Data Classification in High Throughput Screening

Shrilakshmi Bonageri

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science

University of Washington

2020

Committee:

Lilo Pozzo

David Beck

Program Authorized to Offer Degree:

Chemical Engineering

University of Washington

Abstract

Data Classification in High Throughput Screening

Shrilakshmi Bonageri

Chair of the Supervisory Committee:
Lilo Pozzo
Department of Chemical Engineering

Redox flow batteries offer an economical, low vulnerability means to store electrical energy at grid scale. Most commercial flow batteries use concentrated sulphuric acid with vanadium as the active redox material. While they have many inherent benefits, the use of vanadium drives up their cost. Deep eutectic solvents (DESs) are promising electrolyte solvents for less expensive and more sustainable organic redox species due to their biodegradability, low cost, high solvent quality and molecular versatility. High throughput screening (HTS) was used to synthesize and measure thermal and electrochemical properties of DESs to rapidly find an optimum DES for use in redox flow batteries. As a vast amount of data with varying characteristics based on the samples being tested is produced, data classification is essential to automate data analysis. Thus, convolutional neural networks and long short term memory networks were successfully used to

classify HTS data into predefined categories and eliminate noisy data. Suitable data analysis techniques were then developed for each category of classified data to obtain accurate results.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Any achievement, be it scholastic or otherwise does not depend solely on individual efforts but the guidance, encouragement, wisdom and cooperation of intellectuals, elders, and friends. Working on my thesis, with its highs and lows, has truly been a memorable and enriching experience. It is my privilege to have worked with eminent scientists and motivated colleagues. Thus, I take this opportunity to acknowledge everyone who has helped me in successfully completing my thesis.

I am indebted to my mentor Dr. Lilo Pozzo, for her patient guidance and timely support over the course of my research. She has been extremely encouraging and motivated me to venture into new avenues of research that she helped navigate.

I would also like to take the opportunity to extend my sincere gratitude to my committee member, Dr. David Beck, for providing me valuable suggestions and taking the time to help me on my panel despite his busy schedule.

I express special thanks to Jaime Rodriguez and Sage Scheiwiller for being great teammates and collecting experimental data which made my research possible.

I would like to acknowledge the Department of Chemical Engineering at UW, NSF CBET grant #1917340 and the Weyerhaeuser company for extending their financial support to facilitate research.

I would like to thank all my friends for their constant emotional and moral support, making my time at the University of Washington truly wonderful.

Lastly, I would like to thank my family without whom it would not be possible for me to be where I am now.

Chapter 1. INTRODUCTION TO HIGH THROUGHPUT EXPERIMENTATION

1.1 MOTIVATION

Rapid advances in technology bring with them many complex challenges and problems. One of the major problems we face today is environmental pollution caused by the technologies we currently use and its many consequences like global warming and climate change. There is now an immense need to discover new materials which can be used to produce and store renewable energy. Renewable-energy sources, such as solar and wind, are being deployed in larger numbers than ever before, but these sources are intermittent and often unpredictable. These characteristics limit the degree to which utilities can rely upon them. To ensure that renewable energy succeeds in delivering reliable power, cost effective and reliable storage are required at the grid scale [1]. Conventional rechargeable batteries offer a simple and efficient way to store electricity, but development till date has largely focused on transportation systems and smaller devices for portable power or intermittent backup power. However, metrics relating to size, and volume are far less critical for grid storage than in portable or transportation applications. Batteries for large-scale grid storage require durability for large numbers of charge/discharge cycles as well as calendar life, high round-trip efficiency, an ability to respond rapidly to changes in load or input, safe chemistry and low capital costs [2]. Redox flow batteries (RFBs) or redox flow cells (RFCs), shown schematically in Figure 1.1, promise to meet many of these requirements [3]. Redox flow batteries offer an economical, low vulnerability means to store electrical energy at grid scale. Redox flow batteries also offer greater flexibility to independently tailor power rating and energy rating for a given application than other electrochemical means for storing electrical energy. Redox flow batteries are suitable for energy storage applications with power ratings from 10’s of kW to 10’s of MW and storage durations of 2 to 10 hours [4].

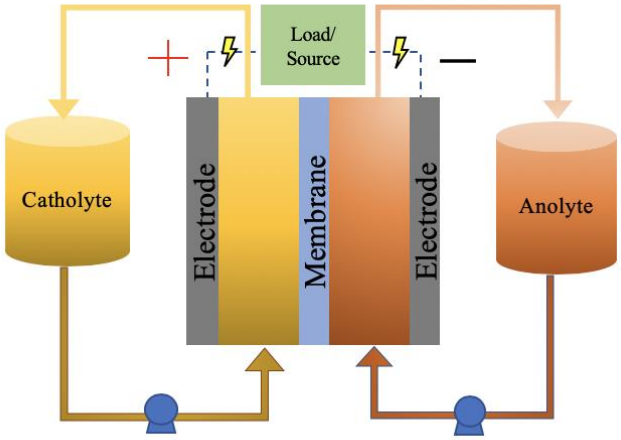


Figure 1.1. Schematic diagram of a redox flow battery

In the discharge mode, an anolyte solution flows through a porous electrode and reacts electrochemically to generate 'free' electrons, which flow through the external circuit. The charge-carrying species are then transported to a separator (typically an ion-exchange membrane), which serves to separate the anolyte and catholyte solutions. The general reactions can be written as follows for the anode (negative electrode) and cathode (positive electrode), respectively [4].

$$A^{n+} + xe^{-} \overset{charge}{\rightarrow} A^{(n-x)+} \; and \; A^{(n-x)+} \overset{discharge}{\rightarrow} A^{n+} + xe^{-} \quad (1.1)$$

$$B^{m+} - ye^{-} \overset{charge}{\rightarrow} B^{(m+y)+} \; and \; B^{(m+y)+} \overset{discharge}{\rightarrow} B^{m+} - ye^{-} \quad (1.2)$$

Most commercial flow batteries use concentrated sulphuric acid with vanadium salts as electrolytes. The electrodes are made of graphite bipolar plates. The surface of the bipolar plates is increased by adding a graphite felt layer which is perfused by the fluent electrolyte [4]. While Vanadium redox flow batteries have many inherent benefits, vanadium is an expensive metal which is not abundant in nature and can be produced only by difficult mining operations and it drives up the cost of a Vanadium redox flow battery system compared to other battery types. Thus, the focus now is to find alternative solvents and redox molecules which are a good fit for use in a redox flow battery. Deep eutectic solvents (DES) are promising electrolyte solvents for less expensive and more sustainable organic redox species due to their biodegradability, low cost, solvent quality and molecular versatility.

1.2 INTRODUCTION TO DEEP EUTECTIC SOLVENTS

Although most ionic solvents are suitable for use in an electrochemical cell, issues such as complex synthetic steps, cost, lack of toxicity data and availability limit their practical use for large scale application [5]. An alternative approach to simplifying the synthesis of Ionic Liquids (ILs) is to start with a straightforward quaternary ammonium halide and decrease the freezing point by complexing the anion to effectively delocalize the charge [6]. Some commonly used materials to synthesize DESs are shown in Figure 1.2(A). These mixtures form eutectics where a depression of freezing points, as shown in Figure 1.2(B), can be as large as 200°C [7]. The use of simple amides, acids and alcohols as complexing agents distinguishes them from other ILs, so the term deep eutectic solvents (DESs) is used [6]. These liquids often have high conductivities, viscosities, and surface tensions as well as negligible vapor pressures and volatility. Chemical structures of some typical components used to make DESs are given in Fig. 3 [8]. The fact that some of the hydrogen bond donors are commonly available as bulk commodity chemicals, such as urea, choline chloride and oxalic acid, gives them the potential for large-scale applications. In addition, the primary advantages of DESs are the extremely low cost of the precursors and their general biodegradability [7].

(A) (B)

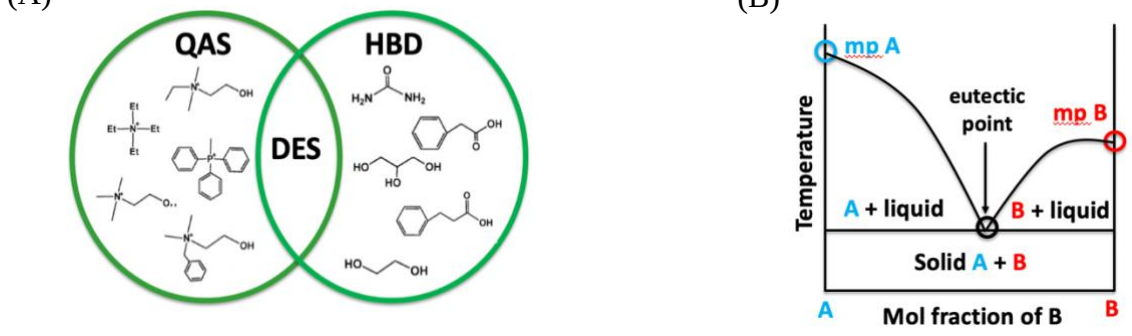


Figure 1.2. (A) Typical components of a DES; (B) Depression in freezing point of a DES

The design space for DES is enormous and high throughput screening of solvent mixtures of varying compositions is required to rapidly identify the composition of mixtures which form deep eutectic solvents and to quantify properties that will make them feasible for use in specific applications. Statistical design of experiments (DOE) is used to produce the optimal number of

samples from the vast design space so that the effect of all the factors involved in the experiment and all the different combinations possible are tested efficiently.

