum of each sample in Dataset 3

Table 3.7 Key peak assignments for Dataset 3.

Sample	Peak 1	Peak 2	Peak 3
Ammonium nitrate (AN)	368nm NO_3(v1)	372nm NO_3	
Ammonium nitrate fuel oxide (ANFO)	368nm NO_3(v1)	372nm NO_3	Fluorescence
Ammonium nitrate/potassium chlorate (AN_$KClO_3$)	368nm NO_3(v1)	372nm NO_3	
Ammonium nitrate/potassium perchlorate (NA_$KClO_4$)	368nm NO_3(v1)	367.5nm ClO_3	
Ammonium nitrate/potassium perchlorate/xylitol (AN_$KClO_4$_xylitol)	368nm NO_3(v1)	395nm CH_2	
Calcium ammonium nitrate (CAN)	368nm NO_3(v1)		
Hexamine	368nm C=O	402nm amine	
Potassium perchlorate	367nm ClO_4 v1	362nm ClO_4 V4	
RDX in ore	366nm ring stretching	396nm CH_2 stretch	271nm NO_2 stretch
Nylon	364nm C-C	373nm C-C	
Urea	368nm C=O	402nm amine	
Xylitol	395nm CH_2	402nm OH	
Ore	None		
Plastic holder (polyethylene)	395nm CH_2	Fluorescence	

[80, 81]

Dataset 3 contains well separated samples such as xylitol and CAN in which peaks are distant and very obviously separate. This dataset also contains poorly separated samples such as AN, CAN and AN_$KClO_3$ where peaks present are at similar positions. Errors in classifying the more poorly separated spectra are a less serious issue. Confusion between well separated spectra show a more significant error in the classifier and present a larger concern for its effectiveness.

ANFO shows a fluorescent background in its spectra. This background could be mathematically subtracted but as it is not likely to reduce the accuracy of classifiers trained on this data, it has not been removed. The plastic sample also produces a fluorescent background in its spectra but again, a machine learning algorithm needs to be able to work with data in the form it is collected in this application. This system is designed to remove the need for significant training for the operator. To achieve this aim, an automatic pre-processing algorithm is required. Removing fluorescent backgrounds entirely automatically without altering the spectra is difficult, even in lab situations [82], thus the fluorescence will be included in the dataset unaltered.

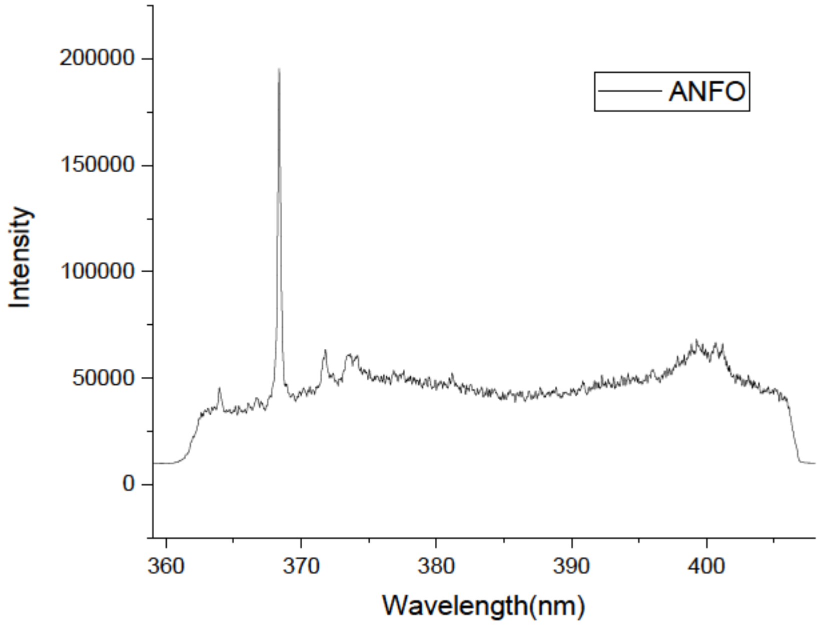


Figure 3.12 The fluorescence here can be seen by the broad raised signal from 363nm through to 405nm. Peaks are still visable through the fluorescence however with a larger array of samples fluorescence may become a larger accuracy problem.

The “ore” sample does not produce a strong response under these conditions. This sample is here primarily for use with the sample “RDX in ore” where the ore was used as a stabilisation agent for the hazardous compound. The “ore” sample is also a possible contaminant or “missed target” sample. It is used in this analysis as it is likely that the system intended for this application will encounter similar materials and recognising them may prove advantageous.

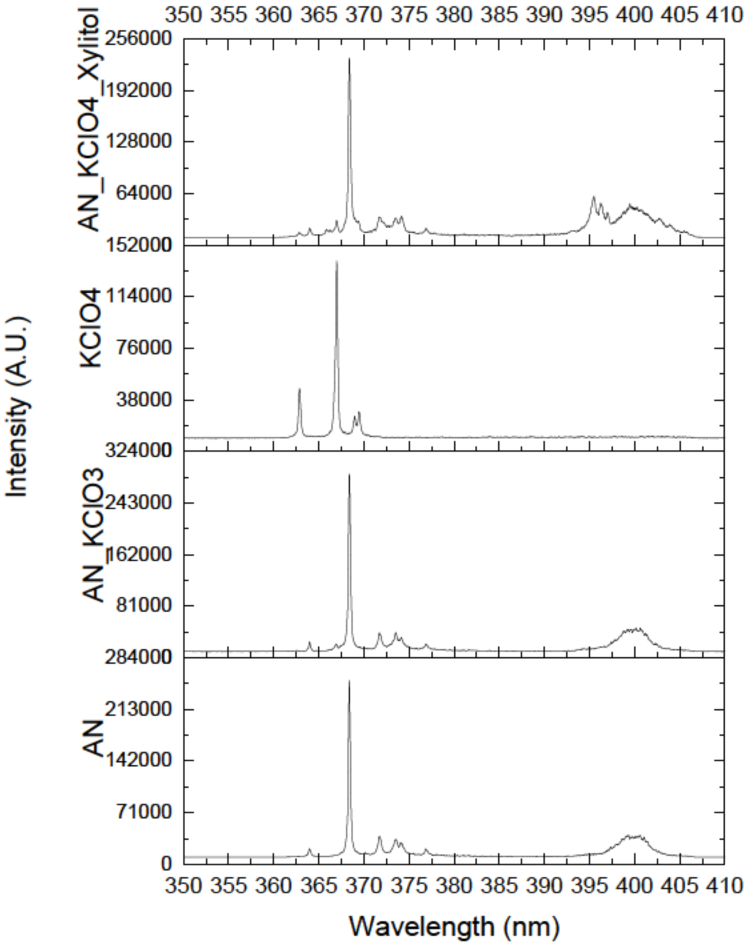


Figure 3.13 Stacked plot showing several AN and potassium mixtures.

Most of the peaks are shared between the samples however some such as the xylitol peak around 395nm. The similarity is high for AN and AN_KClO3 however there is a small peak at 367nm allowing the two to be differentiated.

3.3 Pre-Processing

The data was first examined for notable inconsistencies and samples showing a large, inconsistent inclusion of another material were removed or, in the case of barite, split into two separate classes. Collected and averaged spectral data was tabulated into a matrix aligned by wavelength and imported into MATLAB. This matrix contained each spectrum and its sample as a row of x values. If necessitated by the detector the data was trimmed removing leading and lagging zeros. Once imported into MATLAB samples with either too high or too low signals were removed, the thresholds for these values were determined for Dataset 1 to be responses below 800 and above 60,000. These thresholds were calculated by the detector limit on the high end and the level of noise on the lower end. Samples that produced low to no signal or that saturated the detector were removed.

In some cases, Principal Component Analysis (PCA) was used to reduce the number of x values in the matrix. This was done to reduce the dimensionality dataset. Two different methods were used to select the number of PCs to consider, a set number of PCs (4-64) or a percentage of explained variance (95-99%). In comparison tests the MATLAB “Relieff” or ReliefF function was also used as a means to reduce the dimensionality of the dataset. When using a “Relieff” function each predictor, in this case each point in the spectrum, is ranked by its importance to classifying the sample. The calculation is done by comparing each predictor to a number of others and increasing its importance if those of other classes are dissimilar to it [73, 83]. This allows predictors with low importance to be discarded to reduce the dimensionality of the dataset.

The data was then split into a training set and a testing set and a validation set. This was done in MATLAB by taking samples randomly into a different dataset to produce a testing set one third the size of the training set and a smaller validation set. Code for splitting datasets can be found in appendix MATLAB code.

In some of the classifiers mean centred standardisation was performed on the data before training. This is a method of data standardisation in which the mean (μ) and standard deviation (σ) of each position (i) on the spectra is calculated. Every value x in every sample has the average of that position subtracted and is that divided by the standard deviation.

$$x_{i(s)} = \frac{x_i - \mu_i}{\sigma_i} \qquad \text{Equation 3.1}$$

3.4 Linear Discriminant Analysis

The first classifier developed was the LDA algorithm. This system had few parameters for optimisation within MATLAB. The available options were the structure of the covariance matrix, either using the full covariance matrix or the diagonal of this matrix, and the use of PCA. Tests were performed with and without PCA on both covariance structures. When utilising PCA 8PCs, 16PCs and 95% explained variance were used. This PCA used the singular value decomposition algorithm, centred by the subtraction of column means and used no observation weighting. Different methods of selecting the number of PCs were also tested. LDA classifiers were generated with the use of the classification learner MATLAB package.

3.5 K-Nearest Neighbour

The second classifier developed was the KNN algorithm. This system was optimised along several parameters. The first of these was the distance metric, as discussed above, distance metrics are different mathematical conceptions of similarity. The metrics considered were Euclidian, City block, Minkowski(cubic), cosine and Spearman, the initial state of this parameter was Euclidian. The second was the weighting of distance, the versions of distance weighting considered were none, inverse, and square inverse with the initial setting being no distance weighting. The third, the number of neighbours, K, which had a starting value of 10. The fourth, making use of PCA, the mode of PCA, with initial tests performed without PCA. The fifth, the use of mean centred standardisation on the data. Finally, tests were performed with three data sets to prevent method specificity. Further tests were performed changing the method of verification, from Five-fold to holdout using 20% holdout. A full list of experiments can be found in supplementary materials. Code for training this classifier can be found in appendix MATLAB code.

3.6 Artificial Neural Network

The third classifier, Artificial Neural Networks, could not be optimised as readily as the KNN algorithm. Different training optimisers were used and compared, those being, SGDM, RMSProp and *Adam*. Dataset 1 was split randomly into three different and unequal portions. One, the largest 588 samples of the total dataset (1323 samples) was used for training while the smaller two portions of 441 and 294 samples were used for verification and testing respectively. The other two datasets, due to their smaller size, were only split into two parts, 75% for training and 25% for testing. Minimal optimisation options are possible beyond data shuffle timing and training time due to both limited options and the high accuracy the classifiers were found to have. Code for the training of ANNs can be found in appendix MATLAB code.

3.7 Comparison Tests

Tests were then performed on the best version of each optimiser to understand how they deal with specific situations and oddities in data. The first property tested was effectiveness after dimensionality reduction in the dataset. To test this, two methods of dataset reduction were used Relieff and PCA. In Relieff tests the most important 1000, 500, 100, 50, and 10 data points were used to determine the degree of dataset reduction possible for this dataset. PCA tests were also conducted in order to examine the effects the dimensionality reduction method may have had on the results. In this instance 64, 32, 16, 8, and 4 PC tests were performed. These values were selected based on the number of classes using between 4 times the number of classes and a quarter times the number of classes for the PCs.

The second property tested in these tests was noise tolerance. Different levels of noise were added to the testing data and the value of the boundaries was varied for each test. The boundaries were ±, 1000, 3000, 6000, 9000, and 12,000 with noise randomly generated between the upper and lower bound added into each spectrum in the testing set. These values were chosen as they were between 20% and 1.67% of the maximum signal.

The final tested property was the classification of a sample not present, and unlike those in the training set. The barite impurity was used to test this property, it was removed from the training set for this test. This test examined how the different methods respond to unexpected data.

These comparison tests are based on the needs of the application where real collected data will likely include very noisy data, database size will cause problems in both RAM requirements and processing time and samples not matching any trained class will be encountered.

Chapter 4 – **Evaluating Performance of Machine Learning Techniques**

4.1 Introduction

The accuracy of a machine learning model is influenced by many different factors. The types of algorithms chosen have different tuneable factors that can influence the accuracy of these models. LDA is the least tuneable of the algorithms considered, its accuracy only able to be directly affected by the covariance matrix structure. The KNN and ANN models can be more extensively modified for purpose. Factors influencing accuracy for KNN that will be adjusted are the distance metric, distance weighting, number of neighbours and the use of mean cantered standardisation. ANNs are a more complicated system and as such have a correspondingly wider array of tuneable factors, however with the complexity and effectiveness of ANNs limited tweaking is viable. The properties that can be tweaked include learning time, learning rate and network design.

This chapter will focus on the effects of, and optimisation of, a selection of these parameters. They will be compared based on the number of

(i) **true positives** (TP), classifications that correctly labelling a sample as belonging to a given class;

(ii) **true negatives** (TN), classifications that correctly label a sample as not belonging to a given class;

(iii) **false negatives** (FN) when a sample from a selected class is incorrectly labelled as belonging to a different class; and

(iv) **false positives** (FP) when a sample is incorrectly labelled as belonging to a selected class.

Table 4.1 below shows an example of how true and false positives are defined in a binary confusion matrix.

Table 4.1 Listing the four possible states of a prediction. With the true label being either 1 or 2 and the predicted label of 1 or 2 denoting the model's prediction. This table assigns false positives and negatives for label "1".

	Predicted label "1"	Predicted label "2"
True label "1"	True positive	False negative
True label "2"	False positive	True negative

These four measures (TN, TP, FN, FP) can be used to calculate accuracy precision recall and F-score metrics [84]. These standard measures of model effectiveness highlight different properties of the model and are calculated as shown in equations 4.1 to 4.4.

The primary measure used, *accuracy*, is dominated by the number of true positives. It is most effective when all classes are equally important and similar numbers of samples from each class are available to the model. The other measures become increasingly important and are used to explore results where more complicated dataset conditions, such as varied numbers of spectra between sample classes in the dataset are present. In such cases the accuracy of the model on a class can become overwhelmed by samples that have more spectra in the testing set. The model achieves high accuracy while having large limitations to the ability to classify spectra of samples with low representation. *Precision* is a measure of false positives, precision lower than accuracy indicates a higher rate of false positives in samples with less spectra than the average in the dataset. *Recall* is a measure of false negatives, a recall value lower than that of accuracy indicates samples with less representation have a higher degree of false negatives. The final metric, the *F1 score,* is the harmonic mean of the precision and the recall. This allows a balanced view of both false positives and false negatives [84, 85]. Multiclass metric calculations performed using MATLAB code made by *Manjunatha,P* [86].

$$\frac{TP + TN}{TP + FN + TP + TN} \qquad \text{Equation 4.1}$$

Formula for the calculation of prediction accuracy.

$$\frac{TP}{TP + FP} \qquad \text{Equation 4.2}$$

Formula for the calculation of precision.

$$\frac{TP}{TP + FN} \qquad \text{Equation 4.3}$$

Formula for the calculation of recall.

$$\frac{2 \times TP}{(2 \times TP) + FP + FN} \qquad \text{Equation 4.4}$$

Formula for the calculation of F-score [84].

An ideal system would produce 100% accuracy and further metrics would be unnecessary. Such a system would not be expected from the beginning of the optimisation process and thus other metrics assist in grading a system. For a system to be considered useable, 90% in each metric must be achieved, a good system is one in which 95% is achieved on each metric and a great system would achieve 99% in each metric.

4.2 Statistical Approach

The statistical approach used LDA to classify unknown samples from the dataset. The possible optimisations explored for LDA were the use of PCA and the structure of the covariance matrix. The first parameter tested for the LDA system was whether a diagonal covariance matrix produced adequate results when compared to a full covariance matrix. A diagonal covariance matrix considers less data and is thus a faster and less computationally intensive system. PCA was also used to reduce the amount of data considered to try to reduce the system load of the process. The different forms of PCA used were PCA with a set number of PCs using both 8 and 16 PCs. 16 and 8 were chosen as they were proportional to the number of different sample classes in the dataset. The final form of PCA used was explaining 95% of the variance within the dataset, a standard form of PCA.

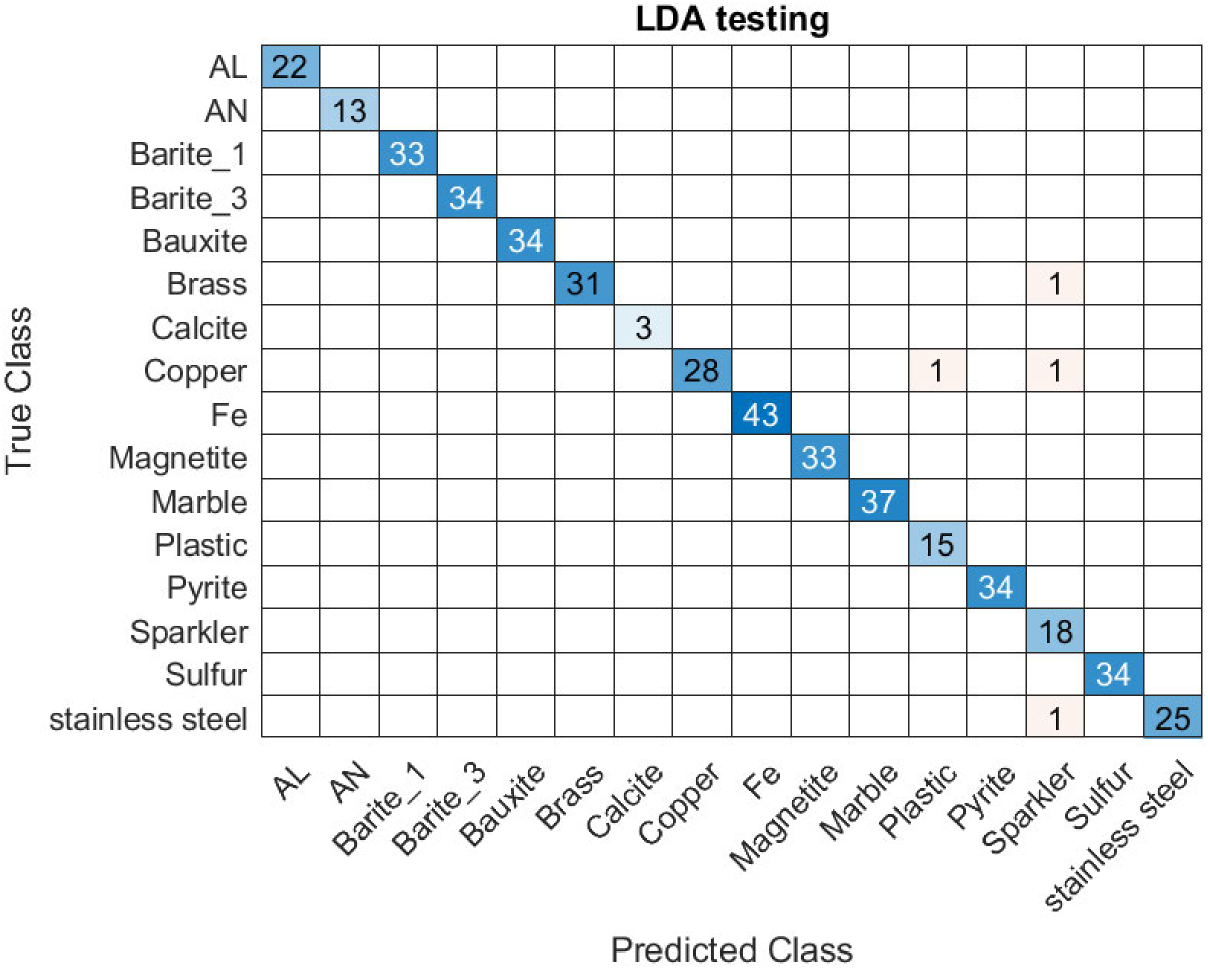
Results

An LDA model was trained with each of the above parameters on training data (882 spectra) from Dataset 1 utilising 5-fold verification. These trained models were then used to classify test data (441 samples) separated from Dataset 1. The results of these tests are reported in table 4.2. Tests were also performed on Datasets 2 and 3 utilising holdout validation. In all conditions Datasets 2 and 3 were classified 100% correctly.

Table 4.2 Results of the trained LDA models on testing data.

	Accuracy	Precision	Recall	F-score
Full covariance	99.1	98.7	99.2	98.9
Diagonal covariance	87.1	89.1	86.1	86.0
PCA explained variance 95%	73.0	77.4	76.3	74.4
PCA 8 components	84.1	88.9	85.2	85.3
PCA 16 components	93.0	94.9	88.5	90.0

The best LDA model was the full covariance algorithm not making use of PCA. It was found to be 99.1% accurate on the testing data. It achieved a precision of 98.7%, a recall of 99.2% and an F1-score of 98.9% indicating that this system produced more false positives than false negatives proportionately. The second most effective model for all metrics was the 16 component PCA followed by the diagonal structure without PCA then the 8 component PCA and explained variance PCAs. The first and second best models both achieved over 90% accuracy however the second model did not reach 90% recall indicating a proportionally high number of false positives. The second model struggles to identify samples with low numbers of spectra. The confusion matrix of the full covariance matrix LDA on the testing data is shown in Figure 4.1.



	AL	AN	Barite_1	Barite_3	Bauxite	Brass	Calcite	Copper	Fe	Magnetite	Marble	Plastic	Pyrite	Sparkler	Sulfur	stainless steel
AL	22															
AN		13														
Barite_1			33													
Barite_3				34												
Bauxite					34											
Brass						31								1		
Calcite							3									
Copper								28				1		1		
Fe									43							
Magnetite										33						
Marble											37					
Plastic												15				
Pyrite													34			
Sparkler														18		
Sulfur															34	
stainless steel														1		25

Figure 4.1 The Confusion matrix of the best LDA system on the testing data.

The noted errors in figure 4.1 are difficult to explain. As shown in Figures 4.2 and 4.3 the incorrectly classified spectra are all very low signal and noisy. The copper confused for a plastic spectrum is likely due to its overall low signal. The incorrectly classified copper spectrum has a small peak around 430nm where the primary plastic peak is located but overall, it has an extremely low signal compared to the normal copper spectra. There are several plastic spectra with extremely low signal and in spite of the shape of the incorrectly classified copper spectrum being preserved, (three primary peaks between 510-520nm) the low signal causes the confusion. The other three spectra all confused for sparkler samples show the same low signal confusion. The sparkler sample showed a high variability in signal, not equal to that of plastic but between this variability and the larger number of peaks in the sparkler signal these errors can be explained in a similar fashion to the prior.

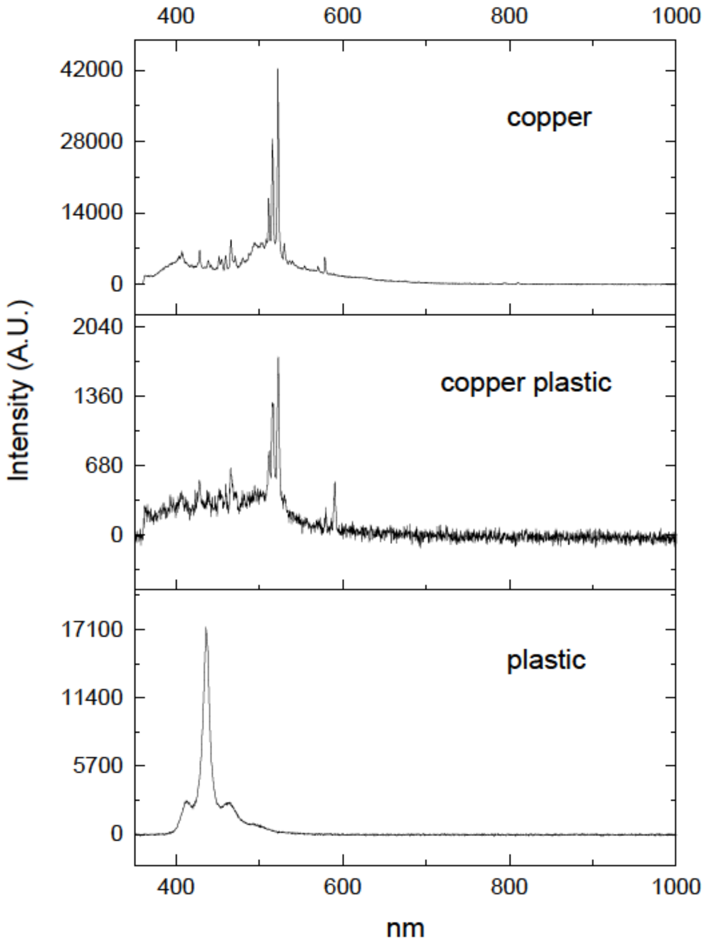
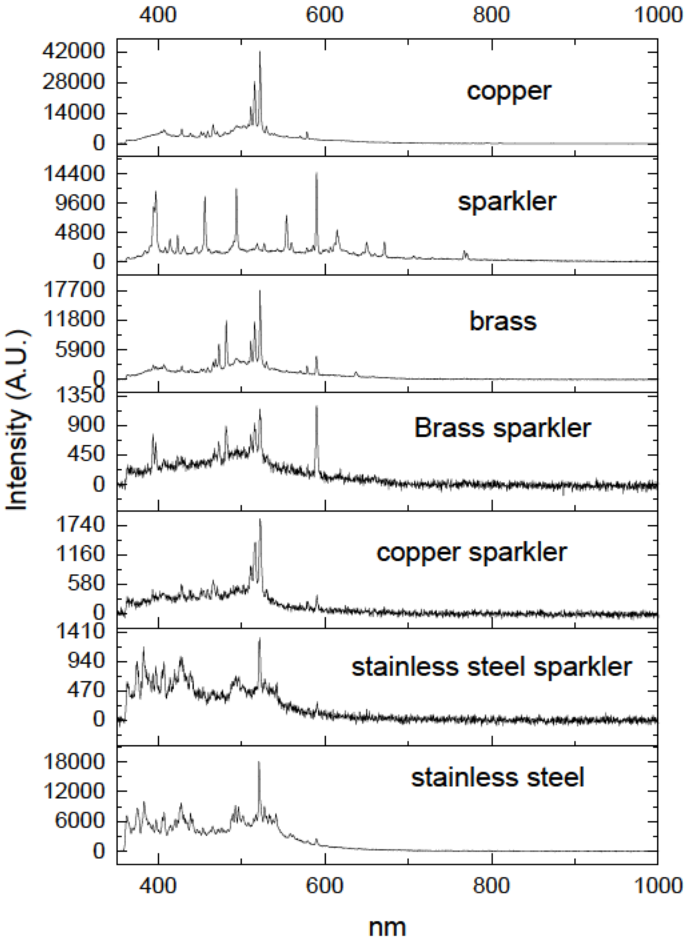


Figure 4.2 (top) A normal copper spectrum, (middle) the copper spectrum that was misclassified as plastic, and (bottom) a moderate signal plastic spectrum.



Figures 4.3 Average copper, sparkler and brass samples, followed by the data that was incorrectly labled as sparklers and finally a normal stainless steel sample.

All samples that were falsely classified possesed spectra with very poor signal:noise ratios.

4.3 K-Nearest Neighbour

For the KNN, the second algorithm tested, more optimisation was possible as there were a larger range of parameters that could be modified. Modifications were made to the distance metric, the number of neighbours considered in the model, the use and method of distance weighting, and utilising mean centred standardisation.

Table 4.3 The initial state of the KNN model using Euclidian distance, 10 neighbours and mean centred standardisation.

Metric	K Value	Distance Weighting	Mean Centred Standardisation	Data Culling
Euclidian	10	None	on	800/60000

4.3.1 Distance Metric

Five choices of distance metrics were tested for their performance against the three datasets. The distance metrics tested were Euclidian, City block, Minkowski (cubic), Cosine and, Spearman Correlation as defined earlier in equations 2.1-2.5 in section 2.6. The set of distance metrics tested were chosen as a mixture of direct and correlational distance metrics. This allows for a mixture of metrics that calculate direct distance including intensity and systems that are more focused on comparing the shape of spectra. Much of the work utilising KNN classifiers on LIBS and RS only utilises Euclidian distance metrics [87, 88] however other metrics such as Cosine and Minkowski are not unheard of [89] and are further considered with other techniques [90, 91]. Testing and training datasets were used to calculate accuracy measures for Dataset 1. Accuracies for Datasets 2 and 3 were calculated using 25% holdout verification in which a quarter of the data is not used in training in order to be used in testing.

Table 4.4 Effectiveness of each distance metric on each data set.

	Set 1 Accuracy	Precision	Recall	F1-score	Set 2	Set 3
Euclidean	95.7	95.6	96.3	95.6	100	100
City block	95.9	95.6	96.7	95.8	100	100
Minkowski	95.7	95.6	96.3	95.6	100	90
Cosine	95.5	95.2	95.8	95.2	97.9	84.3
Spearman	96.4	95.8	96.9	96	100	98.6

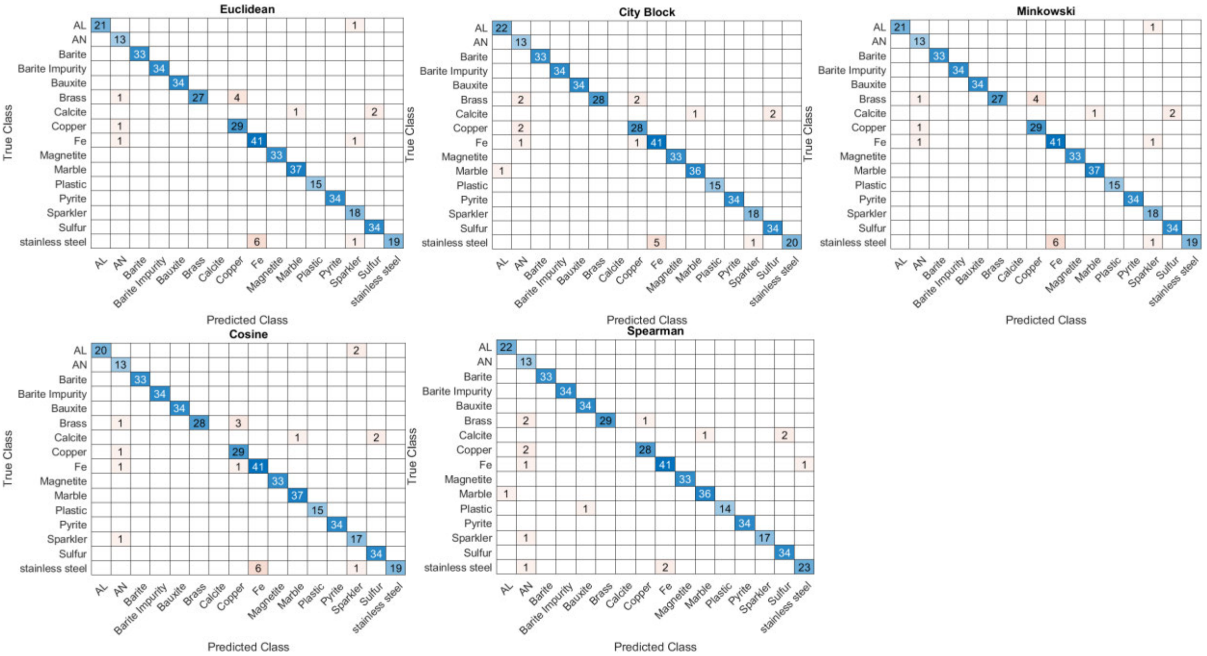


Figure 4.4 Confusion matrices of the five different distance metrics considered.

The most accurate of these on Dataset 1 were the City block and Spearman metrics. These metrics were also the only two metrics to have zero false negatives for both of the energetic materials, a key focus of the project team. Both the Spearman and City block metrics had high false positive rates for AN, giving specific precisions of only 65% and 72.2% respectively.

The other notable errors in the City block model are a stainless-steel recall of 76.9%, though most of the false negatives for stainless steel were classified as iron. This model was also unable to classify any calcite spectra. As only 6 spectra of calcite were included in the training data this is unsurprising. Without distance weighting and with a K larger than the number of samples the accuracy of any unknown sample classification is unlikely to be high.

The Spearman metric produced a lower AN precision than the City block metric, however all other metrics for Dataset 1 were higher. As was expected with these settings this model was also unable to correctly classify any calcite samples. The Spearman metric was also less accurate on Dataset 3, however this amounted to one sample of AN being classified as AN_KClO3.

4.3.2 Distance Weighting

Distance weighting is the practice of considering nearer samples more strongly indicative of classification than samples further away. Mathematically there are multiple ways of doing this, but a simple method is to multiply the vote of each point by the inverse of the distance to that point.

Three different approaches to distance weighting were trialled, equal weighting, inverse distance weighting, and square inverse distance weighting. The results reported in table 4.5 are for squared inverse as this proved to be more effective than inverse weighting.

Table 4.5 Effect of distance weighting on each distance metric and dataset

	Set 1 Accuracy	Precision	Recall	F1-score	Set 2	Set 3
Euclidean	96.6	96.5	94.9	95.2	100	98.6
City block	97.3	97.3	95.6	96.1	100	100
Minkowski	96.6	96.5	94.9	95.2	100	90
Cosine	96.2	96	94.4	94.7	100	75.7
Spearman	97.5	97	95.7	96	100	97.1

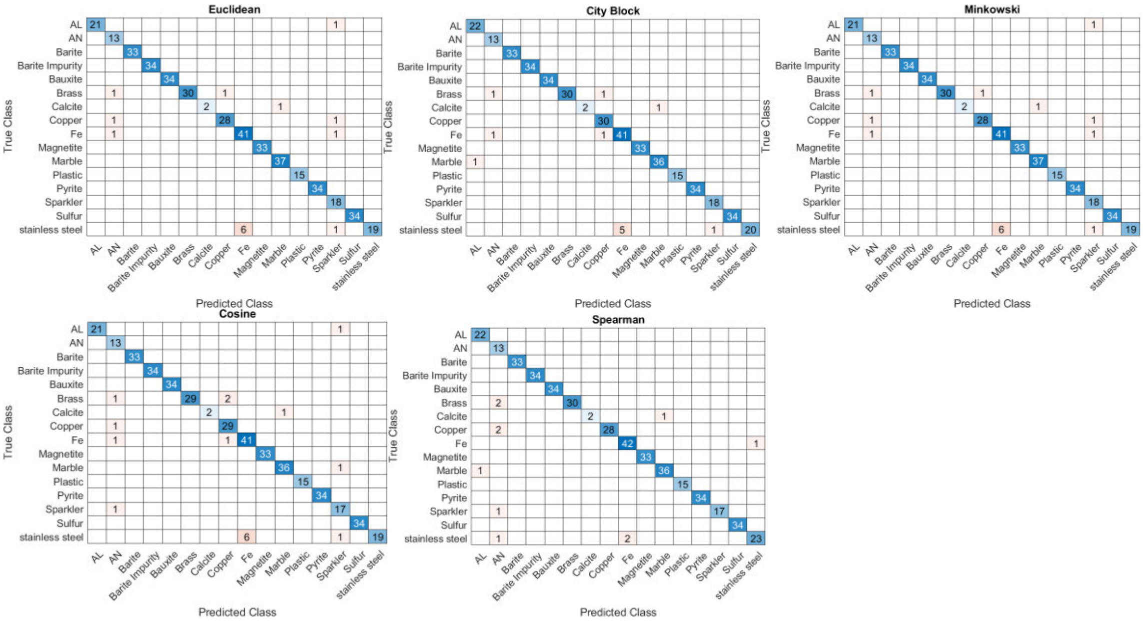


Figure 4.5 Confusion matrices of each distance matrix with squared inverse distance weighting.

Spearman and City block again proved to be the best measures across all metrics. Though again, Spearman produced a specific AN precision of 68.4% having many false positives, most of which were unchanged from the prior condition. This is a problem for an energetics detection program as it would lead to time lost investigating objects which were ultimately harmless.

The accuracy of the Spearman metric showed a similar improvement over all other samples. Two of the three calcite spectra now being correctly identified and errors in other samples lowering. The majority of the errors in the model were failing to differentiate between two low signal sample classes (calcite and sulfur), the AN false positives and stainless steel/iron confusion. The metric did however perform slightly worse on Dataset 3 making the same false classification of AN spectra as AN_KClO3 in two instances.

The City block metric, while less accurate overall, achieved an AN precision of 87%, a significant improvement over its own prior result and the Spearman metric. It was also capable of identifying the same two calcite samples as the Spearman metric. The stainless steel recall was 76.9%, showing no improvement. The consistency of this error shows this metric struggled to differentiate between extremely similar spectra.

4.3.3 Number of Neighbours

The number of neighbours considered was interrogated using both the Spearman and City block metrics as these were the most accurate in tests thus far. Values for K of 1, 5, 10 ,20, 30, 40, 50, and 100 were all tested for accuracy. Tests were conducted using squared inverse distance weighting and equal distance weighting.

Table 4.6 The percentage accuracy of the tested metrics at different values of K. The trend clearly shows a decrease in accuracy at larger K, however this trend is slower with distance weighting.

	1	5	10	20	30	40	50	100
Spearman equal	98.6	97.5	96.4	94.8	93.2	92.1	90.7	80.5
Spearman weighted	98.6	98.2	97.5	96.2	95.2	95	94.3	93.2
City block equal	98.6	98.4	95.9	93.9	92.1	89.6	86.6	78.5
City block weighted	98.6	98.6	97.3	94.3	93.7	93	92.5	90.5

The results of this testing shows a clear trend of decreasing accuracy with the increase in the value of K. A value of K that is too low is likely to cause overfitting, thus a balance between accuracy and higher values of K must be found. Ultimately the value 10 was used because higher values of K with distance weighting produced a higher accuracy and 10 was deemed to be decrease the likelihood of overfitting.

4.3.4 Standardisation of Data

The absence of mean centred standardisation was interrogated. Mean centred standardisation has been used in tests until this point as this is the default for MATLAB. It was found that standardising data for most of the available distance metrics was in fact a detriment to their performance. The exception to this trend was the previously most accurate Spearman metric which was the least accurate with this change.