1.3 FACTORIAL EXPERIMENT DESIGN

The essential feature of good DOE designs is the simultaneous variation of several factors, which is a marked departure from the common practice of experiments that only vary one factor at a time. A *factor* of an experiment is a controlled independent variable; a variable whose levels are set by the experimenter. A *main effect* is a measure of the average change in the response when the control factor is changed. As mentioned in [9], factorial designs offer many advantages: each experimental run gives information on several factors, not just one; the experiment yields as much information about each factor as though it alone had been varied; valuable additional information is available through the ability to check for possible interactions among factors; and in the event that no interactions are found, there is a much broader base for generalizing conclusions on the main effect of any given factor, since the effect has been observed in a variety of experimental conditions.[9, 10]

Fractional Factorial design

A further advance, fractional factorial designs, was introduced by [11]. These designs allow experimenters to study the main effects and low-order interactions of several factors in far fewer runs, than would be required to complete the full factorial designs, by sacrificing the ability to estimate high-order interactions. Fractional factorial designs thus offer great economy of time and resources when, as is often the case, high-order interactions are negligible [10].

These DOE techniques are used to efficiently explore the vast design space of deep eutectic solvents. The factor is the composition of DES precursor materials and the main effect is the change is thermal and electrochemical properties of the synthesized DES. High throughput experimentation is used to synthesize deep eutectic solvents of varying factor and the main effect is observed by high throughput melting point determination and electrochemical testing to rapidly identify optimum DESs for use in redox flow batteries.

1.4 HIGH THROUGHPUT EXPERIMENTATION

High-throughput experimentation is a process that allows automated testing of large numbers of samples. It increases the output and decrease experiment costs and time. It leverages robotics and automation to quickly test large number of samples. It accelerates target analysis, as samples can quickly be screened in a cost-effective way. High throughput experimentation has been successfully implemented and has had a tremendous impact in various fields like catalysis, electronic and functional materials, polymer-based industrial coatings, sensing materials, and biomaterials. Following are some examples of high throughput experimentation and screening found in the literature:

- A practical hydrogen storage medium is required to fully exploit the advantaged of a fuel cell. Spatially resolved infrared imaging is used as a high throughput hydrogen storage candidate screening technique. A combination of pulsed laser deposition and magnetron sputtering was used to study hydrogen sorption capability of multi-compositional samples [12].
- Computational screening methods were used to pace the material discovery for heterogeneous catalysts and electrocatalysts. A density functional theory-based high throughput screening scheme was successfully used to theoretically evaluate the activity of over 700 binary surface alloys to identify a new electrocatalyst for the hydrogen evolution reaction (HER). The predicted activity for the alloy BiPt was found to be higher than Pt, the archetypical HER catalyst. This alloy was synthesized and tested experimentally and showed performance in agreement with the computational screening results [13].

High throughput experiment usually comprises of four steps as shown in Figure 1.3 below

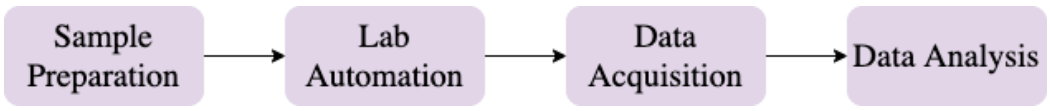


Figure 1.3. Steps involved in high throughput experimentation

1.5 Implementation Of HTE In The Pozzo Lab

1.5.1 *Sample preparation: Opentrons OT-2 pipetting robot*

The OT-2 is a pipetting robot which can be used to automate protocols and workflows used to prepare samples. The robot has open source hardware and software which lets us customize it to specific needs. It also comes with a click-and-drag graphical user interface that can be used to design and run protocols in minutes. It has 11 deck slots which fit eleven 96 well standard microplates, pipette tips and/or tube racks. The robot can fill up a 96 well plate as fast as 20 seconds. Protocols are tailored to prepare samples according to the requirements of the experiments. When performing exploratory experiments, design of experiments (DOE) is used to optimize the sample compositions and determine how the change in sample composition affects its properties. When optimum sample compositions are known, the composition of each of the 96 samples can simply be passed through the protocol. The open source software, hardware and the ease of designing protocols makes the OT-2 a very effective tool to prepare a large number of samples rapidly with minimal human aid. This makes it very suitable for high throughput screening experiments. The accuracy of pipetting by the OT-2 varies according to various factors such as viscosity and volume of the solvents to be pipetted. While it is attractive to produce multiple small-scale samples of materials at once using high throughput tools, it is important to validate the performance of the high throughput system by reproducing materials with good performance in laboratory scale synthesis and performance testing under conventional test conditions. Hence, the OT-2 is better suited to be used for screening of samples rather than accurate analysis of samples.

1.5.2 *High throughput experimentation using parallel experiments*

Musical Robot: HTE to determine melting point of deep eutectic solvents'

High throughput measurement of melting points was made possible through the use of an infrared camera and subsequent image analysis. The melting point of the DES was obtained by recording the temperature profile of the sample as it was heated and locating the inflection point in the profile that results from an increase in thermal conductivity as the sample melts. A python package was developed to acquire the experimental data and obtain accurate melting points of

multiple samples at once, while only requiring a matter of minutes to perform the physical measurement, and at low-cost. The high throughput setup is shown in Figure 1.4. In contrast, standard melting point determination techniques (e.g. differential scanning calorimetry) utilize equipment that is orders of magnitude more expensive and can take up to an hour for analysis of individual samples.

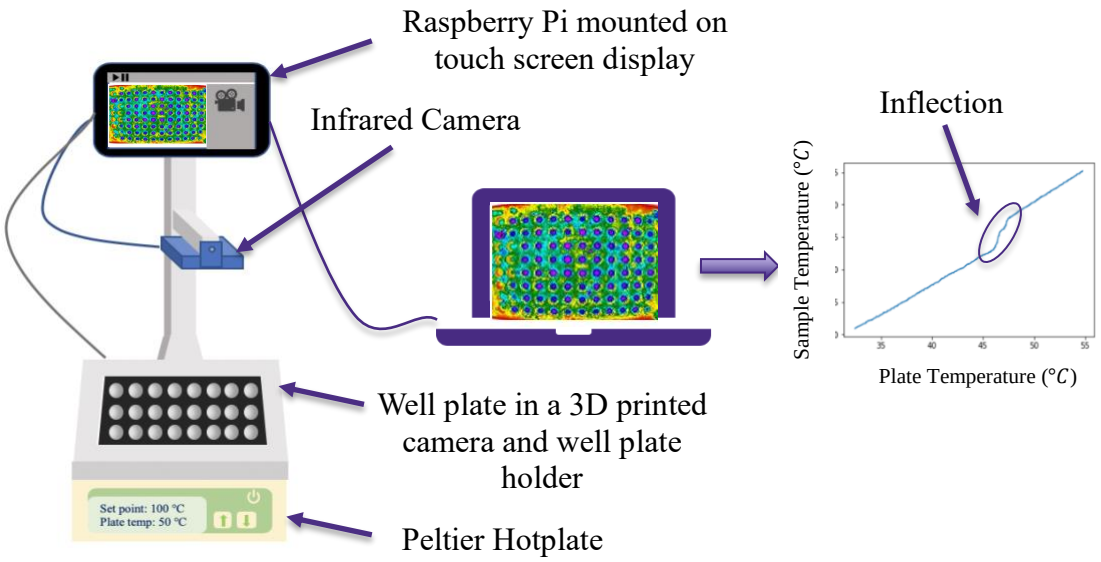


Figure 1.4. High throughput setup to determine melting point of deep eutectic solvents'

Volt Cycle: High throughput electrochemical testing of DES'

Deep eutectic solvents (DES) are considered as promising alternative to non-aqueous solvents used as electrolyte media in redox flow batteries due to their low cost, solvent quality and molecular design versatility. Since the design space of deep eutectic solvents in vast, high throughput synthesis and electrochemical testing of various solvent mixtures capable of forming DES is required to find an optimum deep eutectic solvent.

High throughput synthesis of solvents is carried out by using the Opentrons OT-2 robot. Mixtures of varying compositions of the stock solutions (QASs and HBDs) are formed using design of experiment techniques. A 96 well standard micro plate with screen printed electrodes is used to perform electrochemical tests on the samples. The screen printed well plate is connected to a 96 well connector and the gamry potentiostat. Electrochemical tests such as Cyclic

Voltammetry (CV) and Electrochemical Impedance Spectroscopy (EIS) can be performed on 96 samples at once by using the Gamry Potentiostat and its graphical user interface (GUI). The experimental data obtained from each sample is saved in a separate text file. The high throughput setup is shown in Figure 1.5. The python module built is capable of reading multiple data files and saving the required information such as voltage and current data, experiment parameters from the files in a python dictionary which makes it easier to process, classify, and analyze the data to determine the various properties of the cyclic voltammograms. The obtained properties are then used to identify samples with most suitable properties for use in redox flow batteries.