Table 4.7 effects of removing means standardisation on each distance metric.

	Set 1 Accuracy	Precision	Recall	F1-score	Set 2	Set 3
Euclidean	98.9	99.1	95.2	96.1	100	100
City block	98.6	98.5	97	97.5	100	100
Minkowski	98.9	99.1	95.2	96.1	100	100
Cosine	100	100	100	100	100	100
Spearman	89.8	93.2	85.4	86.2	100	91.4

An increase of accuracy occurred in the majority of KNN models when the data was not standardised. A possible explanation for this effect is the standardisation process makes all the regions of the dataset equally important. In the samples tested for the experimental datasets both the beginning and end of the collected wavelengths had few or no peaks. These regions, dominated by noise, reduce the accuracy when treated equally. Two possible solutions to resolve this loss of accuracy would be considering only the regions of the dataset containing peaks or not standardising the data. As this project seeks to produce the most generally applicable classification model, preselecting peaks would increase model specificity which is undesirable, standardisation of data was not used.

4.3.5 K-Nearest Neighbour Results

The Cosine distance metric proved to be the most accurate on the testing data. With the final settings, the Cosine model produced an accuracy of 100%. Euclidean distance was the second most effective model. It produced an accuracy on Dataset 1 of 98.9% though it had a somewhat poorer recall. The Minkowski distance KNN produced the same results as the Euclidian KNN however, it was significantly slower as it uses a cubic rather than a quadratic in its equations. The Minkowski model is less efficient in dataset classification as datasets grow.

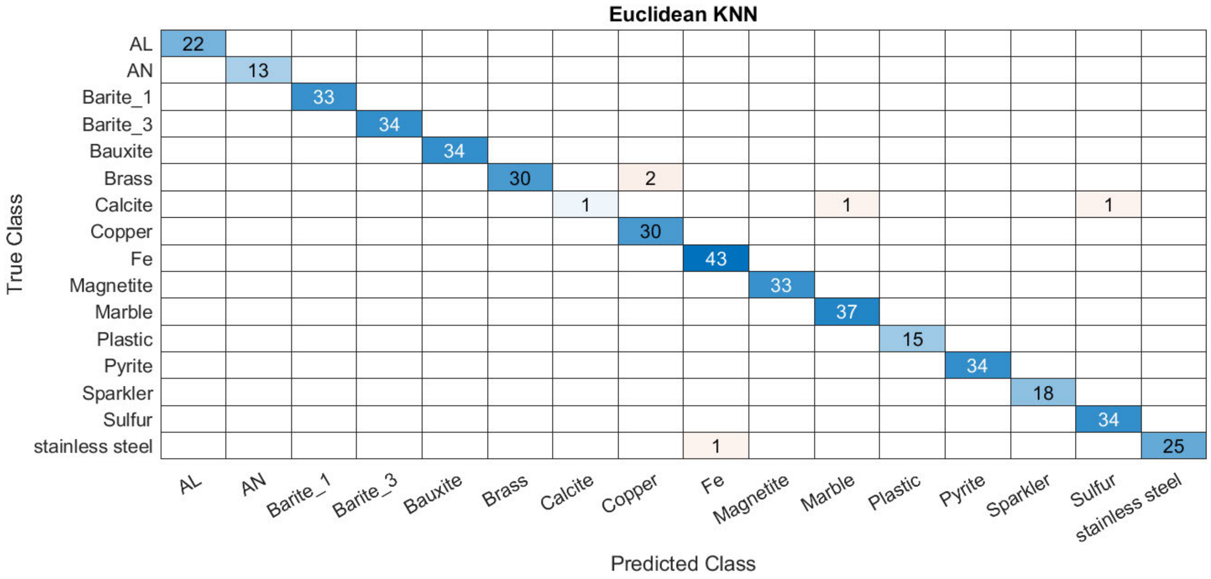
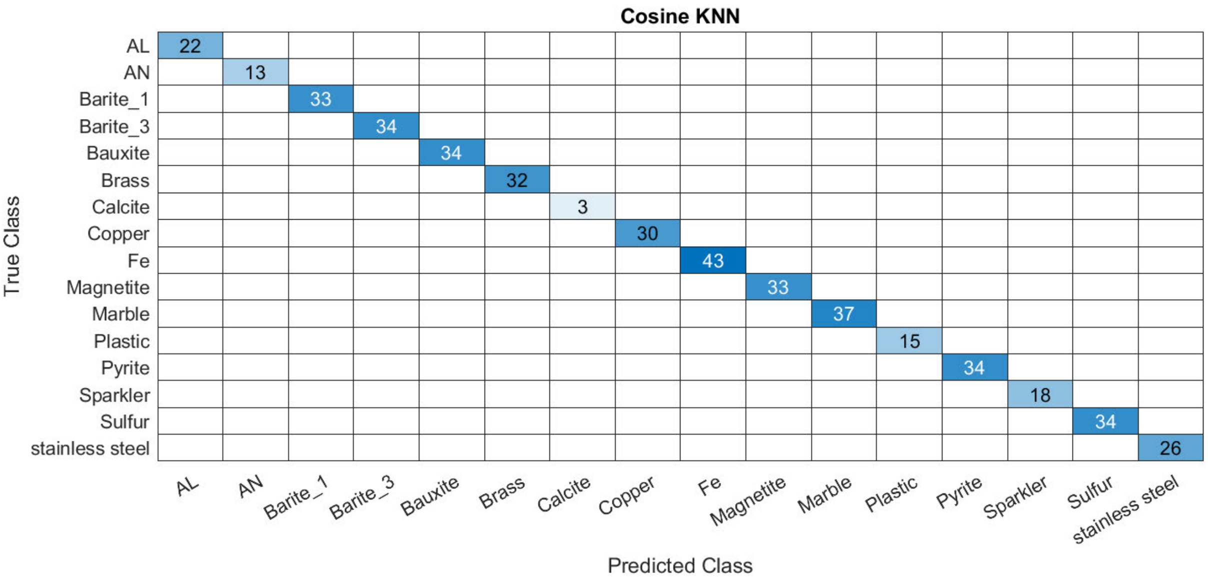


Figure 4.6 Confusion matrix of the final Euclidian model.

The Euclidian metric produced the confusion matrix shown in figure 4.6, with these final and most effective settings. It shows the noted difficulty with these systems to differentiate mixtures from their components. It also has the noted difficulty of determining the identity of calcite spectra. This is likely due to not having enough examples of calcite as many were removed due to low signal data.



Cosine KNN

True Class \ Predicted Class	AL	AN	Barite_1	Barite_3	Bauxite	Brass	Calcite	Copper	Fe	Magnetite	Marble	Plastic	Pyrite	Sparkler	Sulfur	stainless steel
AL	22															
AN		13														
Barite_1			33													
Barite_3				34												
Bauxite					34											
Brass						32										
Calcite							3									
Copper								30								
Fe									43							
Magnetite										33						
Marble											37					
Plastic												15				
Pyrite													34			
Sparkler														18		
Sulfur															34	
stainless steel																26

Figure 4.7 Confusion matrix of the final cosine model

The Cosine metric achieved 100% accuracy with these settings on Dataset 1. The main factor that reduced the accuracy previously appears to be the standardisation method. The Cosine metric with these settings also achieves 100% accuracy on the other two datasets. This indicates that the metric is the most applicable of the KNN models to this data. This metric may still fail to classify noisy data or fail faster from dataset reduction so both the Euclidian KNN and the Cosine KNN will be considered in tests on these properties.

4.4 Artificial Neural Networks

Artificial neural networks are initialised and altered by algorithms that produce pseudo-random changes within the network [55, 73]. While it is possible to set the initiation and system of changes to ensure the same results from training a network twice, this reduces the efficacy of the network. Set networks are only useful for exploring different sections of the training process rather than as a direct trained and useful network. The trained networks were also variable due to this randomisation. Thus, three trained networks will be reported for each category. The optimisation will also be limited as the initial settings were extremely effective.

4.4.1 Stochastic Graded Descent with Momentum

SGDM networks managed a high accuracy on Dataset 1. A three-run average of 98.7% accuracy and 98.7% F1-score were recorded for this dataset. Its largest failure rate was the calcite sample likely due to the low number of viable calcite spectra. In two of the three runs this model failed to identify a single spectrum of calcite. The other errors the model produced are primarily the normal confusion of mixtures and their components such as brass and copper. On Datasets 2 and 3 SGDM was able to achieve 100% accuracy with sufficient training time. This training time, as much as 200 epochs has a high chance of causing overfitting within these networks. The need for extended training times for the Datasets 2 and 3 is likely due to the smaller number of examples in these datasets.

Table 4.8 Average metrics for SGDM system on Dataset 1.

	Accuracy	Precision	Recall	F1-score
Run 1	98.6	98.75	99.4	99.1
Run 2	98.3	98.7	99.2	98.9
Run 3	99.3	99.5	97.4	98.2
Average	98.7	99.0	98.7	98.7

Table 4.9 accuracy of SGDM on datasets two and three with varied training time.

Epochs	Set2	Set3
20	14.3	57.1
50	28.6	88.6
100	62.9	92.9
200	100	100

4.4.2 Root Mean Squared Propagation

RMSProp networks proved the least accurate ANN system particularly on the smaller datasets where this system was unable to reach 100% accuracy. It was also prone to unusual errors such as classing AN as barite impurity and brass as plastic. The first of these errors, classifying AN as impure barite, was seen in each of the three runs the second only seen in a single run.

Table 4.10 Average metrics for RMSProp system on Dataset 1.

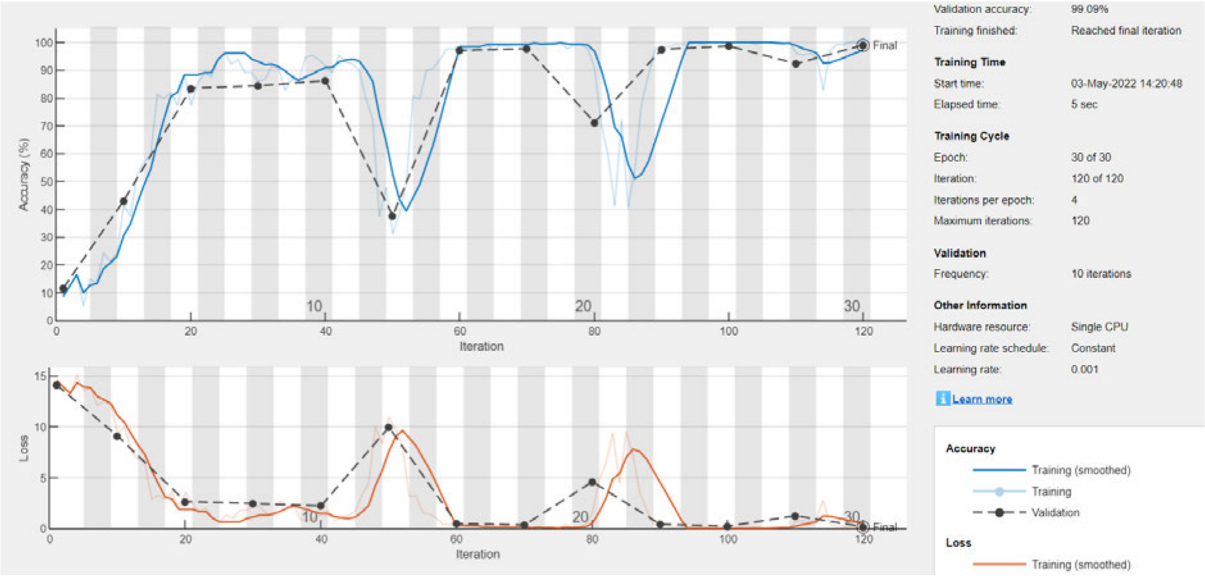
	Accuracy	Precision	Recall	F1-score
Run 1	97.3	97.7	97.3	93.8
Run 2	98.6	98.9	97.5	97.9
Run 3	97.6	97.6	92.9	94.0
Average	97.8	98.1	95.9	95.2

Table 4.11 Accuracy of RMSProp on Datasets 2 and 3 with varied training time.

Epochs	Set2	Set3
20	11.4	22.9
50	11.4	50
100	17.1	57.1
200	40	98.6

RMSProp models tended to have perfect recall on Dataset 3. The model incorrectly classified all errors into the same class ie all nylon spectra were classified as xylitol rather than a varied misclassification. An accuracy of 14% is the expected accuracy of a random guess in Dataset 2 thus the RMSProp Dataset 2 training did not become more accurate than a random guess until the 100-epoch training model. This shows the optimiser had great difficulty training a network to classify that dataset.

The other notable quirk of RMSProp models is within their training. These models, once training is nearly complete have a large downward spike in accuracy which then quickly rises back to near 100%.



Validation accuracy: 99.09%
Training finished: Reached final iteration

Training Time
Start time: 03-May-2022 14:20:48
Elapsed time: 5 sec

Training Cycle
Epoch: 30 of 30
Iteration: 120 of 120
Iterations per epoch: 4
Maximum iterations: 120

Validation
Frequency: 10 iterations

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.001

Learn more

Accuracy

Loss

Figure 4.8 An example of the strange behaviour of RMSProp.

Epochs 12 and 21 show the point at which the network's accuracy spikes downward. This seems to be a result of the anti-zero biasing of RMSProp and is caused by the RMS term falling too close to 0. This leads to the denominator being dominated by the small ε term causing larger changes to the biases resulting in changed classifications. This is consistent with Li Q *et al [92]* finding similar variability when using RMSprop classifiers.

4.4.3 *Adam*

The *Adam* algorithm was the only system to produce a model with perfect classification accuracy. Despite this its average values aside from accuracy are still below that achieved by SGDM. This average indicates that while this system was more effective at identifying the majority of compounds, it was incorrectly classified a wider range of samples than SGDM. *Adam* was also less able to classify small datasets than SGDM and other non-ANN models. It was unable to fully classify Dataset 2 even with 200 epochs of training. This model also misclassified AN as Barite impurity but proved abnormally effective at classifying calcite.

Table 4.12 Average metrics for *Adam* system on Dataset 1.

	Accuracy	Precision	Recall	F1-score
Run 1	98.0	98.5	96.8	97.5
Run 2	98.6	96.0	95.0	95.4
Run 3	100	100	100	100
Average	98.9	98.2	97.3	97.6

Table 4.13 Accuracy of *Adam* on Datasets 2 and 3 with varied training time.

Epochs	Set2	Set3
20	14.3	24.3
50	17.1	87.1
100	57.1	95.7
200	71.4	100

4.4.4 Best and Worst Artificial Neural Networks

SDGM proved the second most accurate system but had the highest rating in the other metrics. The majority of the errors made by the models trained with this algorithm were failing to identify the minimal number of calcite spectra and misclassification of mixtures. It was also most able to classify the smaller datasets after the maximum tested training time.

The *Adam* trained model proved to be the most accurate on average but was lower in the other metrics. *Adam* models also proved less accurate at detecting AN. The Adam models were also less able to classify smaller datasets than SGDM.

RMSProp trained models proved the least effective overall. Each run misclassified some AN as Barite impurity. The average accuracy and the other metrics of this system were the lowest of the three ANN algorithms. This algorithm also proved the least able to train on smaller datasets being unable to achieve 100% on either of the smaller datasets within 200 epochs.

While SGDM proved not to be the most accurate system, it did produce less false negatives in general and specifically had no errors, false positive or negative, on AN. This algorithm also proved most able to classify smaller datasets and thus at this point in the investigation is the most promising ANN algorithm.

4.5 Comparative Analysis

Each of the selected algorithms were able to produce accuracies of over 98% on Dataset 1. The ANN based approaches struggled on the smaller datasets however, they were still able to classify them if trained for long enough. Extended training time, such as is required by ANNs on smaller datasets can lead to overfitting and reduced accuracy. As each algorithm is able to produce a similar level of accuracy no clear recommendation can yet be made. Additional data conditions were tested to attempt to determine which system is best for the intended application.

4.5.1 Dataset Size Reduction with Relieff - Model Performance

Dataset size can be a concern for processing times for some of the algorithms in addition to the memory requirements to run the models. Dataset 1 was reduced using a Relieff algorithm [73] to determine the most and least important data points for class identification. Only the x most important points were kept and used in this test. The values of x tested were 1000, 500, 100, 50, and 10 data points.

Table 4.14 The accuracy, precision, recall and F1-score of each system on progressively smaller datasets.

	Accuracy	Precision	Recall	F1-score
LDA 1000	97.1	95.9	96.3	96
LDA 500	98.4	97.5	98.1	97.7
LDA 100	97.7	96.9	98	97.3
LDA 50	94.3	93.5	94.3	93.6
LDA 10	68	62.5	62.4	62
KNN cosine 1000	100	100	100	100
KNN cosine 500	99.6	99.6	99.6	99.6
KNN cosine 100	98.2	98.4	98.2	97.9
KNN cosine 50	94.6	95.4	88.8	90.3
KNN cosine 10	64.6	67.2	58.2	58.8
KNN Euclidean 1000	98.2	98.7	96.5	97.2
KNN Euclidean 500	98.2	98.4	96.5	97.1
KNN Euclidean 100	95	94.6	93.5	93.7
KNN Euclidean 50	91.6	90.1	90.3	90.2
KNN Euclidean 10	69.4	69.8	68.6	68
SGDM 1000	98	97.6	94	94.7
SGDM 500	90.8	88.3	81.7	81.4
SGDM 100	91.8	86.7	88.6	87.1
SGDM 50	68	63.5	73.8	66.8
SGDM 10	15	7.7	33.3	12.1
RMSProp 1000	95.2	96.4	92.8	93.9
RMSProp 500	86.4	90.6	84.4	81.3
RMSProp 100	43.9	55.4	46.8	45.7
RMSProp 50	27.2	18	28.9	20.7
RMSProp 10	11.9	7.3	25.4	10.7
Adam 1000	93.2	92.1	93.4	92.5
Adam 500	91.2	86.6	83.1	82.6
Adam 100	52.4	54.4	49.9	48.1
Adam 50	26.5	25.4	24.6	24.5
Adam 10	11.2	12.5	13.7	7.39

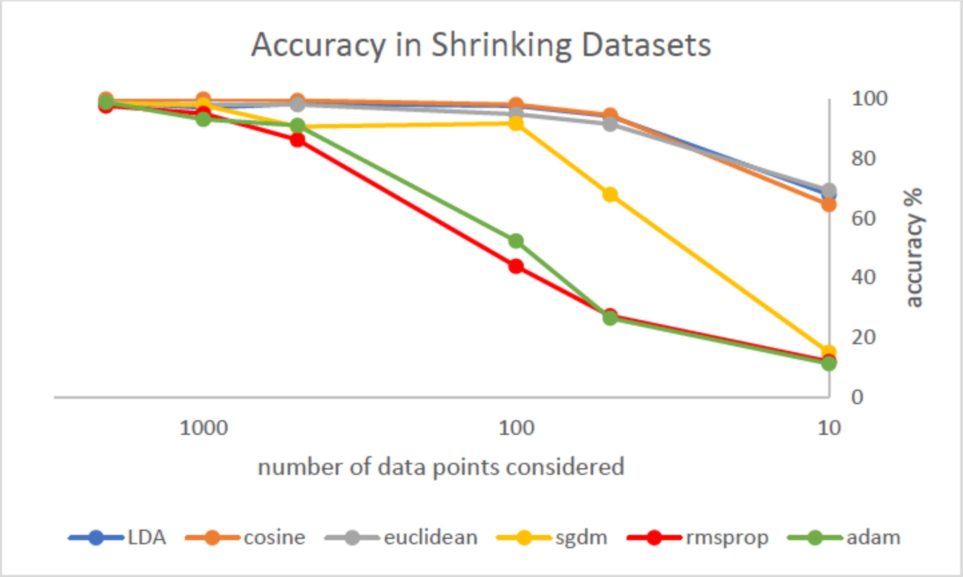


Figure 4.9 Comparing the accuracy of each algorithm with progressively fewer datapoints.

The table 4.14 and figure 4.9 show the effect of the dataset reduction on the accuracy of each method. All systems became less accurate as the dataset shrank, however some proved significantly more resistant to the reduced data than others. The ANNs proved the least able to work with reduced data as is expected for neural networks. The ANNs were able to gain better accuracies with longer training times, for instance *Adam* 100 with 100 epochs to train rather than 30 produced an accuracy of 85.7% and an F1-score of 83.6%. Even when allowed to train for 500 epochs ANN methods could not match the accuracy of the other algorithms. All algorithms showed a marked reduction from 50 data points to 10, as Relieff selects the most important of existing data points without increasing the amount of information they contain this is unsurprising.

4.5.2 Dataset Size Reduction with PCA - Model Performance

An alternate method of dataset reduction is to make use of Principal Component Analysis. The n most impactful PCs were considered for each trial with values of n=4, 8, 16, 32, and 64. These values of n were chosen on the lower end on the basis of explained variance and on the higher end as set values to ensure starting accuracy for the models. 16 was chosen as the centre as this was the number of classes in Dataset 1. Two methods of data reduction were used and compared to ensure that neither method favoured a particular classifier.

Table 4.15 Effects of PCA dataset shrinkage on each algorithm

	Accuracy	Precision	Recall	F1-score
LDA 64	99.3	99.1	99.4	99.2
LDA 32	98.4	97.9	96.1	96.5
LDA 16	93.0	94.9	88.5	90.0
LDA 8	84.1	88.9	85.2	85.3
LDA 4	73.0	77.4	76.3	74.4
Euclidian KNN 64	99.3	97.3	99.1	98.1
Euclidian KNN 32	99.1	99.3	97.3	98.1
Euclidian KNN 16	98.9	99.1	95.2	96.1
Euclidian KNN 8	98.4	98.5	96.6	97.3
Euclidian KNN 4	94.1	94.3	92.4	92.9
Cosine KNN 64	99.1	99.3	97.3	98.1
Cosine KNN 32	99.1	99.3	97.3	98.1
Cosine KNN 16	99.1	99.3	97.3	98.1
Cosine KNN 8	98.6	98.9	96.9	97.6
Cosine KNN 4	93.7	92.0	91.6	91.5
SGDM 64	97.7	97.1	98.3	97.4
SGDM 32	97.8	98.0	98.2	97.9
SGDM 16	94.6	94.5	95.4	94.0
SGDM 8	87.5	93.1	89.5	89.0
SDGM 4	69.4	85.4	85.8	80.3
RMSProp 64	86.2	85.4	79.7	80.2
RMSProp 32	61.7	51.5	61.6	54.7
RMSProp 16	46.3	47.0	55.2	47.1
RMSProp 8	22.9	14.7	21.8	16.5
RMSProp 4	16.6	15.1	18.3	16.6
Adam 64	83.2	76.8	82.4	78.8
Adam 32	50.3	58.0	54.9	54.7
Adam 16	50.0	49.7	57.6	52.7
Adam 8	13.4	12.5	13.4	13.7
Adam 4	14.5	18.9	16.0	13.0

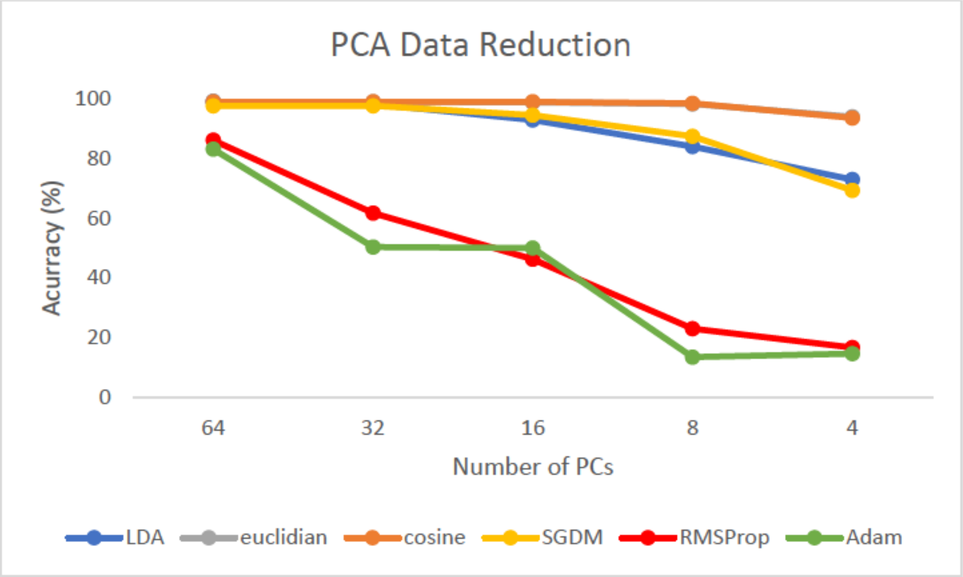


Figure 4.10 Comparison of the accuracies of each algorithm with fewer and fewer PCs.

Data reduction by PCA showed similar trends when shrinking the dataset as the Relieff algorithm method. The KNN algorithms continue to be the most stable and high performing. In this instance the two KNNs are almost identical Euclidian performing only slightly better. The LDA and SGDM both performed less accurately than the KNN algorithms, with the 4PC models performing at approximately 80% accuracy. This level of accuracy is an improvement for the SGDM 4PC model when compared to the Relieff 10 SGDM model. The LDA 4PC model is significantly less accurate than the 4PC KNN models, a decrease in relative accuracy when compared to the Relieff 10 models. Both RMSProp and *Adam* models performed weakly as with the other method of dataset reduction falling quickly to below 20% accuracy. This is likely due in part to training times which were kept consistent, but they also did not maintain accuracy with SGDM. The other notable anomaly is the 16PC model of the RMSProp algorithm. This model outperformed the 32PC model for accuracy, this highlights a common factor with ANNs, each time an algorithm trains a model a different model is produced. Different ANNs produced by the same algorithm with the same settings can be significantly better or worse than each other.

4.5.3 Effect of Noise on Model Performance

Another likely area of concern in the application this program is being recommended for is noise. The data is to be collected without sample preparation and at standoff range under uncontrolled conditions. Each of these conditions will affect the signal-to-noise ratio present in the spectra to be classified. The lack of sample preparation is likely to reduce the strength and clarity of the LIBS signal. This lack of strength is due to uneven surfaces reducing laser ablation and local inhomogeneity altering the local spectrum. Standoff ranges reduce the intensity of the signal due to air scattering resulting in further decreases of the signal to noise ratio. Other factors such as optics will also affect the ablation and subsequent signal to noise ratio.

Robustness to noise in the dataset is an important characteristic to differentiate the usefulness of the algorithms. Models were generated with the same training data and tested on the same dataset six times with varying degrees of random noise added to it. The noise was set in ranges of ±, 1000, 3000, 6000, 9000 or 12,000. These values were chosen as the maximum value of the data was approximately 60,000 thus values of noise between 1.67% and 20% of the total signal were chosen.

Table 4.16 Effects of increasing noise on the effectiveness of each system.

	Accuracy	Precision	Recall	F1-score
LDA1000	41.5	35.5	35.6	35.1
LDA3000	18.8	16.3	17.5	13.8
LDA6000	13.8	9.7	12.0	9.2
LDA9000	9.8	8.1	10.1	7.4
LDA12000	11.1	8.5	9.3	6.8
cosineKNN1000	99.6	99.7	95.8	96.7
cosineKNN3000	98.2	98.6	93.9	95.1
cosineKNN6000	96.8	96.7	96.6	96.6
cosineKNN9000	92.74	83.6	90.9	91.1
cosineKNN12000	88.0	86.5	86.1	84.8
EuclideanKNN1000	96.8	96.6	97.2	96.5
EuclideanKNN3000	97.3	96.8	97.7	97
EuclideanKNN6000	95.7	95.1	96.3	95.3
EuclideanKNN9000	95.7	96.0	96.1	95.8
EuclideanKNN12000	93.2	92.0	92.6	92.1
SGDM1000	95.9	96.8	94.9	95.3
SGDM3000	92.3	94.6	90.3	90.8
SGDM6000	85	80.5	82.2	79.5
SGDM9000	82.3	84.9	84.4	81.9
SGDM12000	77.8	80.2	79.7	75.7
RMSProp1000	86.4	84.7	83.3	81.8
RMSProp3000	66.4	63.1	61.8	59.3
RMSProp6000	51.7	44.4	48.2	44.2
RMSProp9000	41.7	34.2	39.6	34.6
RMSProp12000	41.3	35.5	39.5	35.1
Adam 1000	85.3	84.5	83.1	79.5
Adam 3000	65.8	59.7	60.9	58.1
Adam 6000	53.1	47.6	48.3	45.8
Adam 9000	42.4	37.7	38.5	36.7
Adam 12000	42.9	35.8	41.9	36.3

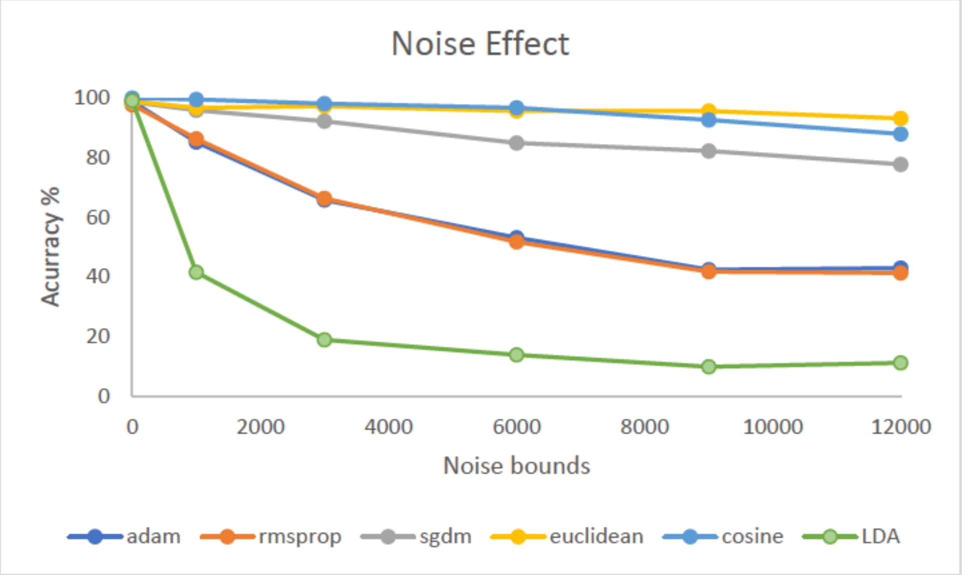


Figure 4.11 Comparison of each algorithm's accuracy with progressively nosier testing data.

As in the other comparative tests, ANNs were more affected by the change than most of the other algorithms. In this instance the most effected system was LDA which lost accuracy extremely quickly down to the point of near guessing, 14% accuracy. While still decreasing the SGDM trained models only showed a moderate drop in accuracy. This result indicates the LDA and two of the three trained ANN models are too specific to their training data. The SGDM algorithm, while still showing signs of specificity to the training data, was far more robust and able to adapt to the random variations. Between the KNN algorithms, the Euclidean algorithm proved slightly more resistant to noise than the Cosine algorithm. The Euclidian 12,000 model maintained an accuracy of about 90% while the Cosine 12,000 model accuracy fell below 90%. The degree of error caused by this test may be indicative of overfitting taking place in the LDA, RMSProp and *Adam* models.

4.5.4 Unexpected Data

In this test models of each type were trained without the barite impurity sample. These models were then tested with barite impurity spectra in testing data to analyse how they would classify data not of one of the known classes. ANNs were, as before, run three times. The result of each instance is reported.

Table 4.17 The classifications of the barite impurity with each different model.

	Classification	Secondary Classification
LDA	AN	Fourteen sulfur
Cosine	Iron	
Euclidean	Stainless steel	One barite
SGDM 1	Sulfur	Two iron
SGDM 2	AN	one sulfur
SGDM 3	Iron	
RMSProp 1	Iron	
RMSProp 2	Sulfur	
RMSProp 3	Iron	
Adam 1	Brass	
Adam 2	AN	
Adam 3	AN	

The three ANNs assigned the sample differently from different randomised starting points rather than reaching the same conclusion each time. This highlights a weakness of ANNs, as each model trained is unique any update or change to a model can lead to different and unexpected behaviour. This issue was more prevalent in SGDM models than those of the other ANN algorithms. Beyond this problem *Adam* produced the strangest result of uniquely labelling the unknown sample as brass and RMSProp had the least concerning classification of iron. This test again affirms the strange and undesirable behaviours of LDA models in less-than-ideal conditions. Showing not only an undesirable AN classification but also splitting the spectra nearly in half (14/34) with a classification of sulfur. The Euclidian KNN model classified the spectra primarily as stainless steel with one spectrum classed as barite. This barite classification may be chance, or the model may have recognised a spectrum of the impurity in which barite was visible. Finally, the cosine model simply classified all unknown spectra as iron.

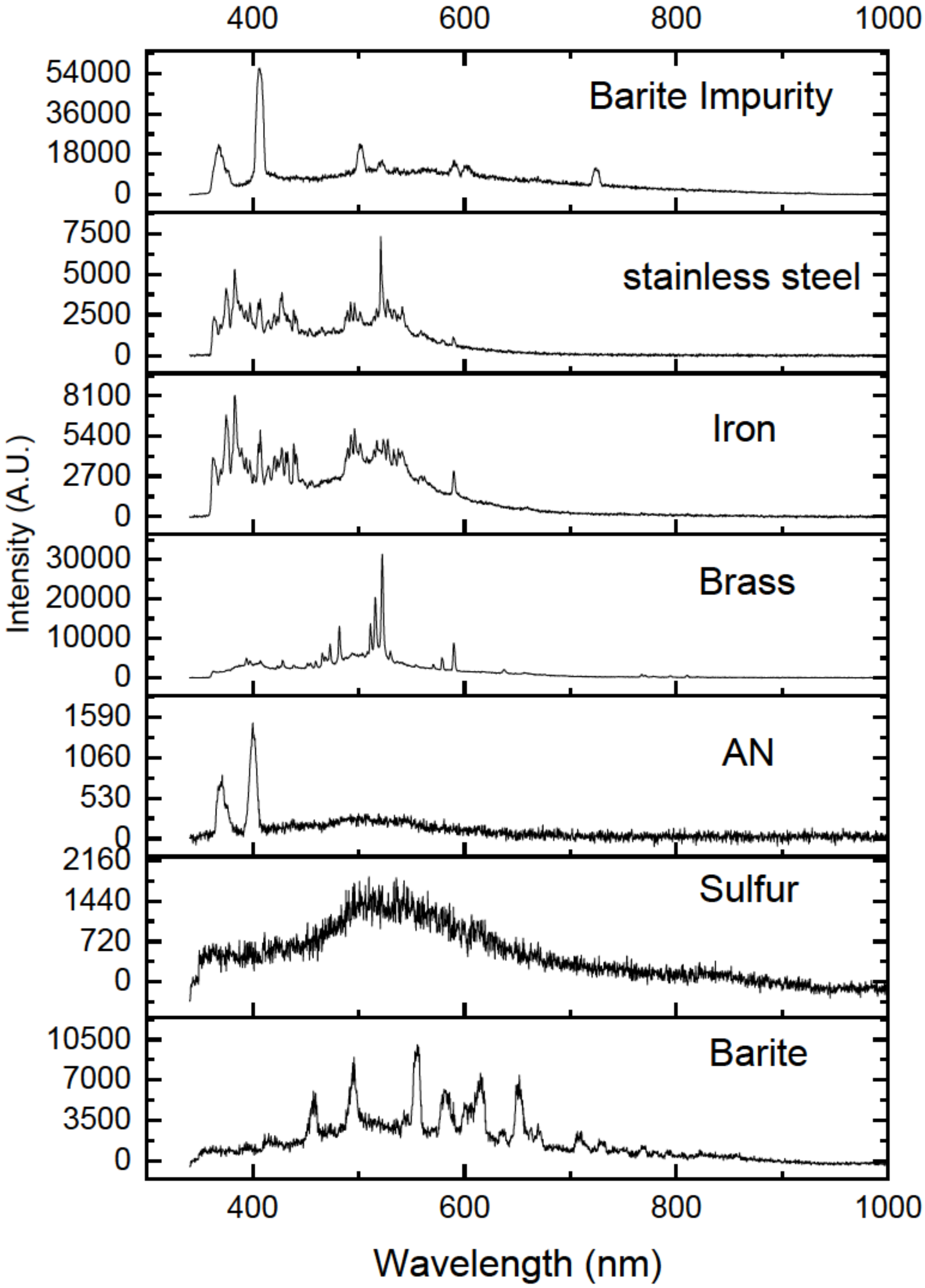


Figure 4.12 Stacked plot of Barite impurity and the spectra of the samples the models assigned it to.

Figure 4.12 grants some insight into why these classifications might have been chosen by the models. The clearest is that of AN which while much less intense has a similar pair of initial peaks around 370nm and 410nm and a slightly humped structure. Stainless steel and iron appear to be relating less to any individual peak and more the high intensity of the impurity spectra. This may also explain the *Adam* classifier's brass result as brass spectra are high intensity though they do have an extremely intense peak with no similar peak in the impurity spectrum. Finally, it would appear the sulfur classifications may be based on overall shape without regard to peaks or the models may simply be using sulfur as an “other” category.

4.5.5 Summary

These comparative tests show that the cosine KNN, while the most accurate under ideal situations and still relatively robust, falls behind the Euclidian KNN model when the conditions are worsened, particularly in high noise situations. LDA was shown to be unable to accurately classify noisy data with accuracy decreasing greatly even with the lowest tested noise level. LDA maintained accuracy almost as well as the KNN models under the effects of dataset reduction, however it proved ill-suited for classing samples of no known class. The ANN models all did poorly with both dataset reduction and noise. The SGDM model was able to handle noise and data reduction better than the other two ANN models, it did however, show the most variability of unknown classification. The ANNs difficulty with dataset reduction was in part due to requiring more training, tests allowing the ANNs to train longer did show an improvement however, even in such tests, accuracy was still lower than with other methods.