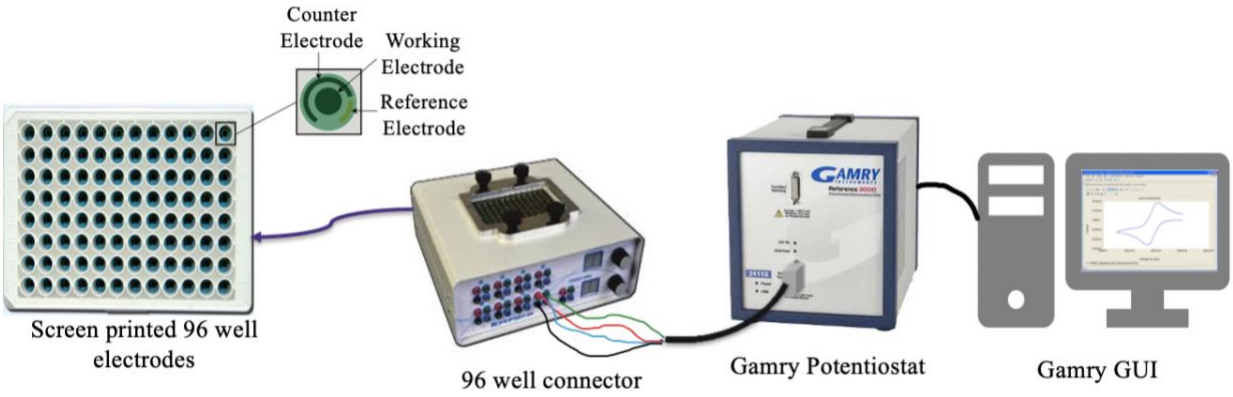


Figure 1.5. High throughput setup for electrochemical testing of deep eutectic solvents'

1.5.3 *Data Acquisition*

High throughput experimentation produces a large amount of data which makes it essential to implement seamless data acquisition, data management and data storage techniques. Data acquisition has to be automated to minimize the time taken to complete all the experiments and make high throughput screening seamless. This can be done effectively by using a high-level programming language like python. A single script of python can be used to enable an end to end system which can be used to run the experiment, collect the data, classify the data and analyze it.

1.5.4 *Data Analysis*

The data obtained from the high throughput experimentation has to be analyzed to determine necessary properties of the samples. This can be done by using the conventional analytic methods or by using machine learning/ deep learning techniques. However, a large amount of training data is usually required to use the latter techniques. Since, high throughput

experimentation provides us with a large amount of data, using machine learning and deep learning techniques to classify and analyze data have proven to be effective in several cases.

1.5.5 *Problems encountered in analyzing data obtained through THE*

High throughput screening has several inherent benefits, but an efficient, automated and integrated setup is required in order to fully utilize its benefits. Seamless methods to handle user input, perform high throughput experiments, and acquire, manage and finally analyze the obtained data are required to produce significant results autonomously with minimal user intervention. Various problems are encountered in this process and the issue described below ascertained the importance and need of data classification in high throughput screening.

The high throughput experimentation setup to determine melting point produces temperature profiles of samples during phase change. The melting points of samples manifest as an inflection in these plots. The melting points could be obtained by determining the derivative of the temperature profiles since the derivative will comprise a peak at the point of inflection. The sample temperature at the peak is then considered as the sample's melting point. However, this analytical approach could not provide accurate results because some of the temperature profiles obtained did not have an inflection point and the derivatives will still show peaks, due to noise and random fluctuations, which would be erroneously interpreted as the melting point as illustrated in Figure 1.6.

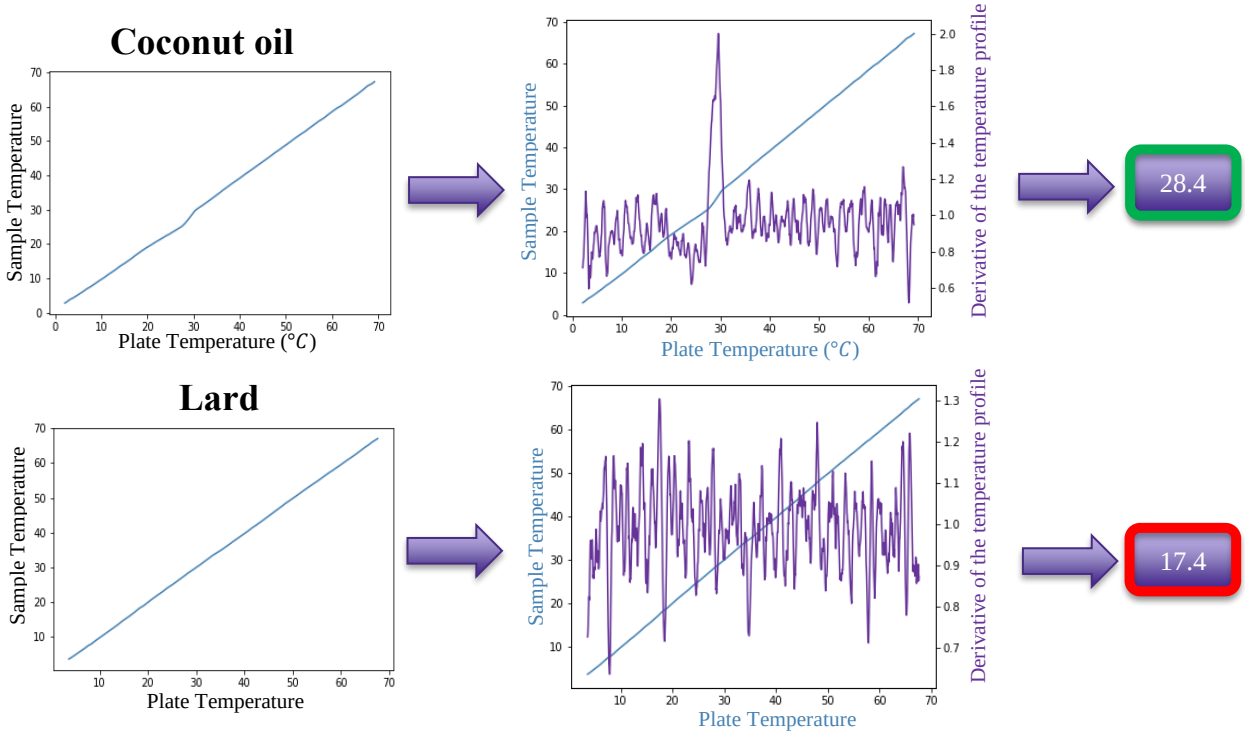


Figure 1.6. Inaccurate determination of melting point as a consequence of using the same analytical approach on two different types of data

1.6 Need For Data Classification In High Throughput Screening

There are many more reasons that necessitate data classification in high throughput screening since the data produced can vary widely. High throughput screening produces a vast amount of data and classification find its purpose in these three key processes in data analysis as shown in Figure 1.7**:**

- **Quality Control:** Data classification can be used to ensure the quality of the data by eliminating data with:

1. Artefacts due to the instruments used.
2. Artefacts caused due to external interference
3. Artefacts caused due to erroneous instrument settings/ calibration.
4. Artefacts caused during automatic instrument maintenance.

The Figure 1.6 showcases the need of quality control as pre-sorting the data to classify temperature profiles between ‘clear inflection point’ and ‘no clear inflection point’ would

eliminate the problem of mistakenly identifying noise as a melting process, while still doing this with autonomous methods that do not involve a human screening process.

- **Anomaly Detection:** Data classification can be used to detect anomalies in the samples. Anomalies can occur due to:

1. Unexpected sample behavior or reactions.
2. Inaccurate sample preparation
3. Sample Contamination

- **Data Binning:** Classifying data into predefined categories to enable appropriate analysis

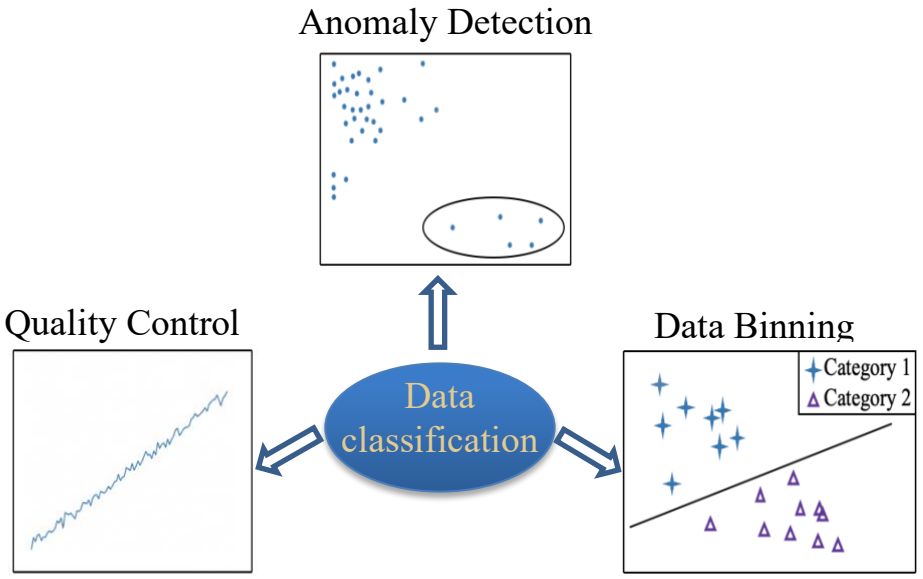


Figure 1.7. Key purposes of data classification prior to analysis in a high throughput setup

1.7 CONCLUSION

High throughput screening produces a vast amount of data with varying characteristics based on the sample being tested. The automation of the experimental setup also increases the ambiguity in the data. As a consequence, it is essential to classify data in order to automate the data analysis. Appropriate techniques for data analysis can be developed for each category of classified data to obtain accurate data.