Chapter 5 – **Combining LIBS and RS – Exploring Methods for Data Fusion**

5.1 Introduction to Data Fusion

Data fusion combines multiple different sources of data to achieve improved accuracy and more specific inferences than could be achieved by a single sensor alone [93]. There are several "levels" and approaches to data fusion. The levels, called high-level, mid-level and low-level fusion, are split based on the degree of data processing that occurred before the fusion of the two data types. Low-level fusion makes use of minimal data pre-processing and combines data before training. Mid-level fusion makes use of more extensive pre-processing such as dimensionality reduction or feature selection but still occurs before model creation. High-level fusion refers to fusion beyond the level of model creation. This can include decision trees using multiple models or simple fusion where two models are trained. Decision trees using multiple models determine which model the data should be sent to in order to be classified. In the case of simple fusion, the predictions of the two models are combined to produce a result [94].

Herein one approach will be used for high and mid-level fusion while two low-level fusion methods will be investigated. The high-level approach tested is a simpler high-level fusion technique of training a model for each data type and combining the predictions of both to produce the final classification. The tested mid-level approach makes use of PCA as a dimensionality reduction technique. The two low-level approaches are the addition of the LIBS and RS spectra into a single spectrum and the concatenation of the two data types into a single dataset.

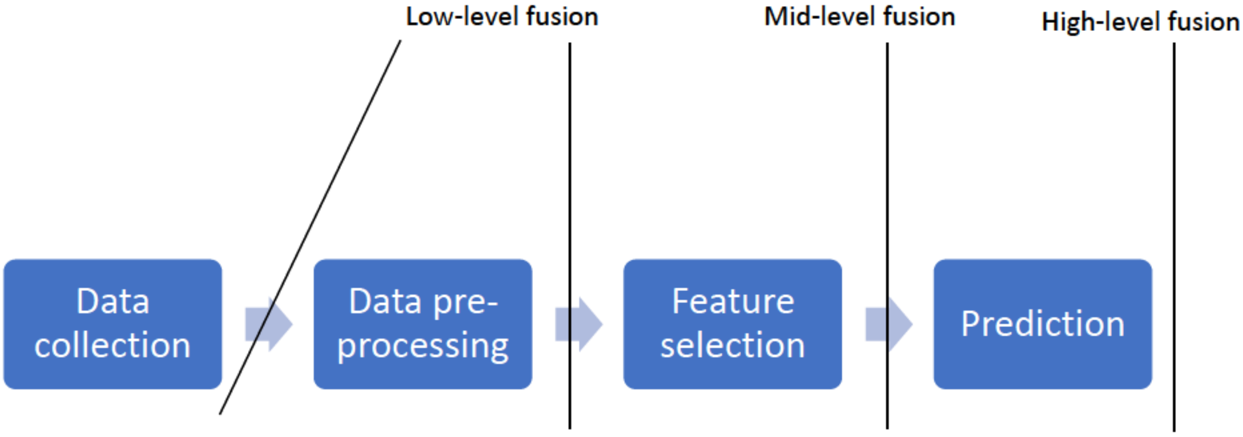


Figure 5.1 Diagram showing the fusion point of each level of data fusion; low-level fusion any time after collection and before feature selection, mid-level after feature selection and before prediction and high level after prediction.

A separate dataset comprising individual LIBS and RS spectra for each sample was used in this chapter in order to develop an accurate fusion best practice. The prior datasets were incompatible with a fusion investigation, samples having LIBS or RS data but not both separately.

5.2 Data Collection

This dataset was collected using a Continuum Surelite III Laser (operating conditions: 355 nm, 5 ns pulse width, 10 Hz repetition rate) and a PIMax4 spectrometer (1024 channels) with similar optics as the other datasets. The spectrometer was calibrated for RS for a Nd:YAG laser (355nm) against a mercury (Hg) lamp. The spectrometer was calibrated to the Hg(I) emission lines which appear at 365.0nm, 404.7nm, 435.8nm, 546.1nm and 578.0nm. The data was taken from the spectrometer into the LightField program and exported as .spe files for import into excel. In excel, the data was smoothed via averaging ten shots into a single spectrum then imported into Origin. Data was examined in Origin and finally imported into MATLAB for use.

5.3 Sample Preparation

Sample preparation was again undertaken by Dr Ula Alexander. The samples for this dataset were mounted on an aluminium backing plate with carbon tape. This dataset consisted of metal, mineral, crystalline powder, and organic sample types. The identities of each sample type are listed in table 5.1 and the number of scans for each sample type are listed in figure 5.2. Metal and mineral samples were used natively while crystalline samples were pressed in the same manner as the other datasets (procedure found in section 3.2.1). These samples were pressed into two-millimetre-thick discs with a hydraulic press with 300 bar of pressure and mounted on the backing with carbon tape.

Table 5.1 The proposed molecular formula of each sample and the source of each sample

Sample	Molecular formula	Source
Aluminium (Al)	Al	Flinders University workshop
Ammonium Nitrate (AN)	NH_4NO_3	DSTG
Barite	$BaSO_4$	Flinders University rock collection
Copper	Cu	Flinders University workshop
Iron	Fe	Flinders University workshop
Marble	$CaCO_3$	Flinders University rock collection
Potassium chlorate	$KClO_3$	Sigma Aldrich
Potassium perchlorate	$KClO_4$	Sigma Aldrich
Potassium nitrate	KNO_3	Sigma Aldrich

5.4 Dataset Exploration

The fusion dataset consists of an equal number of LIBS and RS spectra of 9 sample classes with a distribution after pre-processing shown in figure 5.2 for a total of 1005 LIBS and RS spectra. Example plots of each sample's LIBS and RS spectra are shown in figure 5.3.

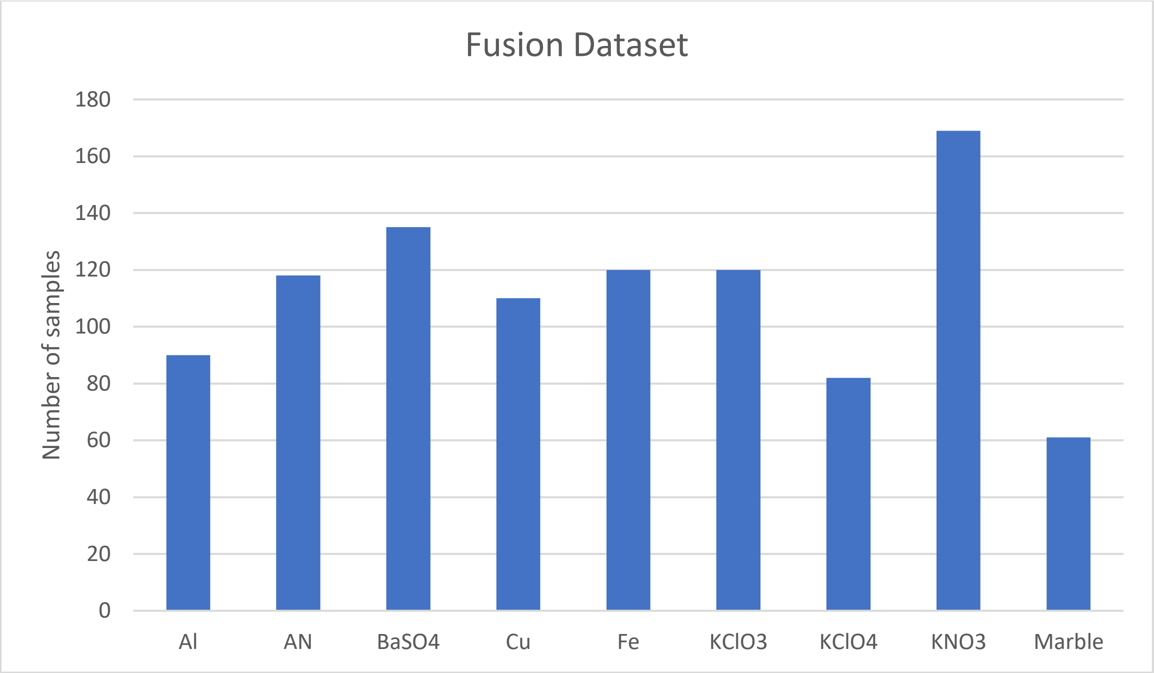


Figure 5.2, Distribution of spectra from sample types in the fusion dataset.

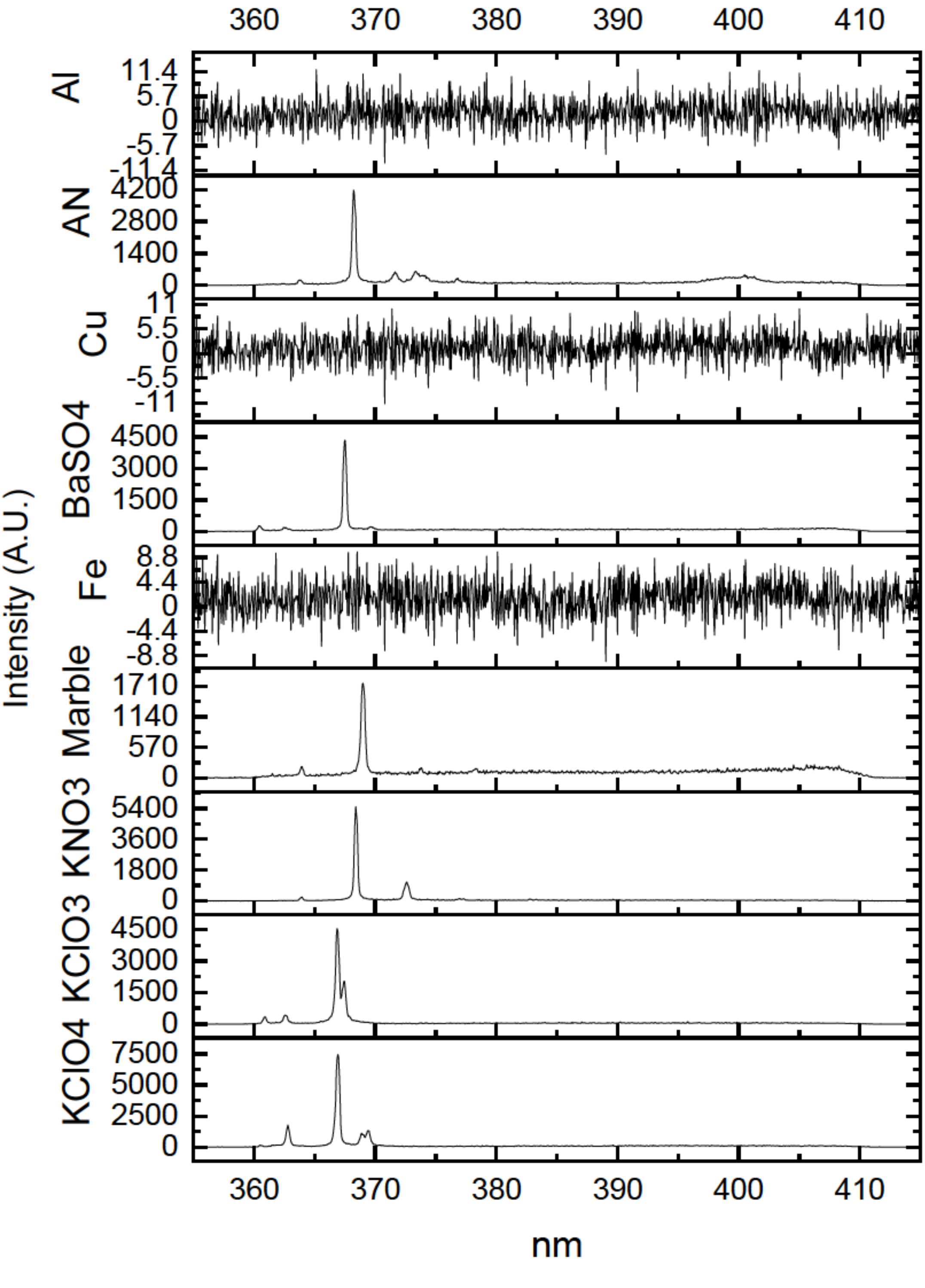


Figure 5.3 Example RS spectra of each sample type in the fusion dataset.

Table 5.2 Main and notable secondary peak in RS spectra.

Sample	Key Peak	Secondary Peak
Aluminium	None	None
Ammonium nitrate [95]	368 NO_3(v1)	372nm NO_3
Barite[96]	367.5nm SO_4 v1	360nm SO_4 v2
Copper	None	None
Iron	None	None
Marble	369nm CO_3	363.9nm CO_3
Potassium Chlorate	367.5nm ClO_3	366.9nm ClO_3
Potassium Perchlorate [97]	367nm ClO_4 v1	362nm ClO_4 V4
Potassium Nitrate [97]	368.4nm NO_3 (v1)	372.6nm NO_3

Three of the sample types in this dataset (aluminium, copper and iron) are not RS active, producing only a small amount of noise in place of a RS spectrum. The remaining spectra all have a single large peak at a similar positions varying from 367-369nm though in RS this is a significant range of variance. The primary peak of the barium, chlorate and perchlorate spectra were all extremely close, however all have small secondary peaks that differentiate them from the others. The potassium nitrate and AN samples also had primary and secondary peaks at very similar positions however a third peak further into the AN spectra allows the two to be differentiated.

The RS spectra in this dataset are all equally shifted approximately a quarter nm (20 wavenumbers) lower than literature values.

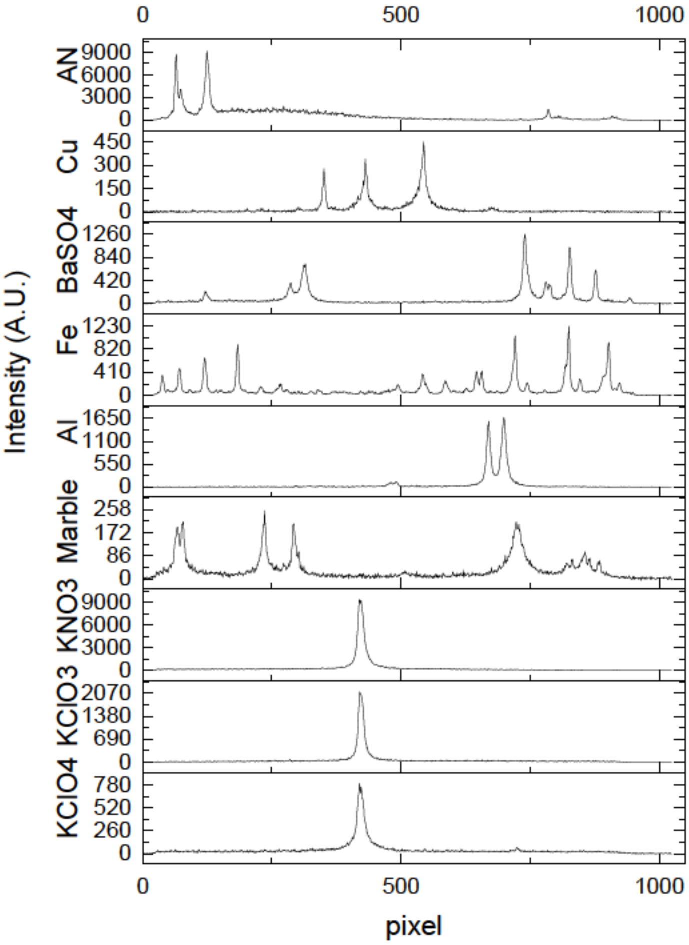


Figure 5.4 Example LIBS spectra of each sample in the fusion dataset. The spectra here are ordered by detector pixel as this is how they will be used in the models. Wavelength ordered data can be found in appendix 2.

Table 5.3 Key peak assignments for LIBS samples [39].

Sample	Key peak 1	Key peak 2	Key peak 3	Key peak 4
Aluminium	394nm/p669 Al(I)	396nm/p697 Al(I)		
Ammonium nitrate	AN Raman			
Barite	578nm/p780 Ba(I)	582nm/p827 Ba(I)	585nm/p876 Ba(II)	553nm/p310 Ba(I)
Copper	510nm/p351 Cu(I)	515nm/p431 Cu(I)	521nm/p544 Cu(I)	
Iron	532nm/p825 Fe(I)	527nm/p721 Fe(I)	537nm/p903 Fe(I)	495nm/p184 Fe(I)
Marble	393nm/p236 Ca(II)	396nm/p293 Ca(II)	383nm/p79 mg(I) doublet	422nm/p728 Ca(I)
Potassium Chlorate	404nm/p421 K(I)			
Potassium	404nm/p421 K(I)			

Perchlorate				
Potassium Nitrate	404nm/p421 K(I)			

LIBS spectra are reported with both wavelength and pixel number in table 5.3 and compared by pixel number as this is how the classifiers consider the spectra. LIBS spectra were taken with different centre wavelengths to ensure spectra with notable peaks were gained from each sample with a narrow grating. Considering these spectra in this fashion may make classification easier for the models. It is not a direct comparison to the intended application but should allow for an examination of the most effective method of data fusion.

The most notable sample of this dataset is the attempted LIBS spectrum of AN. No component of AN is highly LIBS active, instead this sample shows an AN RS spectra [39]. Other light samples such as sugars and organic energetics would likely produce similarly lacking responses. This is another example of the need for the use of both systems in tandem to identify energetics accurately.

The metal samples are the opposite of the AN, highly LIBS active, producing clear spectral peaks. The copper spectrum has three main peaks in a region otherwise only occupied by the potassium peaks. The iron spectrum shared similarities to barium sulphate in the second half of the spectrum having three similarly placed main peaks. The three potassium-based samples look nearly identical each one having a single main peak at approximately pixel 420. The only notable difference between the potassium-based samples is in intensity, KNO_3 being more intense than $KClO_3$ and $KClO_3$ being more intense than $KClO_4$.

5.5 Processing and Training

The dataset for fusion testing was produced and processed separately from the other datasets. Ten shot averaged data was examined to ensure no data with notable abnormalities such as the backing material (aluminium) showing through. Any such data was removed from the training and testing datasets and retained for use as unusual data. The data was then filtered to remove spectra with low signal intensities from the LIBS dataset (maximum signal lower than 200) with corresponding RS spectra removed from the RS dataset. This processed data was then used for testing the fusion methods.

All data fusion tests were performed using a Euclidean KNN with a K value of 10 and inverse distance weighting (code in appendix 1 MATLAB). All models were able to determine the identity of every spectrum in the validation dataset successfully. The degree of confusion within the models will be considered in addition to the computational load to determine the most efficient and robust model for this investigation.

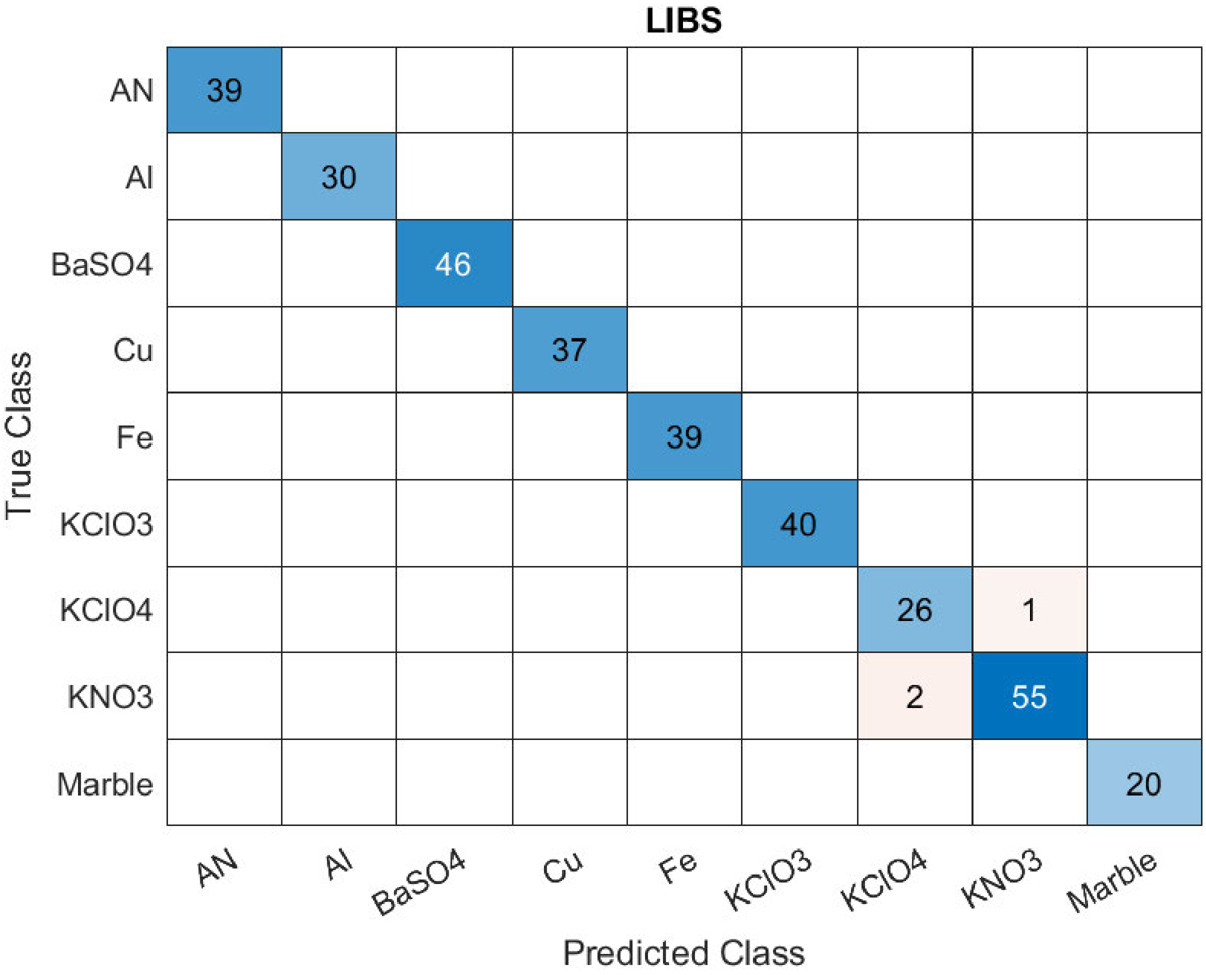
5.6 Assessing Fusion Methods

Classification in this dataset proved to be simple for all fusion models. Each one produced correct labels for every spectrum in the verification dataset. A new metric is thus required for comparison between models. To this end the score values from the classifications were compared allowing for the comparison of how certain the labels were. Score values are the normalised (to total 1) totals of the votes for an unknown spectrum to each class in a dataset. For instance, in a KNN without distance weighting and a k value of 5, if three of the nearest five samples were sample “a” one was “b” and one was “d” then the scores of this sample would be a 0.6 b 0.2 and d 0.2 (total of 1). These score values would be interpreted as a 60% certainty the sample was class a with a 20% chance it was instead class b or d.

5.7 Results

No Fusion

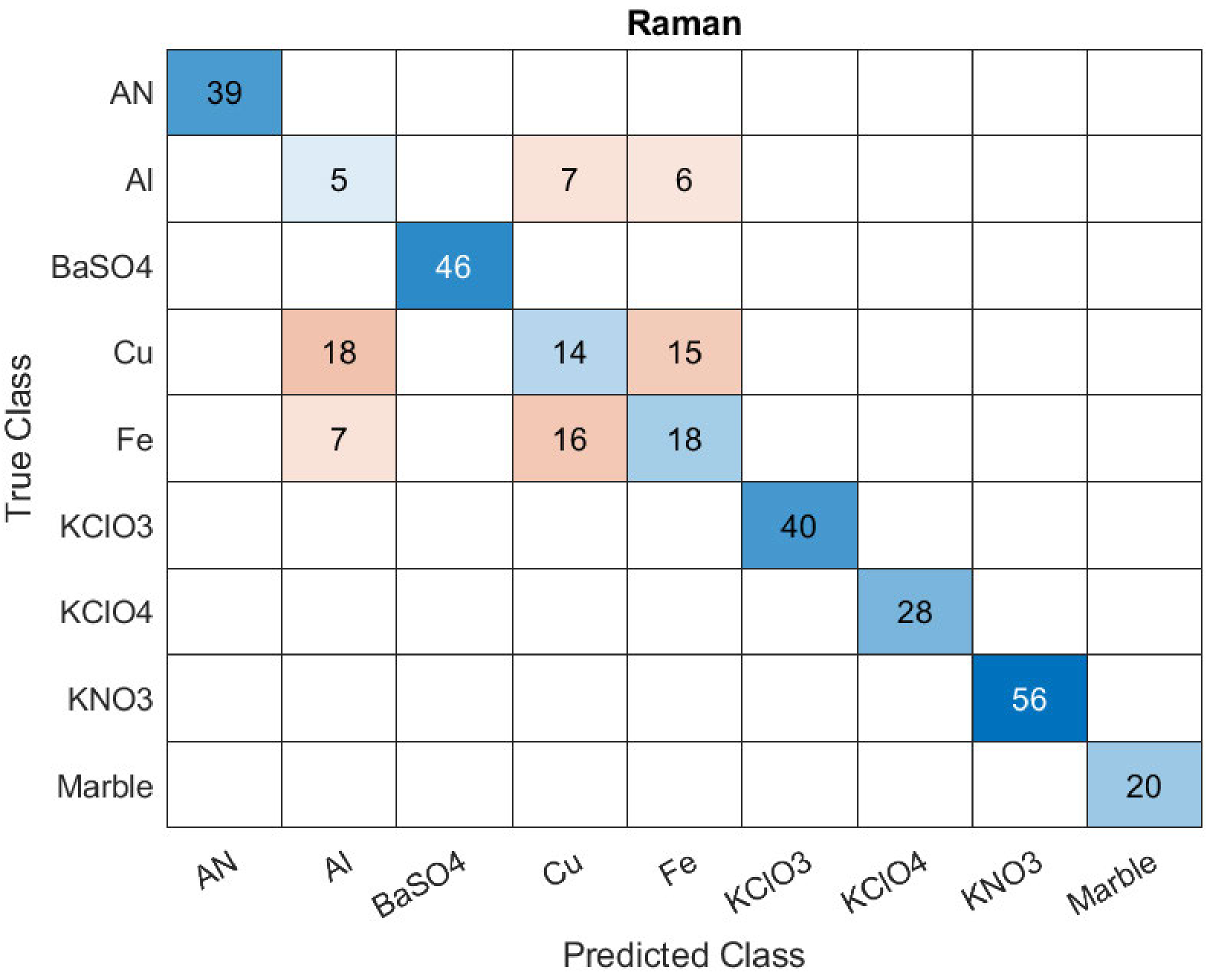
For comparison LIBS and RS were run on the dataset separately. LIBS on this dataset struggled to properly identify the difference between $KClO_4$ and KNO_3 leading to some misclassification and a large amount of confusion. It is able to differentiate $KClO_3$ from KNO_3 and $KClO_4$. Other than this confusion with the potassium samples the LIBS KNN is able to fully classify the rest of the dataset as shown in figure 5.5.



	AN	Al	BaSO4	Cu	Fe	KClO3	KClO4	KNO3	Marble
AN	39								
Al		30							
BaSO4			46						
Cu				37					
Fe					39				
KClO3						40			
KClO4							26	1	
KNO3							2	55	
Marble									20

Figure 5.5 Confusion matrix of the LIBS model of the fusion dataset.

A RS based attempt to classify this system was unable to differentiate the three RS inactive metal samples. The model assigned spectra from each sample evenly to the three metals. Beyond the three metal samples the RS model is able to differentiate all spectra accurately with minimal confusion. The confusion matrix of this model is shown in figure 5.6.



	AN	Al	BaSO4	Cu	Fe	KClO3	KClO4	KNO3	Marble
AN	39								
Al		5		7	6				
BaSO4			46						
Cu		18		14	15				
Fe		7		16	18				
KClO3						40			
KClO4							28		
KNO3								56	
Marble									20

Figure 5.6 Confusion matrix of the RS model.

Separately the RS and LIBS were able to identify all but the most spectrally similar samples. The low confusion may be partially due to an absence of confounding spectra, such as organics. Many organics produce similar LIBS spectra and the inclusion of multiple organics in the dataset would likely have increased confusion with LIBS models. The confusion observed in both models is expected, LIBS struggling to identify the samples that produce one primary potassium peak and RS unable to differentiate samples that are not RS active.

5.8 Low-Level Fusion

Low level data fusion is the simplest category of data fusion. This category covers methods in which the data is directly combined, often but not always, with a scaling factor [93, 94]. Low-level fusion methods have been used for LIBS-RS data before to identify mineral samples [98]. Two low-level methods will be compared; directly adding LIBS and RS data into one spectrum and concatenating the two data types into a single vector.

Concatenation

Concatenating the LIBS and RS spectra into a single vector produces a dataset with twice as many x values and thus computational time with this method was substantially slower when compared to other methods. While this is not a concern within this limited dataset, it would be a concern for larger datasets. Other than this, the method was found to be extremely effective, correctly identifying every spectrum and only suffering from a small amount of minor confusion detailed in table 5.4. Validation spectrum 223, a $KClO_3$ sample, with an abnormally intense LIBS was the only confusion unique to this model within the low-level fusion testing.

When removed spectra were tested on the concatenated model, only the first spectrum produced any confusion. This spectrum was a $KClO_4$ sample and the model was 94% certain of this categorisation. The sample was confused with $KClO_3$ indicating that this confusion was more likely to be a result of the intensity of the main potassium peak not the other peaks. The lack of spectra with peaks at the pixel value (~255 and 280) of the background aluminium spectrum in this $KClO_4$ spectrum limit the confusion from this error.

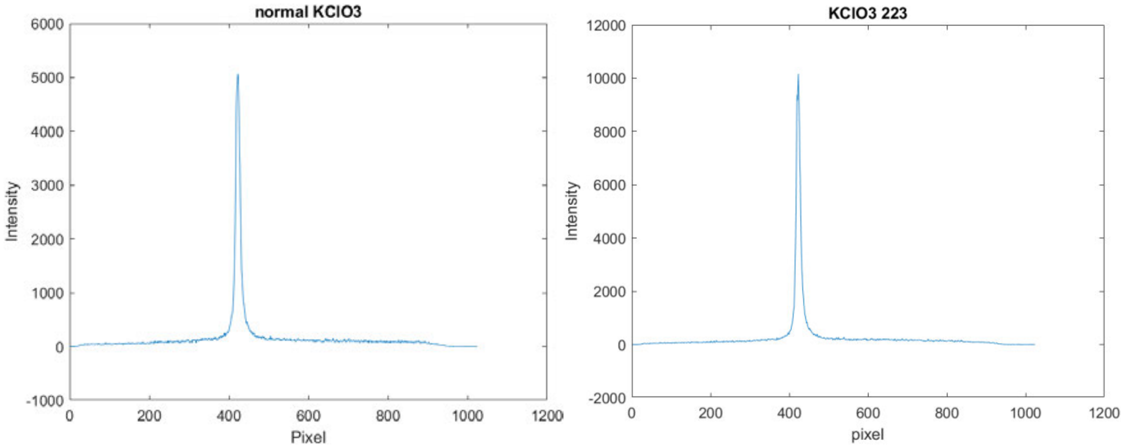


Figure 5.7 An average $KClO_3$ LIBS spectrum and 5.8 $KClO_3$ 223 approximatly twice as intense as the other sample.

Addition

LIBS and RS data were combined in a pixelwise addition in this fusion method. For instance, the RS data at pixel 243 was added to the LIBS data from pixel 243 to produce a single overlain spectrum from each pair. Addition of the datasets does not increase the dimensionality of the dataset compared to a single data type, thus avoiding slowdown, but it is not as precise as concatenation. When adding data, the spectra could become more confused if peaks overlap, particularly if a RS and a LIBS peak overlap from different spectra. The removed samples when tested with this model were classified correctly with no confusion.

Overall

The Concatenation and Addition models had the same points of confusion barring Validation spectrum 223 and Removed spectrum 1. All were minor points of confusion resulting in over 90% certainty of the classification. The spectra in the validation data which created confusion for these two models were 3 $KClO_4$ samples and one $KClO_3$ sample. Examining these four samples indicates these are likely due to an unusually intense LIBS signal.

Table 5.4 Number and degree of confusion from low-level fusion models to two significant figures.

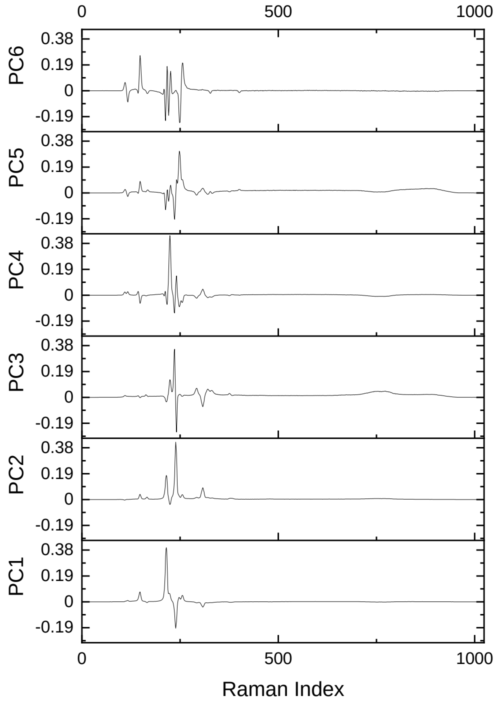
Model	$KClO_4$ confusion	$KClO_3$ confusion
Concatenation	3	2
certainty	96%, 93%, 95%	97% 98%
Addition	3	1
certainty	96% 96% 95%	98%

5.9 Mid-Level Fusion

Mid-level data fusion covers fusion methods performed after a larger degree of pre-processing than low-level fusion. In this instance a number of PCs from PCA of each data type have been combined. This method has been used in forensic applications [34, 93]. PCA is a statistical data reduction technique designed to examine the most important factors in a dataset in a small number of variables.

PCA was performed on the training data of both LIBS and RS portions of the datasets (PCA settings as section 3.4). Tests were performed with 3, 4, 5, and 6 PCs taken from the RS and LIBS datasets and concatenated together (RS first) to produce the training data. The verification data was then transformed into the PC space. An equal number of PCs were taken from the RS and LIBS verification data and concatenated to produce verification data with totals of 6, 8, 10 and 12 points to match the training data. The models were then trained on the training dataset and tested with the verification data.

Different numbers of PCs were selected for the models based on the total variance explained by each PC in the LIBS dataset. The LIBS dataset was chosen as it had lower explained variances, RS PCs were matched to the number of LIBS PCs. The numbers of PCs tested were 3 (90% explained variance), 4 (96% explained variance), 5 (98.9% explained variance) and 6 (99.5% explained variance). The variance explained by each individual LIBS and RS PC can be understood by comparing the loadings of the two datasets seen in figures 5.8 and 5.9.



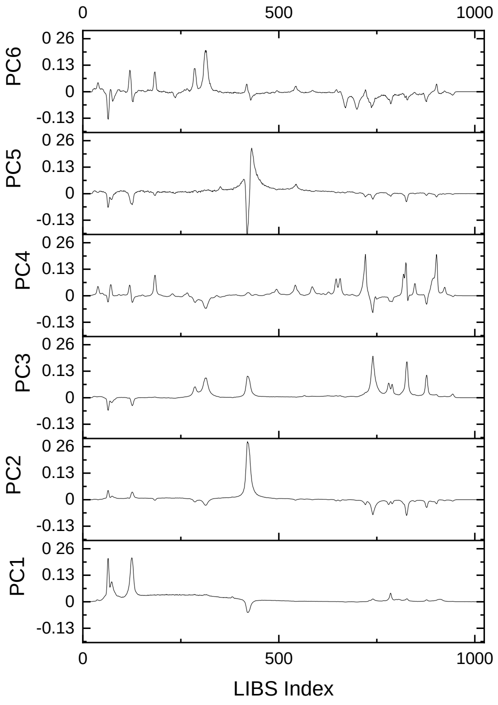


Figure 5.8 (left) The loadings of the RS dataset. Figure 5.9 (right) The loadings of the LIBS dataset.

The highly localised loadings of the RS dataset between pixels 200 and 250 in figure 5.8 align with the highly localised RS dataset in figure 5.3. The LIBS loadings, figure 5.9, show a wider range of variables affecting PCs. Comparing the loadings to the dataset in figure 5.4, PC1 appears to positively consider peaks in marble, iron and AN while negatively considering the potassium-based samples. PC2 strongly considers the three potassium-based samples while negatively considering the iron and barium peaks. PCs 3 and 4 consider a great deal of the spectrum with PC4 not considering the potassium-based samples. LIBS PC 5 appears to differentiate between $KClO_4$, $KClO_3$, and KNO_3. PC 6 seems to consider every peak in the dataset weakly.

Within the first three mid-level fusion models, 6, 5, and 4PCs, only one new source of confusion was encountered. All three models had a small degree of confusion on the same $KClO_4$ and $KClO_3$ spectra as the low-level fusion models. The 6PC model proved the least confused only having minor confusion on these same four spectra as the low-level fusion methods. The 5PC model had two more spectra it found confusing, another $KClO_3$ and a copper spectrum with a low intensity LIBS region. The 4PC model while confused on less spectra was less than 90% certain of one of the $KClO_3$ samples. Table 5.5 notes the degree of confusion of each spectra.

The final model, the 3PC model, proved to be significantly less effective than the other PCA based models. The 3PC model had significantly more spectra cause confusion and the classification of these spectra was less certain. A higher number of $KClO_4$ spectra were confused in this model as well as new spectra. These new spectra were a marble spectrum, three $BaSO_4$ spectra, and an iron spectrum. Table 5.5 shows the certainty of these spectra notably this model fell below 70% in one instance and below 90% in several.

Table 5.5 Number and degree of confusion from mid-level fusion models to two significant figures.

Model PCs	$KClO_4$	$KClO_3$	Other
6	2	2	
certainty	99%, 99%	94%, 94%	
5	3	2	1
certainty	99%, 97%, 98%	91%, 93%	Cu 96%
4	2	2	
certainty	99.7%[1], 97%	94% 90%	
3	9	2	5
certainty	96%, 88%, 99.4%[1], 87%, 99%, 99.4%[1], 99%, 98%, 99.4%[1]	87% 76%	$BaSO_4$ 74%, 94%, 88%, Fe 96%, Marble 69%

1) Reported to three significant figures as rounding would result in 100% certainty.

5.10 High-Level Fusion

The category of high-level data fusion covers fusion at the classification level. The method considers the data separately and only combines the data types once classification is completed [93]. Complex high-level data fusion can include a number of steps in a decision tree based on the initial categorisation [32, 48]. In this study, a simple form of high-level data fusion will be considered combining the predictions of a pair of LIBS and RS models.