Chapter 2. DATA ACQUISITION AND DATA PROCESSING

2.1 NEED FOR DATA ACQUISITION AND DATA MANAGEMENT TECHNIQUES

One of the key benefits of high throughput experimentation is that it can rapidly conduct a large number of experiments providing us with large amounts of data. Thus, a streamlined pipeline and workflow of data with minimum human intervention is also required to efficiently utilize the obtained data and to optimize the end-to-end process of high throughput screening. Efficient and effective ways to handle user inputs, establish communication between various instruments and robust data handling are essential to automate the laboratory experimental procedure and to ensure integrity, accountability, and preservation of data. Using a programming language like Python is beneficial because a common interface can be used to perform the experiment and to handle the input and output of data. The experimental data obtained can also be further processed and analyzed using Python. Moreover, many modern laboratories are now successfully driven fully by software platforms and high-level programming languages like Python. Below are some of the examples found in the recent literature.

1. The automation of data acquisition from various instruments is necessary for an high throughput experimentation setup. The data acquired from all the instruments also has to be consistent and compatible with data from other instruments that are part of the setup for an easy and efficient flow of data. The open-source, flexible, and extensible Python package for Laboratory Automation, Control, and Experimentation (PLACE) uses modular organization and clear design principles; therefore, it can be easily customized or expanded to meet the needs of diverse laboratories [14].

2. “Applying artificial intelligence to materials research requires abundant curated experimental data and the ability for algorithms to request new experiments. ESCALATE (Experiment Specification, Capture and Laboratory Automation Technology) is an ontological framework and opensource software package—that solves this problem by providing an abstraction layer for human- and machine-readable experiment specification, comprehensive and extensible (meta-) data capture, and structured data

> reporting. ESCALATE simplifies the initial data collection process, and its reporting and experiment generation mechanisms simplify machine learning integration. An initial ESCALATE implementation for metal halide perovskite crystallization was used to perform 55 rounds of algorithmically-controlled experiment plans, capturing 4336 individual experiments."[15]

This chapter describes the use of python for data acquisition, data management, data preprocessing and data analysis performed to automate and integrate the end-to-end process of high throughput screening in order to obtain significant results.

2.2 Data Acquisition Using Python

High throughput experiments produce enormous amounts of data making it essential to have an organized and efficient way of acquiring the data and for managing it. For example, electrochemical testing on all samples in a completely filled 96 well standard micro plate produces 96 text files (.txt) with all the information pertinent to the experiment. To plot and analyze the data obtained using an application such as Microsoft Excel for each sample would be tedious and time consuming. Using conventional applications such as Microsoft Excel to plot and analyze large amounts of data obtained by high throughput experiments may then prove to be the largest bottleneck in the process of high throughput screening. Thus, a streamlined pipeline and workflow of data with minimum human intervention is required to optimize the process. The programming language Python was used in the following ways to acquire, manage and save data.

Python interface for the gamry potentiostat

The Pozzo group's Gamry Reference 600 potentiostat, which is used for performing electrochemical tests in this work, also has its own graphical user interface (GUI) but this has the following shortcomings which makes it unsuitable for high throughput experiments:

- The data obtained from the experiment has to be exported as a text file to be saved.
- The data analysis has to be done manually after plotting the data in an application like Microsoft Excel.

The above issues were tackled by developing a Python interface based on Component Object Module (COM). COM is a technology from Microsoft that allows objects to communicate without the need for either object to know any details about the other or even the language it's implemented in. The Python module 'comtypes' was used to implement COM using Python and establish communication between the gamry potentiostat and a computer. The data obtained using the Python interface could be readily plotted and analyzed to determine its properties.

Python SDK for the FLIR Lepton IR camera

An infrared camera is also used by the Pozzo group to capture the melting process of deep eutectic solvent (DES) samples on a well plate. As with the case of the gamry potentiostat, the camera also comes with a graphical user interface (GUI). When using the GUI, videos of the recorded melting process have to be manually saved as tiff files on the local disk and then imported to a Python script to analyze videos to determine the melting point. Using the python SDK eliminates this extra step and enables the end-to-end process of running the experiment and analyzing the data to be executed using a single Python script.

2.3 DATA MANAGEMENT

After importing to Python, experimental data is then converted into arrays or Python data dictionaries and saved as pickle files. The well-plate number or file name are then used as python dictionary keys. All data is saved in the form of python pickle files. The Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it 'serializes' the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script. The python libraries 'pandas' and 'tabulate' are then used to present the data in the form of tables. The python library 'matplotlib' is also used to visualize the data.

2.4 DATA PREPROCESSING

Data preprocessing is done to convert raw experimental data into a consistent and efficient format. The raw data can often be incomplete or inconsistent. It has to be preprocessed so that it can be accurately classified and analyzed. Data preprocessing include the following steps.

Data cleaning

Data cleaning consists of resolving all the inconsistencies in the data such as units and filling of missing values. The necessary conversions are made to make all units consistent with each other. Missing values are also filled by using 1-D interpolation or spline interpolation. Both these techniques are implemented by using the functions *interp1d* and *BSpline* available in the Python package *SciPy* in its module *interpolate*.

Data integration

Data integration is required to integrate all the required data from various sources.The cleaned data from various files are integrated into python dictionaries with their file names or well-plate numbers as the keys. The dictionaries are then saved as pickle files so that the they can be used later in any python script. Saving them as a pickle files also will eliminate the need to clean the data again if used in a different python script.

Data transformation

The data is standardized or scaled to make it consistent and compatible with other sources of data. This also improves the performance of machine learning and deep learning models. The result of **standardization** (or **Z-score normalization**) is that the features will be rescaled so that they'll have the properties of a standard normal distribution with

$$\mu = 0, \sigma = 1$$

where μ is the mean (average) and σ is the standard deviation from the mean; standard scores (also called z scores) of the samples are calculated as follows:

$$z = \frac{x-\mu}{\sigma} \tag{2.2}$$

Standardizing the features so that they are centered around 0 with a standard deviation of 1 is not only important if we are comparing measurements that have different units, but it is also a general requirement for many machine learning algorithms. Intuitively, we can think of gradient descent as a prominent example (an optimization algorithm often used in logistic regression, SVMs, perceptrons, neural networks etc.) with features being on different scales, certain weights may update faster than others since the feature values x_j play a role in the weight updates

$$\Delta w_j = -\eta \frac{\partial j}{\partial w_j} = \eta \sum_i \left(t^{(i)} - o^i\right) x_j{}^{(i)} \tag{2.2}$$

where η is the learning rate, *t* the target class label, and *o* the actual output. Other intuitive examples include K-Nearest Neighbor algorithms and clustering algorithms that use, for example, Euclidean distance measures. In fact, tree-based classifiers are probably the only classifiers where feature scaling doesn't make a difference.

An alternative approach to Z-score normalization (or standardization) is called **Min-Max scaling**. In this approach, the data is scaled to a fixed range - usually 0 to 1. The cost of having this bounded range in contrast to standardization is that we will end up with smaller standard deviations, which can suppress the effect of outliers. A Min-Max scaling is typically done via the following equation:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2.3}$$

Data reduction

Reducing the quantity of data aims at selecting pertinent information only as an input to the data mining algorithm. Thus, data reduction identifies and, eventually, leads to discarding information which is irrelevant or redundant. Ideally, after data reduction has been carried out, the user has to do with datasets of smaller dimensions representing the original dataset. It is also assumed that the reduced dataset carries the acceptable or identical amount of information as the original dataset [16]. The aim of data reduction is not to lose extractable information but to increase the effectiveness of the machine learning or deep learning algorithms [17]. The variance of multiple high dimensional data sets can be compared with each other using dimensionality reduction techniques like Principle Component Analysis and Multidimensional scaling. This is particularly useful when different datasets are to be used to train a machine learning algorithm.

Data discretization

Data discretization is defined as a process of converting continuous data attribute values into a finite set of intervals with minimal loss of information [18]. The process of discretization has become one of the most effective data preprocessing techniques [19-21]. Discretization translates quantitative data into qualitative data, procuring a nonoverlapping division of a continuous domain [22]. There is a necessity to use discretized data by many machine learning algortihms. Among its main benefits, discretization causes that the learning methods show remarkable improvements in learning speed and accuracy [22]. Neural networks always have a fixed input size. Multiple datasets of varying sizes can be converted to the required size by using linearly spaced discretized values.

2.5 Data Classification In High Throughput Screening

As mentioned earlier, high throughput experimentation produces a large amount of data. Analyzing the data obtained manually for each sample at a time would be cumbersome and prove to be the bottleneck in the process of obtaining meaningful results from the high throughput experiments. As all the other steps involved in high throughput experimentation, the data obtained from the experiments also has to be analyzed using an automated analytical system with minimal human aid.

It is rather obvious that a high throughput experiment performed to screen samples can produce various types of results based on the different kinds of samples that are present. Some samples might meet the screening criteria and others might not. The first need for classification arises because samples meeting the screening criteria have to be separated from the samples which do not. Furthermore, all the samples meeting the criteria might not produce the exact same type of data.

Data obtained from an experiment usually have different characteristics such as:

- Noise in the data due to instruments or human interference.
- Missing observations in some datasets due to external factors or issues from the instruments.
- Artefacts due to unexpected reactions or sample behavior

Some of the samples might also be anomalies and could produce data with characteristics that are completely different from those that are expected for an experiment and potentially

incompatible with the modeling approach that is being used. The same analytical approach cannot be used to analyze data belonging to different classes to obtain meaningful results. Hence, classification of the data is absolutely essential prior to analyzing it appropriately based on the class it belongs to obtain significant results.

2.6 INTRODUCTION TO NEURAL NETWORKS

Computers are extremely fast at numerical computations, far exceeding human capabilities. However, the human brain has many abilities that would be desirable to translate into a computer. These include: the ability to quickly identify features, even in the presence of noise: to understand, interpret, and act on probabilistic or fuzzy notions (such as "Maybe it will rain tomorrow"); to make inferences and judgments based on past experiences and to relate them to situations that have never been encountered before; and to suffer localized damage without losing complete functionality (i.e. fault tolerance). So even though the computer is faster than the human brain in numeric computations, the brain far outperforms the computer in many other tasks. This is the underlying motivation for trying to understand and model the human brain.