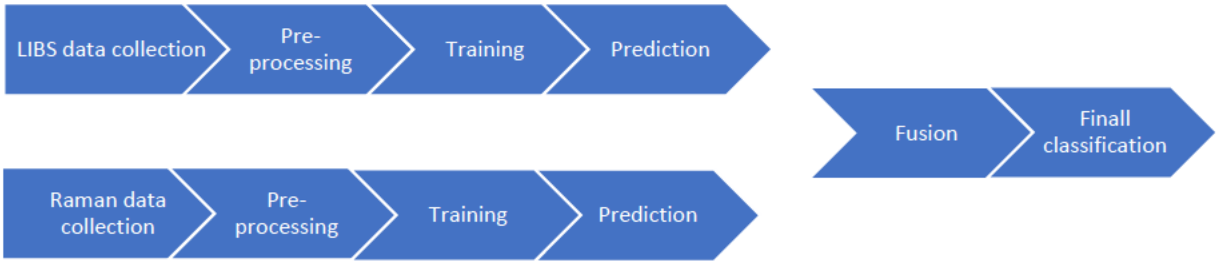


Figure 5.10 Flow chart showing the point of fusion in high-level fusion.

Individual classifiers were trained on the LIBS and RS spectra separately. The score value for the two are combined to produce a single prediction. While this approach produced accurate labels for each sample it includes the full confusion from the two individual models in its score values. This approach takes longer to perform than either the addition low-level or the mid-level model requiring two full models. Tables 5.6 and 5.7 show an example of this addition from the removed samples. In this example the RS data was 100% certain and added a 1 to the correct label of $KClO_4$. However, this method does not reduce the uncertainty from the LIBS spectrum, simply averaging it with that of the RS. In this dataset LIBS and RS have separate spectra that produce error but, in a dataset where both LIBS and RS are uncertain of a sample, this method would be unable to improve results. This simple high-level approach while effective, does not seem to be an ideal approach to data fusion in this application.

Table 5.6 (left) The unfused LIBS score values of three removed samples 5.7 (right) The fused score values of the same samples.

$KClO_3$	$KClO_4$	KNO_3
0	0.9263	0.0737
0.0877	0.7502	0.1621
0.0865	0.9135	0

$KClO_3$	$KClO_4$	KNO_3
0	1.9263	0.0737
0.0877	1.7502	0.1621
0.0865	1.9135	0

5.11 Conclusion

The-low level fusion approaches demonstrated some of the least confusion in their classifications of this dataset. Addition was shown to be slightly better in low-level fusion than concatenation and produces a smaller dataset. Both methods of low-level fusion produce large datasets and the addition method could prove detrimental if peak locations coincide between the LIBS and RS spectra of different samples. The low-level fusion methods thus have major flaws.

The mid-level fusion approach proved able to use very small amounts of data for this dataset and still classify all samples correctly. The most effective range of PCs considered was found to be between 95-99% explained variance in this case using 4, 5 or 6PCs. 5PCs (98.9% explained variance) proved effective as the least number of PCs to produce no confusion larger than the effect of one neighbour with distance weighting.

The high-level fusion method tested preserves all the confusion of both methods while being able to differentiate between all classes. High-level fusion requires keeping two full models and using both of them producing a larger computational load.

Chapter 6 – **Recommendations Regarding the Best Protocol**

6.1 Summary

Machine Learning (ML) and data fusion techniques were tested to develop a best practice method of identifying unknown samples from their LIBS and RS spectra. Chapters 3 and 4 describe the process of determining the most effective ML technique and chapter 5 describes the analysis of fusion techniques. From the work in these chapters both a ML technique and a fusion technique have been selected for use.

6.2 Machine Learning Technique

All tested algorithms were found to be able to produce high accuracies, 97% or higher, however, the most robust model was the Euclidian KNN. This algorithm was not initially as accurate as some of the others tested, however it maintained accuracy in both shrunken datasets and noisy datasets.

The Euclidian KNN also has a host of other advantages including the ability of the Euclidian distance metric to make use of K-D tree search algorithms. K-D tree search algorithms split the dataset into segments and based on the position of the new datapoint, only compares the new data to data in nearby segments, significantly reducing the processing load of large datasets [99]. KNN algorithms in general are understandable and easy to weight if an error is noticed so that if an error is noticed it is less likely to produce false negatives in an area of concern.

KNN scores are also easily accessible allowing a method of calculating the certainty of the result and a method for determining if the data is of no known class when data weighting is considered. Denying classification can be achieved with distance weighting by collecting the pre-normalised score values for each of the K neighbours and summing them. The value produced by this process can then be used to determine the degree of similarity of a particular test spectrum to the training data. The value can also be used to reject a classification if it is below the threshold of the class it is to be assigned to.

6.3 Fusion Technique

The fusion techniques tested were all capable of correctly classifying every spectrum in the fusion dataset. There were however varying degrees of confusion within these classifications. The confusion indicates which methods would be likely to produce errors in larger datasets with more confounding samples. The fusion techniques also varied in the amount of data for the classifier to consider, a serious consideration for larger datasets. Larger datasets take more processing power and, as such on a given set of hardware more time. This slowing effect is exponential with the dimensionality of the data and thus lower dimensionality is vital.

Using a concatenation approach, while effective, produces a significantly larger dataset increasing computational recourses required resulting in slowing classification. While effective, the addition approach risks confusion when the two methods produce peaks at the same point. The mid-level PCA approach produces less confusion and requires less computational recourses than the tested high-level approach.

The most efficient technique while maintaining high accuracy and low confusion is the mid-level PCA approach. The number of PCs required will depend on the dataset the classifier is intended to model. More complex datasets will require more PCs. This report therefore recommends utilising PCA accounting for more than 97% of the variation within the dataset. Explaining 95% variance is shown herein to leave greater room for confusion.

Chapter 7 - **Conclusion and Further Work**

7.1 Conclusion

In this study the aim was to determine the best set of machine learning-based data processing protocols for identifying whether an unknown 'sample' is an energetic material, using LIBS/RS. Pre-processing, machine learning technique and data fusion techniques were all considered towards this aim. A small degree of pre-processing was found to be effective. These were primarily removal of spectra that contained unusual signals from the training data and removal of spectra with low or no signal. Several machine learning techniques were tested to determine their applicability. Varied forms of KNN, LDA and ANN models were trained and compared. Of these techniques, Euclidian KNNs were found to be the most applicable. Finally fusion techniques were considered with a small range of low, mid and high-level fusion techniques tested. The mid-level PCA technique was found to be the most efficient technique while retaining accuracy.

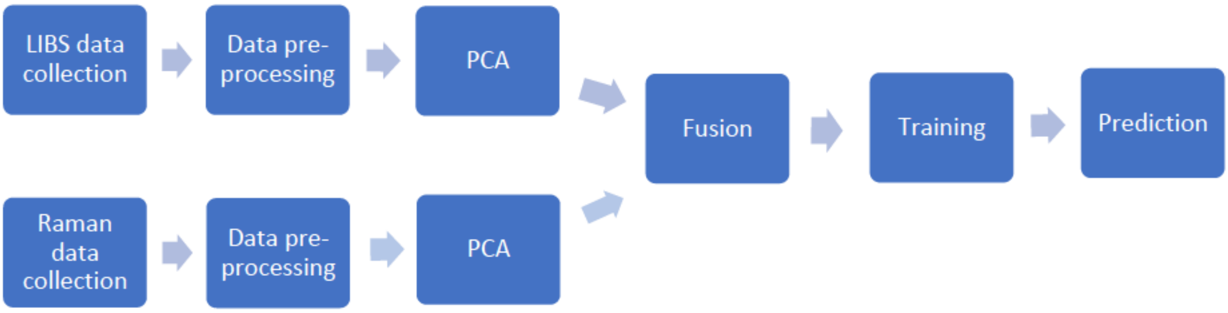


Figure 7.1 Simple flow chart of the recommended procedure.

The overall aim was therefore achieved with a machine learning based data processing procedure for identifying any given list of unknown samples and determining from LIBS and RS whether the unknown is an energetic material.

7.2 Further Work

Several questions and methods to extend this work were discovered during the course of the investigation. These include

- A greater degree of Pre-processing, normalisation will likely be required with larger datasets collected at different times.
- A model for the identification of the most commonly used energetics and the non-energetic samples likely to be encountered will need to be significantly larger and more expansive. This may highlight issues not apparent in this smaller scale test.

- While the three ML techniques tested here cover some of the more common methods of identifying spectra of this type, other ML types, support vector machines, decision trees etc, may prove to be more effective than anticipated
- Some tests were performed with non-energetic samples with similar spectra to energetics present. Tests on mixed samples of energetics and background materials will be necessary to determine the limits of trace detection with this system.
- More detailed high-level fusion techniques. Making use of a decision tree of models rather than using only two models would also be worthy of investigation. The tested high-level technique has flaws, such as maintaining the full confusion of both models, that more advanced and complex high-level fusion methods can avoid.

Exploring each of these aspects would extend this work considerably. Expanding these aspects would lead to refinements and extra knowledge that would enhance the overall recommended procedure.

References

1. Overton, I. (2020) A decade of global IED harm reviewed in, Action on Armed Violence,
2. Wold, S., Esbensen, K. & Geladi, P. (1987) Principal component analysis, *Chemometrics and Intelligent Laboratory Systems.* **2**, 37-52.
3. Goodman, M. B. (2015) An Ounce of Prevention in *Building a Global Shield to Defeat Improvised Explosive Devices*, Center for American Progress, Center for American Progress.
4. Bogue, R. (2011) Detecting mines and IEDs: what are the prospects for robots?, *The Industrial Robot.* **38**, 456-460.
5. IEDs in, GlobalSecurity org,
6. Wolf, S. J., Bebarta, V. S., Bonnett, C. J., Pons, P. T. & Cantrill, S. V. (2009) Blast injuries, *The Lancet.* **374**, 405-415.
7. Sekhar, P. K. & Wignes, F. (2016) Trace detection of research department explosive (RDX) using electrochemical gas sensor, *Sensors and Actuators B: Chemical.* **227**, 185-190.
8. Robledo, L., Carrasco, M. & Mery, D. (2009) A survey of land mine detection technology, *International Journal of Remote Sensing.* **30**, 2399-2410.
9. Liu, R. & Wang, H. (2010) Detection and localization of improvised explosive devices based on 3-axis magnetic sensor array system, *Procedia Engineering.* **7**, 1-9.
10. López-López, M. & García-Ruiz, C. (2014) Infrared and Raman spectroscopy techniques applied to identification of explosives, *TrAC Trends in Analytical Chemistry.* **54**, 36-44.
11. Kotidis, P., Deutsch, E. & Goyal, A. (2015) Standoff detection of chemical and biological threats using miniature widely tunable QCLs. **9467**.
12. Holmgren, E., Ek, S. & Colmsjö, A. (2012) Extraction of explosives from soil followed by gas chromatography–mass spectrometry analysis with negative chemical ionization, *Journal of Chromatography A.* **1222**, 109-115.
13. DeTata, D., Collins, P. & McKinley, A. (2013) A fast liquid chromatography quadrupole time-of-flight mass spectrometry (LC-QToF-MS) method for the identification of organic explosives and propellants, *Forensic Science International.* **233**, 63-74.
14. Ewing, R. G., Atkinson, D. A., Eiceman, G. A. & Ewing, G. J. (2001) A critical review of ion mobility spectrometry for the detection of explosives and explosive related compounds, *Talanta.* **54**, 515-529.
15. Blue, R., Vobecka, Z., Skabara, P. J. & Uttamchandani, D. (2013) The development of sensors for volatile nitro-containing compounds as models for explosives detection, *Sensors and Actuators B: Chemical.* **176**, 534-542.
16. Sekhar, P., Brosha, E., Mukundan, R., Linker, K., Brusseau, C. & Garzon, F. (2011) Trace detection and discrimination of explosives using electrochemical potentiometric gas sensors, *Journal of hazardous materials.* **190**, 125-32.
17. Moore, D. S. & Scharff, R. J. (2009) Portable Raman explosives detection, *Analytical and Bioanalytical Chemistry.* **393**, 1571-1578.
18. De Lucia, J. F. C. & Gottfried, J. L. (2013) Classification of Explosive Residues on Organic Substrates Using Laser Induced Breakdown Spectroscopy in
19. Castro-Suarez, J. R., Pacheco-Londoño, L. C., Vélez-Reyes, M., Diem, M., Tague, T. J. & Hernandez-Rivera, S. P. (2013) FT-IR Standoff Detection of Thermally Excited Emissions of Trinitrotoluene (TNT) Deposited on Aluminum Substrates, *Applied Spectroscopy.* **67**, 181-186.

20. Major, K. J., Sanghera, J. S., Farrell, M. E., Holthoff, E., Pellegrino, P. M. & Ewing, K. J. (2021) Spectral Considerations for Standoff Infrared Detection of RDX on Reflective Aluminum, *Applied Spectroscopy.* **76**, 163-172.
21. Wallin, S., Pettersson, A., Östmark, H. & Hobro, A. (2009) Laser-based standoff detection of explosives: a critical review, *Analytical and Bioanalytical Chemistry.* **395**, 259-274.
22. Wilschefski, S. C. & Baxter, M. R. (2019) Inductively Coupled Plasma Mass Spectrometry: Introduction to Analytical Aspects, *Clin Biochem Rev.* **40**, 115-133.
23. Justes, D. R., Talaty, N., Cotte-Rodriguez, I. & Cooks, R. G. (2007) Detection of explosives on skin using ambient ionization mass spectrometry, *Chemical Communications*, 2142-2144.
24. Na, N., Zhang, C., Zhao, M., Zhang, S., Yang, C., Fang, X. & Zhang, X. (2007) Direct detection of explosives on solid surfaces by mass spectrometry with an ambient ion source based on dielectric barrier discharge, *Journal of Mass Spectrometry.* **42**, 1079-1085.
25. Zhang, W., Tang, Y., Shi, A., Bao, L., Shen, Y., Shen, R. & Ye, Y. (2018) Recent Developments in Spectroscopic Techniques for the Detection of Explosives in *Materials*
26. Tabrizchi, M. & Ilbeigi, V. (2010) Detection of explosives by positive corona discharge ion mobility spectrometry, *Journal of Hazardous Materials.* **176**, 692-696.
27. Firtat, B., Moldovan, C., Brasoveanu, C., Muscalu, G., Gartner, M., Zaharescu, M., Chesler, P., Hornoiu, C., Mihaiu, S., Vladut, C., Dascalu, I., Georgescu, V. & Stan, I. (2017) Miniaturised MOX based sensors for pollutant and explosive gases detection, *Sensors and Actuators B: Chemical.* **249**, 647-655.
28. Izake, E., Sundarajoo, S., Olds, W., Cletus, B., Jaatinen, E. & Fredericks, P. (2013) Standoff Raman spectrometry for the non-invasive detection of explosives precursors in highly fluorescing packaging, *Talanta.* **103**, 20-7.
29. Gares, K. L., Hufziger, K. T., Bykov, S. V. & Asher, S. A. (2016) Review of explosive detection methodologies and the emergence of standoff deep UV resonance Raman, *Journal of Raman Spectroscopy.* **47**, 124-141.
30. González, R., Lucena, P., Tobaria, L. M. & Laserna, J. J. (2009) Standoff LIBS detection of explosive residues behind a barrier, *Journal of Analytical Atomic Spectrometry.* **24**, 1123-1126.
31. Harmon, R., DeLucia, F., McManus, C., McMillan, N., Jenkins, T., Walsh, M. & Miziolek, A. (2006) Laser-induced breakdown spectroscopy – An emerging chemical sensor technology for real-time field-portable, geochemical, mineralogical, and environmental applications, *Applied Geochemistry.* **21**, 730-747.
32. Shameem, K. M. M., Choudhari, K. S., Bankapur, A., Kulkarni, S. D., Unnikrishnan, V. K., George, S. D. & Kartha, V. B. (2017) A hybrid LIBS-Raman system combined with chemometrics: an efficient tool for plastic identification and sorting, *Analytical and Bioanalytical Chemistry.* **409**, 3299+.
33. Moros, J., ElFaham, M. M. & Laserna, J. J. (2018) Dual-Spectroscopy Platform for the Surveillance of Mars Mineralogy Using a Decisions Fusion Architecture on Simultaneous LIBS-Raman Data, *Analytical Chemistry.* **90**, 2079-2087.
34. Merk, V., Huber, D., Pfeifer, L., Damaske, S., Merk, S., Werncke, W. & Schuster, M. (2021) Discrimination of automotive glass by conjoint Raman and laser-induced breakdown spectroscopy and multivariate data analysis, *Spectrochimica Acta Part B: Atomic Spectroscopy.* **180**, 106198.
35. Moros, J., Lorenzo, J. A. & Laserna, J. J. (2011) Standoff detection of explosives: critical comparison for ensuing options on Raman spectroscopy–LIBS sensor fusion, *Analytical and Bioanalytical Chemistry.* **400**, 3353-3365.
36. Bellou, E., Gyftokostas, N., Stefas, D., Gazeli, O. & Couris, S. (2020) Laser-induced breakdown spectroscopy assisted by machine learning for olive oils classification: The effect of the experimental parameters, *Spectrochimica Acta Part B: Atomic Spectroscopy.* **163**, 105746.
37. Anabitarte, F., Cobo, A. & Lopez-Higuera, J. M. (2012) Laser-Induced Breakdown Spectroscopy: Fundamentals, Applications, and Challenges, *ISRN Spectroscopy.* **2012**, 285240.