The neuron is the basic computational unit of the brain. A human brain has approximately 10_{11} neurons acting in parallel. The neurons are highly interconnected, with a typical neuron being connected to several thousand other neurons. Early work on modeling the brain started with models of the neuron. The McCulloch-Pitts model of the neuron was one of the first attempts in this area[23]. The McCulloch-Pitts model is a simple binary threshold unit. The neuron shown in Figure 2.1 receives a weighted sum of inputs from connected units, and outputs a value of one (fires) if this sum is greater than a threshold. If the sum is less than the threshold, the model neuron outputs a zero value. Mathematically we can represent this model as [24]:

$$y_i = \Theta\left(\sum_j w_{ij}x_j - \mu_i\right) \tag{2.4}$$

Where y_i is the output of the neuron i, w_{ij} is the weight from neuron j to neuron i, x_j is the output of neuron j, μ_i is the threshold for neuron i, and Θ is the activation function defined as

$$\Theta(netinput) = \begin{cases} 1, & if\ netinput \geq 0 \\ 0, & otherwise. \end{cases}$$

A node layer is a row of those neuron-like switches that turn on or off as the input is fed through the net. Each layer's output is simultaneously the subsequent layer's input, starting from an initial input layer receiving given input data. Deep learning or Artificial neural network is the name we use for 'stacked neural networks'; that is, networks composed of several layers.
An artificial neural network learning algorithm is a computational learning system that uses a network of functions to understand and translate a data input of one form into a desired output, usually in another form. ANNs map inputs to outputs to find correlations. They are known as universal approximators, because they can learn to approximate an unknown function $f(x) = y$ between any input x and any output y, assuming they are related at all (by correlation or causation, for example). In the process of learning, a neural network finds the right `f`, or the correct manner of transforming x into y, whether that be $f(x) = 3x + 12$ or $f(x) = 9x - 0.1$ [25].

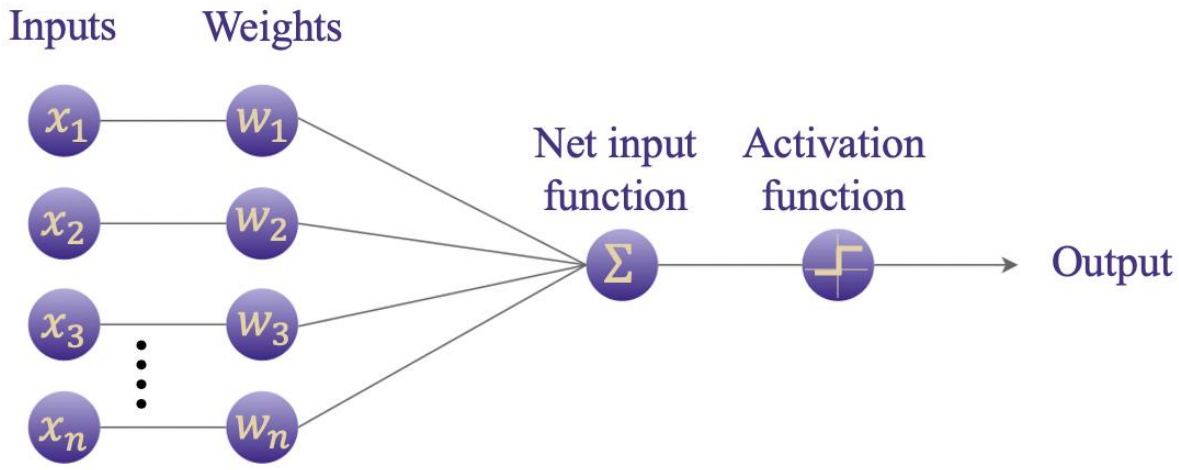


Figure 2.1. Representation of a neuron

Artificial neural networks are powerful learning models that can be used successfully for a myriad supervised and unsupervised machine learning tasks. They are suited especially well for machine perception tasks, where the raw underlying features are not individually interpretable. This success is attributed to their ability to learn hierarchical representations, unlike traditional methods that rely upon hand-engineered features [24]. Hence, neural nets can be successfully used for classification and prediction tasks. As mentioned earlier, the data obtained from high throughput experimentation has to be classified and binned into appropriate categories before it can be analyzed to make useful deductions from it. This was done in by using various machine

learning and deep learning techniques like Logistic Regression, Convolutional Neural Networks (CNNs) and Long short term memory (LSTM) networks as explained herein.

2.7 CONVOLUTIONAL NEURAL NETWORKS (CNN)

A Convolutional Neural Network (ConvNet/CNN) is a deep learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. They are hierarchical neural nets mainly used for tasks pertaining to Image processing like object detection, object segmentation and object classification in images. They are designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns.

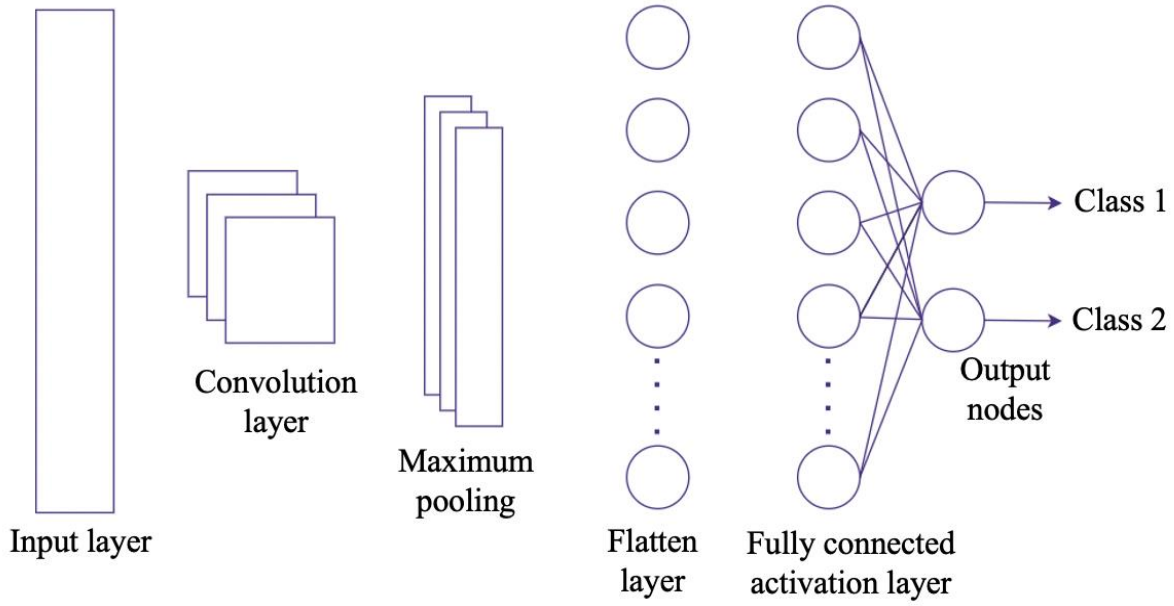


Figure 2.2. Layers in a CNN

Image classification, which can be defined as the task of categorizing images into one of several predefined classes is a fundamental problem in computer vision. It forms the basis for other computer vision tasks such as localization, detection, and segmentation [26]. Although the task can be considered second nature for humans, it is much more challenging for an automated system. Some of the complications encountered include viewpoint-dependent object variability and the high in-class variability of having many object types. Traditionally, machine learning techniques use hand designed features extracted from image descriptors to train the models. But

for a classification problem, the image descriptors change based on the class the image belongs to and hence the task of extracting the features analytically becomes formidable [27]. Neural networks tackle this by synthesizing their own features. CNNs exploit multiple layers of nonlinear information processing, for feature extraction and transformation as well as pattern analysis and classification. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex [28]. An image is input directly to the network, and this is followed by several stages of convolution and pooling. Thereafter, representations from these operations feed one or more fully connected layers. Finally, the last fully connected layer outputs the class label. A typical Convolution neural network as shown in Figure 2.2 has the following layers [29]:

1. Convolution Layer

The convolutional layers serve as feature extractors, and thus they learn the feature representations of their input images. The neurons in the convolutional layers are arranged into feature maps. Each neuron in a feature map has a receptive field, which is connected to a neighborhood of neurons in the previous layer via a set of trainable weights, sometimes referred to as a filter bank[30]. Inputs are convolved with the learned weights in order to compute a new feature map, and the convolved results are sent through a nonlinear activation function. All neurons within a feature map have weights that are constrained to be equal; however, different feature maps within the same convolutional layer have different weights so that several features can be extracted at each location [27, 30].