38. Hahn, D. W. & Omenetto, N. (2010) Laser-Induced Breakdown Spectroscopy (LIBS), Part I: Review of Basic Diagnostics and Plasma—Particle Interactions: Still-Challenging Issues within the Analytical Plasma Community, *Applied Spectroscopy.* **64**, 335A-336A.
39. Kramida, A., Ralchenko, Yu., Reader, J. and NIST ASD Team (2020) (2021) NIST Atomic Spectra Database (version 5.8) in, National Institute of Standards and Technology, Gaithersburg, MD.
40. Vogt, D., Schröder, S. & Hübers, H.-W. (2017) *Investigation of Normalization Methods using Plasma Parameters for Laser Induced Breakdown Spectroscopy (LIBS) under simulated Martian Conditions*.
41. Lazic, V., Palucci, A., Jovicevic, S. & Carpanese, M. (2011) Detection of explosives in traces by laser induced breakdown spectroscopy: Differences from organic interferents and conditions for a correct classification, *Spectrochimica Acta Part B: Atomic Spectroscopy.* **66**, 644-655.
42. Mistek, E. (2019) Raman Spectroscopic Examination of Bloodstains for Forensic Purposes: Background and Race Determination in (Lednev, I. K. & Halámek, J., eds), ProQuest Dissertations Publishing,
43. Wiercigroch, E., Szafraniec, E., Czamara, K., Pacia, M. Z., Majzner, K., Kochan, K., Kaczor, A., Baranska, M. & Malek, K. (2017) Raman and infrared spectroscopy of carbohydrates: A review, *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy.* **185**, 317-335.
44. Dietze, D. R. & Mathies, R. A. (2016) Femtosecond Stimulated Raman Spectroscopy in pp. 1217-1217
45. McGregor, H., Wang, W., Short, M. & Zeng, H. (2016) Clinical utility of Raman spectroscopy: current applications and ongoing developments, *Advanced Health Care Technologies*, 13.
46. Larkin, P. (2011) *Infrared and Raman Spectroscopy : Principles and Spectral Interpretation*, Elsevier, Saint Louis, UNITED STATES.
47. Chen, H., Golder, M. R., Wang, F., Jasti, R. & Swan, A. K. (2014) Raman spectroscopy of carbon nanohoops, *Carbon.* **67**, 203-213.
48. Matroodi, F. & Tavassoli, S. H. (2014) Simultaneous Raman and laser-induced breakdown spectroscopy by a single setup, *Applied Physics B.* **117**, 1081-1089.
49. Vahid Dastjerdi, M., Mousavi, S. J., Soltanolkotabi, M. & Nezarati Zadeh, A. (2018) Identification and Sorting of PVC Polymer in Recycling Process by Laser-Induced Breakdown Spectroscopy (LIBS) Combined with Support Vector Machine (SVM) Model, *Iranian Journal of Science and Technology, Transactions A: Science.* **42**, 959-965.
50. Serrano, J., Moros, J., Sánchez, C., Macías, J. & Laserna, J. J. (2014) Advanced recognition of explosives in traces on polymer surfaces using LIBS and supervised learning classifiers, *Analytica Chimica Acta.* **806**, 107-116.
51. Araújo, D. C., Veloso, A. A., de Oliveira Filho, R. S., Giraud, M.-N., Raniero, L. J., Ferreira, L. M. & Bitar, R. A. (2021) Finding reduced Raman spectroscopy fingerprint of skin samples for melanoma diagnosis through machine learning, *Artificial Intelligence in Medicine.* **120**, 102161.
52. Lee, W., Lenferink, A. T. M., Otto, C. & Offerhaus, H. L. (2020) Classifying Raman spectra of extracellular vesicles based on convolutional neural networks for prostate cancer detection, *Journal of Raman Spectroscopy.* **51**, 293-300.
53. Mohammed, M. (2017) *Machine learning : algorithms and applications*, Boca Raton, Florida

London, England

New York : CRC Press.
54. Dulhare, U. N., Ahmad, K. & Bin Ahmad, K. A. (2020) *Machine Learning and Big Data : Concepts, Algorithms, Tools and Applications*, John Wiley & Sons, Incorporated, Newark, UNITED STATES.
55. Nielsen, M. A. (2015) *Neural Networks and deep learning*, Determination press.
56. Wolpert, D. H. (2002) The supervised learning no-free-lunch theorems, *Soft computing and industry*, 25-42.
57. Wolpert, D. (1992) On the connection between in-sample testing and generalization error, *Complex Systems.* **6**.

58. Koga, S., Zhou, X. & Dickson, D. W. (2021) Machine learning-based decision tree classifier for the diagnosis of progressive supranuclear palsy and corticobasal degeneration, *Neuropathology and Applied Neurobiology.* **47**, 931-941.
59. Murata, T., Yanagisawa, T., Kurihara, T., Kaneko, M., Ota, S., Enomoto, A., Tomita, M., Sugimoto, M., Sunamura, M., Hayashida, T., Kitagawa, Y. & Jinno, H. (2019) Salivary metabolomics with alternative decision tree-based machine learning methods for breast cancer discrimination, *Breast Cancer Research and Treatment.* **177**, 591-601.
60. Mu, Y., Liu, X., Yang, Z. & Liu, X. (2017) A parallel C4.5 decision tree algorithm based on MapReduce, *Concurrency and Computation: Practice and Experience.* **29**, e4015.
61. Zhao, O. (2021) Machine Learning Algorithms: Decision Trees in Huawei Enterprise Support Community.
62. Banerjee, A., Burlina, P. & Diehl, C. (2006) A support vector method for anomaly detection in hyperspectral imagery, *IEEE Transactions on Geoscience and Remote Sensing.* **44**, 2282-2291.
63. Niu, W., Lu, J. & Sun, Y. (2022) Development of shale gas production prediction models based on machine learning using early data, *Energy Reports.* **8**, 1229-1237.
64. Lee, T. R., Wood, W. T. & Phrampus, B. J. (2019) A Machine Learning (kNN) Approach to Predicting Global Seafloor Total Organic Carbon, *Global Biogeochemical Cycles.* **33**, 37-46.
65. Sundararajan, N. (2002) *Fully Tuned Radial Basis Function Neural Networks for Flight Control*, First edition. edn, New York, NY : Springer US : Imprint: Springer.
66. Dash, P. K., Mishra, S. & Panda, G. (2000) A radial basis function neural network controller for UPFC, *IEEE Transactions on Power Systems.* **15**, 1293-1299.
67. Tsangaratos, P. & Ilia, I. (2016) Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size, *CATENA.* **145**, 164-179.
68. Stephens, C. R., Hugo Flores, H. & Ana Ruíz, L. (2018) When is the Naive Bayes approximation not so naive?, *Machine Learning.* **107**, 397-441.
69. AS, C. S. The Unscrambler Methods User Manual in, CAMO Software AS,
70. Balakrishnama, S. & Ganapathiraju, A. (1998) Linear discriminant analysis-a brief tutorial, *Institute for Signal and information Processing.* **18**, 1-8.
71. Yu, S. K. Y. (2015) Honours thesis Classification of Mobile Phone Glass Display Screens by Laser Induced Breakdown Spectroscopy (LIBS).
72. Lei, B. (2017) *Classification, parameter estimation, and state estimation : an engineering approach using MATLAB*, Second edition. edn, Hoboken, New Jersey : Wiley.
73. Mathworks MATLAB documentation in
74. Ruder, S. (2016) An overview of gradient descent optimization algorithms, *arXiv preprint arXiv:160904747*.
75. Duchi, J., Hazan, E. & Singer, Y. (2011) Adaptive subgradient methods for online learning and stochastic optimization, *Journal of machine learning research.* **12**.
76. Kingma, D. P. & Ba, J. (2014) Adam: A method for stochastic optimization, *arXiv preprint arXiv:14126980*.
77. Doolette, S. L. (2014) *Classification of Organic Materials with Laser Induced Breakdown Spectroscopy*, Flinders University.
78. ChemSpider database in
79. Zhang, W., Zhou, R., Liu, K., Yan, J., Li, Q., Tang, Z., Li, X., Zeng, Q. & Zeng, X. (2020) Sulfur determination in laser-induced breakdown spectroscopy combined with resonance Raman scattering, *Talanta.* **216**, 120968.
80. Raman Band Correlation Table in
81. Iqbal, Z. S., K. Bulusu, Suryanarayana Autera, J. R. (1972-10-01) Infrared and Raman Spectra of 1,3,5-Trinitro-1,3,5-Triazacyclohexane (RDX) in Defense Technical Information Center.

82. Cadusch, P. J., Hlaing, M. M., Wade, S. A., McArthur, S. L. & Stoddart, P. R. (2013) Improved methods for fluorescence background subtraction from Raman spectra, *Journal of Raman Spectroscopy.* **44**, 1587-1595.
83. Kira, K. & Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. Paper presented at the *Aaai*.
84. Fawcett, T. (2006) An introduction to ROC analysis, *Pattern Recognition Letters.* **27**, 861-874.
85. Brownlee, J. (2020) How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification in *Imbalanced Classification* Machine Learning Mastery.
86. Manjunatha, P. (2021) Multiclass metrics of a confusion matrix in
87. Cao, Z., Cheng, J., Han, X., Li, L., Wang, J., Fan, Q. & Lin, Q. (2022) Rapid classification of coal by laser-induced breakdown spectroscopy (LIBS) with K-nearest neighbor (KNN) chemometrics, *Instrumentation Science & Technology*, 1-9.
88. Li, X., Yang, S., Fan, R., Yu, X. & Chen, D. (2018) Discrimination of soft tissues using laser-induced breakdown spectroscopy in combination with k nearest neighbors (kNN) and support vector machine (SVM) classifiers, *Optics & Laser Technology.* **102**, 233-239.
89. Cui, X., Zhao, Z., Zhang, G., Chen, S., Zhao, Y. & Lu, J. (2018) Analysis and classification of kidney stones based on Raman spectroscopy, *Biomed Opt Express.* **9**, 4175-4183.
90. Şahin, D. Ö., Akleylek, S. & Kılıç, E. (2021) On the Effect of k Values and Distance Metrics in KNN Algorithm for Android Malware Detection, *Advances in Data Science and Adaptive Analysis.* **13**, 2141001.
91. Othman, N. H., Lee, K. Y., Radzol, A. R. M., Mansor, W. & Rashid, U. R. M. (2019). Classification of Salivary Adulterated NS1 SERS Spectra Using PCA-Cosine-KNN. Paper presented at the *2019 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*.
92. Li, Q. & Chen, G. (2021) Recognition of industrial machine parts based on transfer learning with convolutional neural network, *PLOS ONE.* **16**, e0245735.
93. Castanedo, F. (2013) A Review of Data Fusion Techniques, *The Scientific World Journal.* **2013**, 704504.
94. Smolinska, A., Engel, J., Szymanska, E., Buydens, L. & Blanchet, L. (2019) Chapter 3 - General Framing of Low-, Mid-, and High-Level Data Fusion With Examples in the Life Sciences in *Data Handling in Science and Technology* (Cocchi, M., ed) pp. 51-79, Elsevier.
95. Gillen, G., Najarro, M., Wight, S., Walker, M., Verkouteren, J., Windsor, E., Barr, T., Staymates, M. & Urbas, A. (2015) Particle Fabrication Using Inkjet Printing onto Hydrophobic Surfaces for Optimization and Calibration of Trace Contraband Detection Sensors, *Sensors.* **15**, 29618-29634.
96. Dimova, M., Panczer, G. & Gaft, M. (2006) Spectroscopic study of barite from the Kremikovtsi Deposit (Bulgaria) with implication for its origin, *Geološki Anali Balkanskog Poluostrva.* **2006**.
97. Shiv K. Sharma, D. E. B., Anupam K. Misra, Tayro E. Acosta (2011) Detection of Chemicals with Standoff Raman Spectroscopy, *Spectroscopy*.
98. Rammelkamp, K., Schröder, S., Kubitza, S., Vogt, D. S., Frohmann, S., Hansen, P. B., Böttger, U., Hanke, F. & Hübers, H.-W. (2020) Low-level LIBS and Raman data fusion in the context of in situ Mars exploration, *Journal of Raman Spectroscopy.* **51**, 1682-1701.
99. Chen, Y., Zhou, L., Tang, Y., Singh, J. P., Bouguila, N., Wang, C., Wang, H. & Du, J. (2019) Fast neighbor search by using revised k-d tree, *Information Sciences.* **472**, 145-162.

Appendix 1 MATLAB code

Splitting a dataset

```
for i=1:"1/3 of data y's" %%a third of how much, here a third of all y's
take(i,1)=(i*3)-randi(3); %% selects one y in every 3
```

```
end
'validation'='data'(take,:); %takes the selected third into a new matrix
'testingdata'='data' %%define testing data
'testingdata'(take,:)=[];%% deletes validation data from testing data
%%randi to take a random third
```

Data culling

```
maxes=max(transpose('datamatrix'))%%finds the highest value in row of
datamatrix
n=1  %% alright, this finds everything below a
for i=1: “number of y's" %%given value, set to 1000 here
    if maxes(1,i)<1000 %% and deletes the relevant columns from
delmat(1,n)=i; %%a table this works by writing the y into a matrix
n=n+1
    end
end
'datamatrix'(delmat,:)=[]; %%and deleting the y’s in that matrix or table from
the data
```

Training a KNN

```
modelKNN=fitcknn(data, names,... %%data is the training data without labels,
names is your lables
"NumNeighbors", 10,... %%sets the value of K any number works
"distance", "cosine",... %%distance metric, options used are Euclidean
cityblock minkowski(cubic) cosine Spearman
"DistanceWeight", "inverse",... %%options are none inverse and squaredinverse,
custom functions can also be used
"NSMethod", "exhustive",...%%options KDtree or exhustive, KDtree only supports
linier distance metrics not correlational ones
"crossval", "on",...%% turns on cross validation
"kfold", 5,...%%sets the crossval method, here set to a 5fold Kfold
"Standardize", "on");%% turns on standardization, if you leave it out defaults
to off
```

Classifying with a trained KNN

```
['lable', 'KNNscore', 'cost']=predict('knnmodel','validation data'); %predicts
lables and also generates cost and score matrixes, “Kfoldpredict” for cross
validated models otherwise identical
'conmat'=confusionchart('validation names', 'lable') %%produces a confusion
matrix from the prediction
'metrics'=multiclass_metrics_common('conmat'.NormalizedValues)%%calculates
F1score precision recall and accuracy
```

Training an ANN

```
layers=[featureInputLayer(2048); fullyConnectedLayer(16); softmaxLayer;
classificationLayer]%% define the layers, how many data points you want to put
in how many hidden neurons and how you want it to output
NNnames=NNtraining(:,1); NNdata=NNtraining(:,2:end);
NNnames=table2array(NNnames); NNdata=table2array(NNdata); %%extracts data and
labels from a table and converts them to a matrix and a cell array the NN can
use
```

```
optionstest=trainingOptions('sgdm', 'plots', 'training-progress', 'shuffle',
'every-epoch', 'ValidationData', {valdatamat, valnames,} validationFrequency',
10); %%defines options, a lot to chose from
Nnet=trainNetwork(NNdata, NNnames, layers, optionstest)%% begins network
training
predmat=classify(Nnet, testdata); %%predicts with new data, into labels
NNconmat=confusionchart(testnames, predmat) %%same as for KNN predictions
makes a confusion matrix
NNmetrics=multiclass_metrics_common(NNconmat.NormalizedValues)%%calculates
F1score accuracy recall and precision
```

PCA and brining new data in to a PC space

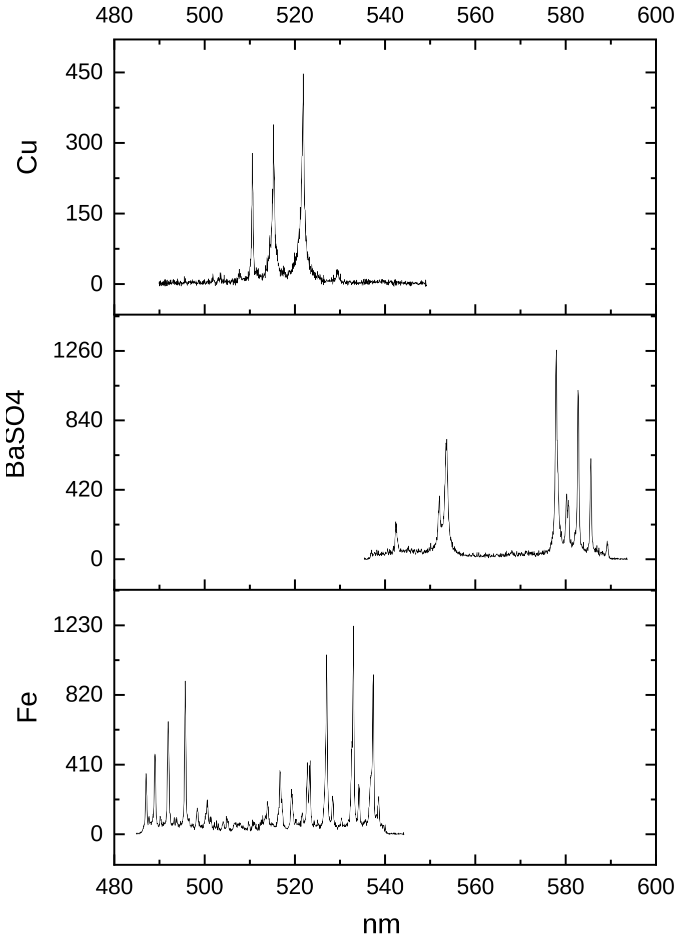
```
[coeffR,scoreR,~,~,explainedR,muR] = pca(Ramantestdata); %%perform PCA with
base settings to produce coefficets scores explained variance and row means

PCA6Rval=(Ramanvaldata-muR)*coeffR(1:end,1:x);%%new data into PCA, this will
give you the first x PCs only
```

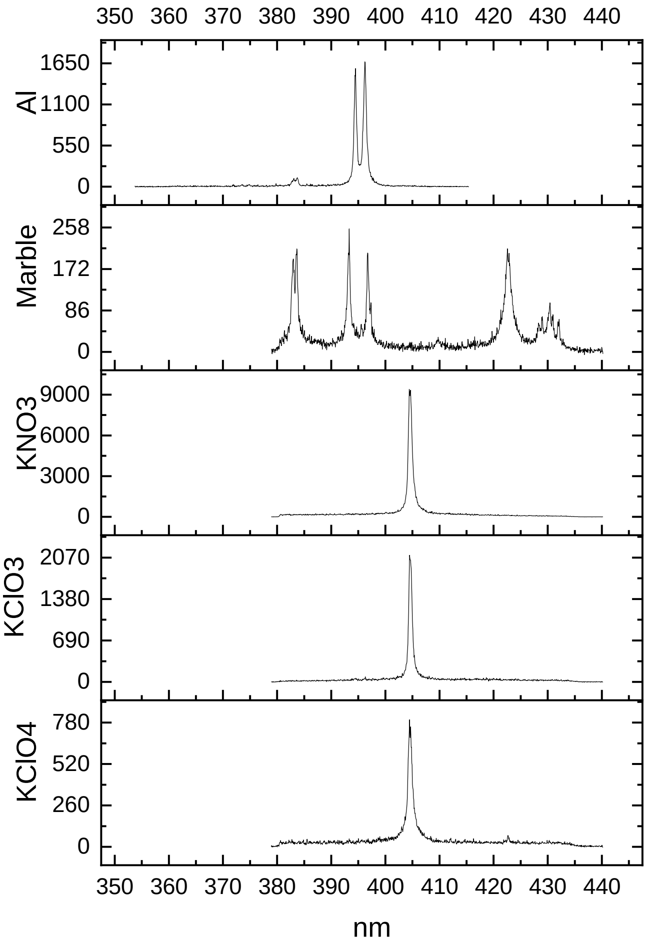
Making use of the Relieff function

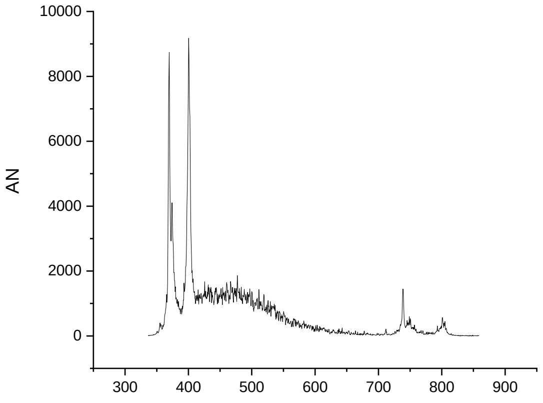
```
[idx,weights]=relieff('data','classes','number of negbours') %% this generates
indexies for the most impactful individual wavelengths
'relieffdata'='data'(:,idx(1:'x')); %% takes the x most impactful points into
a new matrix
```

Appendix 2

Fusion dataset spectra by wavelength


Cu
BaSO4
Fe
nm


Al
Marble
KNO3
KClO3
KClO4
nm


AN