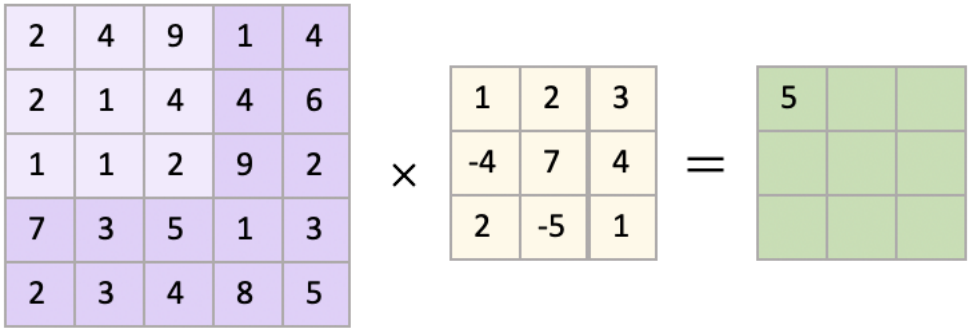


Figure 2.3. Convolution operation

2. Maximum Pooling

The purpose of the pooling layers is to reduce the spatial resolution of the feature maps and thus achieve spatial invariance to input distortions and translations [27, 30]. Given an input image of size 4 x 4, if a 2 x 2 filter and stride of two is applied, max pooling outputs the maximum value of each 2 x 2 region as shown in the image.

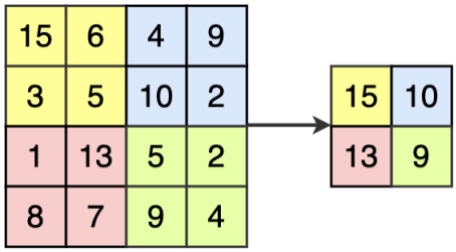


Figure 2.4. Maximum Pooling

3. Fully Connected Layers

Several convolutional and pooling layers are usually stacked on top of each other to extract more abstract feature representations inmoving through the network. The fully connected layers that follow these layers interpret these feature representations and perform the function of high-level reasoning [31]. The last fully connected layer outputs the class labels.

The CNNs can be trained using labelled data (Supervised learning) belonging to different categories and then be used to classify new data belonging to one of the categories in the trained dataset. The motivation behind using CNNs is that the data obtained from both high throughput experimentation can be plotted to produce images. The characteristic features in the plots also vary based on the category the data belongs to. A CNN can also be trained to identify inconsistencies in the data such as noise, missing data points or artefacts arising from unexpected reactions since all these can be discerned from a plot of the data. Convolutional networks can thus be successfully trained to classify images belonging to different categories.

2.8 LONG SHORT TERM MEMORY NETWORKS (LSTM)

In traditional neural networks, the information from a previous layer is not pertained by the next layer. This can cause issues while using them for tasks in which the present observation is dependent on previous observations such as text prediction or time series forecasting (sequential data). Recurrent neural networks address this issue and they have networks with loops in them to persist information. LSTMs are a special kind of recurrent neural networks capable of learning long term dependencies. This unique feature of these networks makes them suitable for use in tasks pertaining to pattern recognition in sequential data (Sequence classification) and forecasting for time series (sequential) data.

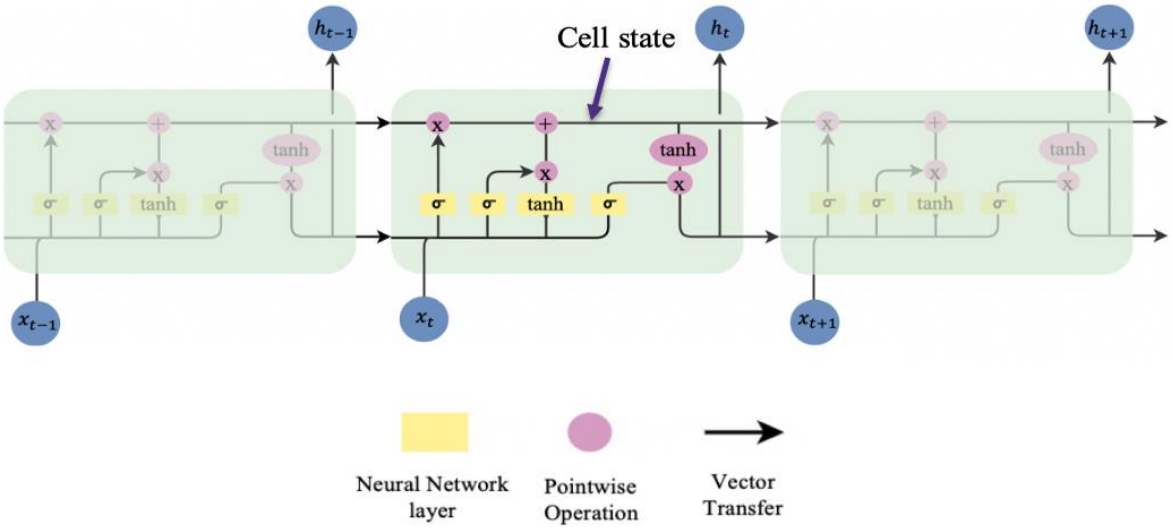


Figure 2.5. Repeating module of a LSTM

The repeating module, A, of a LSTM looks at some input x_t and outputs a value h_t. The chain like structure allows information to be passed from one step of the network to the next. The repeating modules of a LSTM network are different in that instead of having a single neural network layer, there are four, interacting in a very special way. In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denotes its content being copied and the copies going to different locations.

The key to LSTMs is the cell state as shown in Figure 2.5, the horizontal line running through the top of the diagram shown below, which is used to retain information learnt previously. Information is added or deleted from the cell state based on the present information being processed by the network through the use of gates. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!” [32]

The motivation behind using LSTM is that there is a distinguishable and discriminative pattern in data belonging to each category obtained from both the melting point and cyclic voltammetry experiments. In the cyclic voltammetry data, the distinguishable pattern in different categories of data is the number of peaks. In the data obtained from HTE to determine melting point of DES’, some samples have a temperature profile with an inflection and some samples have a linear temperature profile. The inflection in the plot is thus a discernable pattern in the data which can be extracted by the LSTM. The CNN and the LSTM architectures used to classify data are shown in Figure 2.6.

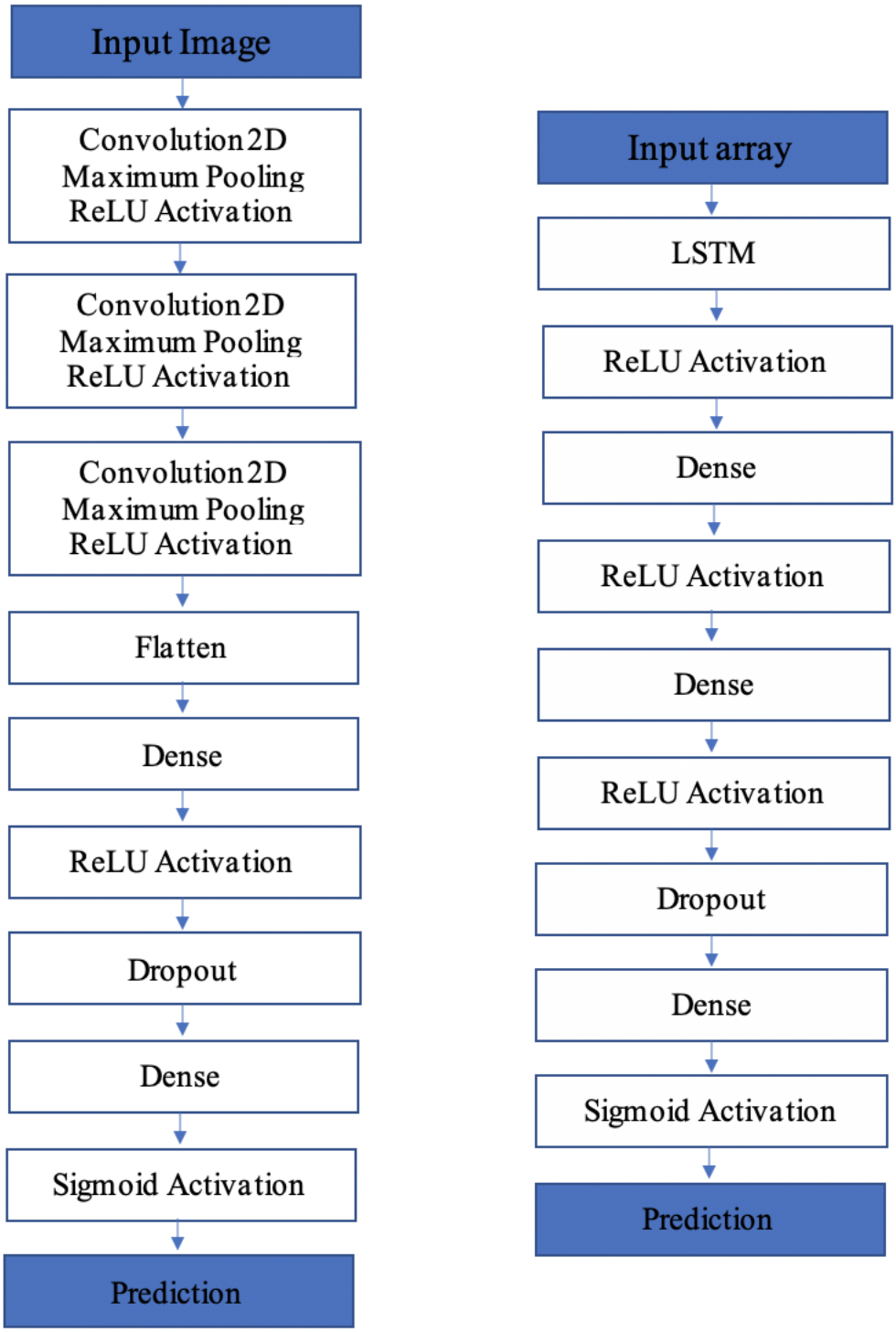


Figure 2.6. Neural network architectures used for data classification

2.9 TRAINING THE NEURAL NETWORKS

Neural nets have to be trained using existing data before they can be used to make predictions or classify new data. Neural networks in general use learning algorithms to adjust their free parameters (i.e., the biases and weights) in order to attain the desired network output. The most common algorithm used for this purpose is backpropagation [27]. Backpropagation computes the gradient of an objective (also referred to as a cost/loss/ performance) function as shown in Figure 2.7 to determine how to adjust a network's parameters in order to minimize errors that affect performance [29].

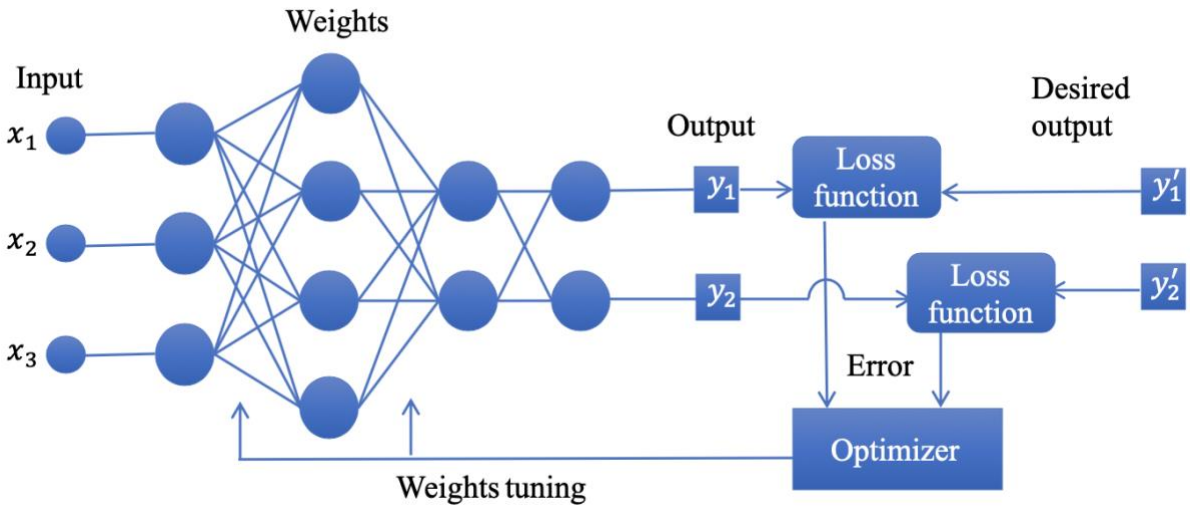


Figure 2.7. Optimization of loss function using backpropagation

Neural networks generate outputs only based on the correlation formed (i.e., the biases and weights calculated) between the inputs and the outputs during the training. A good training dataset thus needs to have the following qualities usually designated as the 5 V's of data as shown in Figure 2.8(A):

1. **Volume** referring to the enormous amount of data to effectively train a neural net.
2. **Velocity** refers to the high-speed accumulation of data to make the process efficient.
3. **Variety** referring to a data arising from different sources.
4. **Veracity** refers to data without any inconsistencies (i.e. preprocessed data)
5. **Value** refers to the usefulness of the data.

The training data set thus has to be vast and must have the above-mentioned qualities to effectively train a neural network and obtain accurate results. The accuracy of the neural networks is evaluated by using a validation dataset and a test dataset. The training and validation datasets are obtained by splitting a dataset as shown in Figure 2.8(B). The test dataset or the holdout dataset consists of data which the network has never seen before and is not a part of the training or validation dataset.

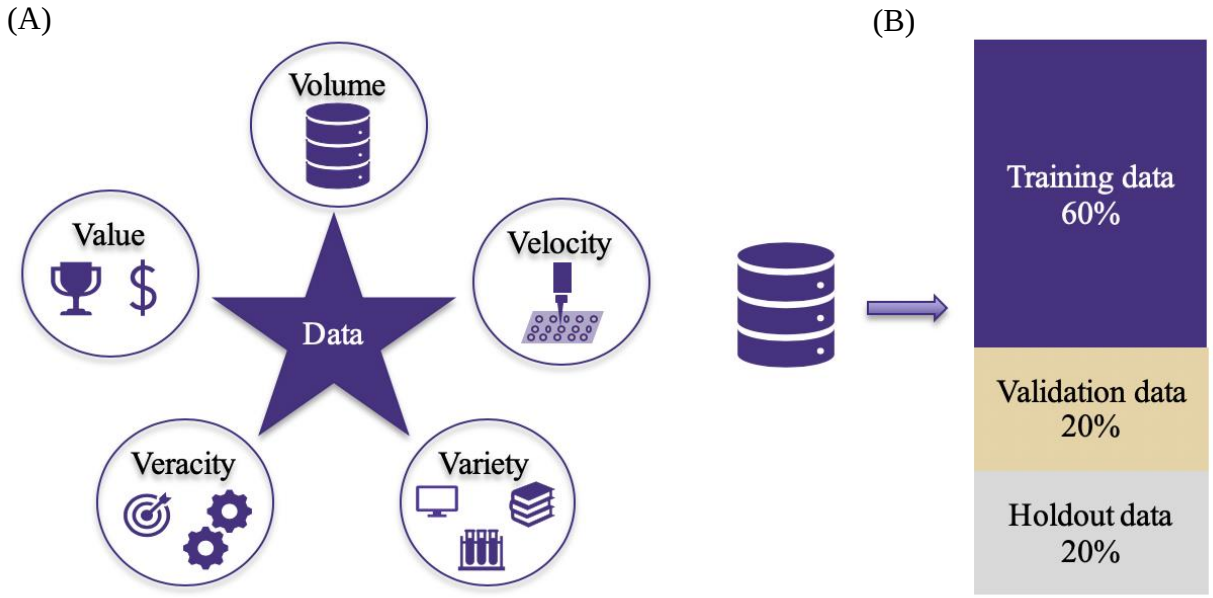


Figure 2.8. (A) The five V's of data; (B) Splitting a dataset into training, validation and holdout data

Generating Synthetic Data

Synthetic data is data that is generated programmatically. It is a fast and cheap way to generate the vast amount of data required by machine learning and deep learning models. The data generated has to mimic all the features of the experimental data and has to be compatible for use with the experimental data itself. This can be achieved by either simulating the experiment analytically or by simply using statistical models which generate data similar to the experimental data. The synthetic data has to meet all the required constraints to create realistic data profiles. Since the experimental data can sometimes be noisy due to instrumentation or external factors and can have artefacts due to some unexpected reactions, it is conventional to add some amount of noise to the data analytically. The chapters 3 and 4 cover more information about the methods

used to generate synthetic data to mimic both the cyclic voltammetry dataset and the phase change temperature profiles.

2.10 Root Mean Square Deviation (RMSD)

The root mean square deviation or the root mean square difference is the measure of difference or variation between the same features in different datasets as shown in Figure 2.9. The sum of the squared difference between each datapoint of the feature is different curves is calculated and then is divided by the number of datapoints to obtain root mean square deviation.

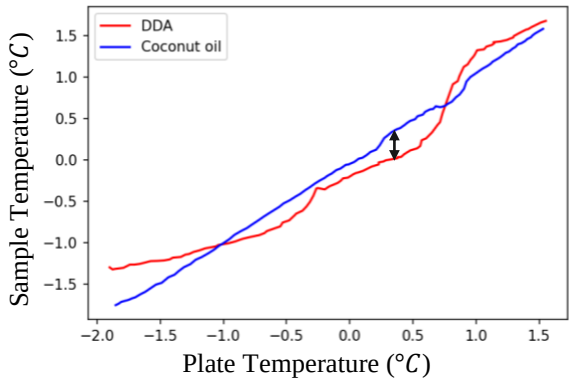


Figure 2.9. Measuring the variance between two curves

Suppose the first curve has feature values from y_1 to y_n and the second curve has feature values from x_1 to x_n, the root mean square deviation is calculated using the below formula:

$$RMSD = \frac{\sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}}{n} \quad (2.5)$$

It is used to determine how features vary amongst same dataset and also to compare two datasets. In this case, root mean square deviation was used to capture the variance of features in both the synthetic dataset and the experimental dataset. It was also used to compare the experimental dataset with the synthetic dataset and measure the difference between them. The following steps were followed to determine the Root mean square deviation of the experimental dataset, synthetic dataset and between the experimental and synthetic dataset. The temperature profiles of Dodecanoic acid (DDA) and Phenyl propanoic acid (PPA) during phase change are used as examples.

1. The number of datapoints in all the datasets have to be the same. This is done by fitting all the curves with a spline function and obtaining equal number of values at linearly spaced intervals.
2. All the datasets have to be standardized so that they are consistent and compatible with the other datasets. The result of standardization (or Z-score normalization) is that the features will be rescaled so that they'll have the properties of a standard normal distribution

$$\mu = 0, \sigma = 1$$

The Figure 2.10(A) and 2.10 (B) below show how standardization makes two curves consistent and compatible with each other.

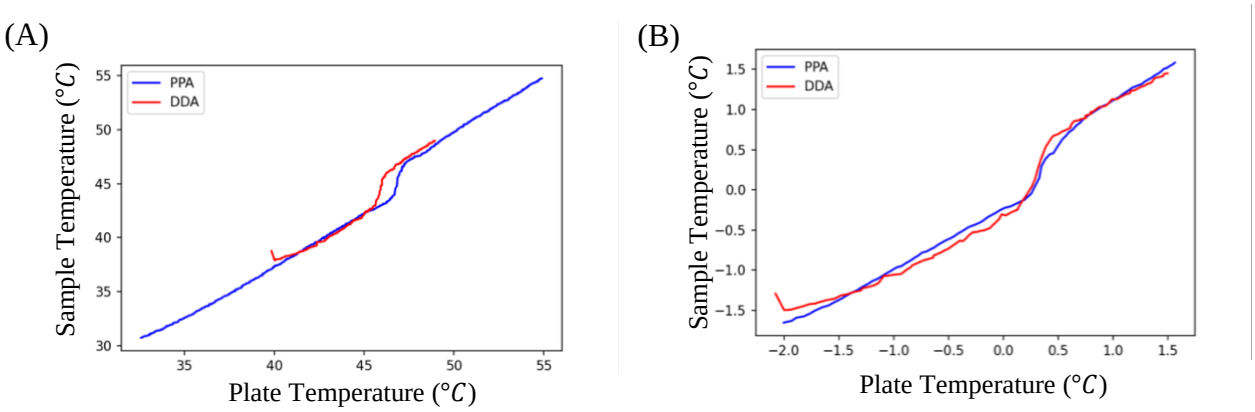


Figure 2.10. (A) Temperature profiles before normalization; (B) Temperature profiles after normalization

3. The root mean square deviation is calculated between each curve and all the other curves in the dataset. Calculating RMSD between all the curves in a dataset gives us the distribution of RMSD for a dataset. The Figure 2.11 shows the distribution of RMSD obtained for the experimental dataset from which the PPA and DDA curves were obtained.

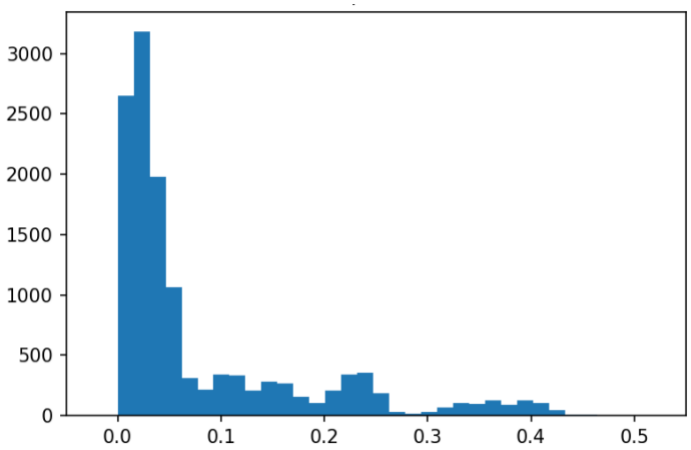


Figure 2.11. Root mean square deviation in an experimental dataset

4. A root mean square deviation of zero means that there is no difference between the curves. The larger the value, the more is the variance between multiple curves.

The above procedure was performed on various datasets, both experimental and synthetic to capture the variance of essential features and also to compare the variance of features in the experimental dataset and the synthetic datasets that were generated.

2.11 MULTIDIMENSIONAL SCALING

Multidimensional scaling (MDS) seeks a low-dimensional representation of the data in which the distances respect well the distances in the original high-dimensional space. It is a technique used for analyzing similarity or dissimilarity data. It attempts to model similarity or dissimilarity data as distances in a geometric space. The data can be ratings of similarity between objects, interaction frequencies of molecules, or trade indices between countries[33].

MDS was performed on synthetic and experimental datasets to measure the similarity/dissimilarity in the datasets. MDS reduced the dimensionality of every curve in the dataset to a point in geometric space as shown in the Figure 2.12**.**

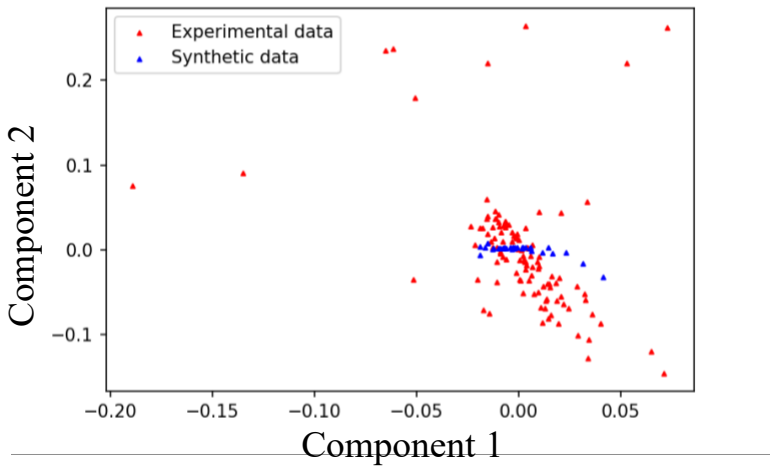


Figure 2.12. Multidimensional scaling of experimental and synthetic datasets

2.12 CONCLUSION

Deep learning algorithms require a vast amount of preprocessed and consistent data to work effectively. With efficient data acquisition and management techniques, the data produced by high throughput experimentation can be effectively used to train neural networks to perform data classification and data analysis.

Convolutional neural networks (CNNs) and Long short term memory networks(LSTMs) were used to classify data obtained from high throughput experimentation to determine melting point of deep eutectic solvents to eliminate the samples with noisy temperature profiles and to identify the samples with an inflection in their temperature profiles to enable appropriate analysis. This is described in detailed in Chapter 3. They were also used to classify cyclic voltammograms into predetermined categories based on the type of reaction the sample undergoes to enable determination of the correct characteristics. This is described in detail in Chapter 4.

Chapter 3. HIGH THROUGHPUT MEASUREMENT OF DEEP EUTECTIC SOLVENTS' MELTING POINT USING IR BOLOMETRY

Deep eutectic solvents (DES) are novel solvents that can be easily produced at low-cost for several important applications, such as chemical synthesis, extractions, electrochemistry, and even pharmaceutical drug delivery. An important property which is used to determine their feasibility in various applications is their melting point. Melting point of new samples is usually measured using a *Differential scanning calorimetry* (DSC) which takes up to hours to make a single measurement and analyze it. The design space for DES is vast and using DSC to measure melting temperature of all the synthesized eutectic mixtures would take an enormous amount of time. Hence, high throughput measurement of melting points is required to rapidly identify eutectic mixtures with melting points that are feasible for their specific application.

3.1 WORKING PRINCIPLE

High throughput measurement of melting points was made possible through the use of an infrared camera and subsequent image analysis as shown in Figure 3.1. The melting points of various samples were obtained by recording an infrared video of a well plate filled with samples being heated simultaneously using a temperature-controlled plate. The infrared camera detects the infrared energy(heat) being radiated from the well plate and converts it into an electronic signal, which is then processed to produce a thermal image on a video monitor and perform temperature calculations. The thermal video produced stores surface temperature as the pixel value. Monitoring the pixel value at a particular sample location provides us with the temperature profile for each sample as it is heated/cooled. Similarly, the temperature of the temperature-controlled plate is simultaneously monitored by analyzing pixels surrounding each sample. By plotting the sample and plate temperatures it is possible to locate an inflection point in the profile that results from an increase in thermal conductivity as the sample melts, ultimately yielding the melting point. This idea was derived from a paper *Kawakami, Kohsaku. 'Parallel*

thermal analysis technology using an infrared camera for high-throughput evaluation of active pharmaceutical ingredients: A case study of melting point determination.' Aaps Pharmscitech 11.3 (2010): 1202-1205 [34]. A python package *'Musical Robot'* was developed to obtain accurate melting points for multiple samples at once, within a matter of minutes from performing the parallel physical measurement, and at very low-cost. In contrast, standard melting point determination techniques utilize equipment that is orders of magnitude more expensive and can take up to an hour to analyze each individual sample.

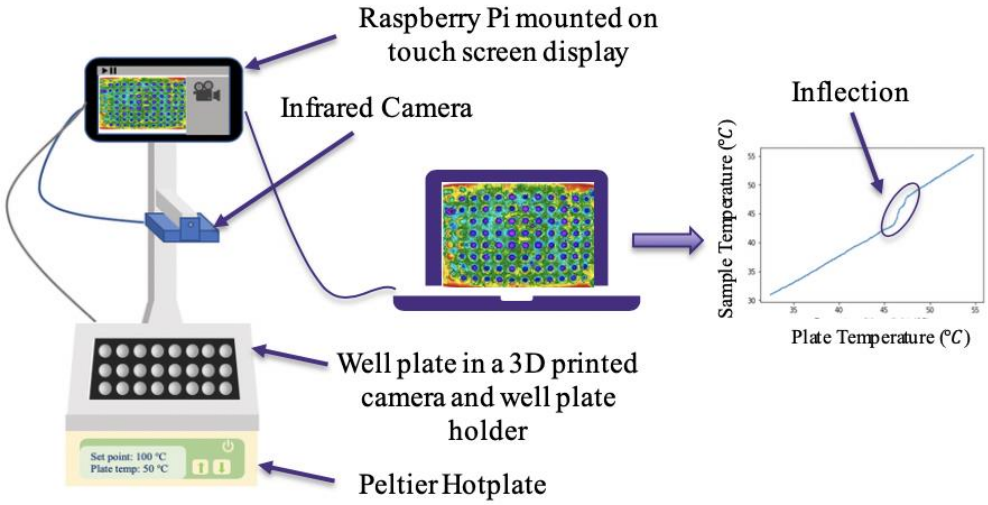


Figure 3.1. Setup for high throughput determination of deep eutectic solvents' melting point

3.2 Materials And Methods

These are the materials used to conduct high throughput experimentation and the methods developed to determine melting points for samples in parallel.

3.2.1 *FLIR Lepton camera*

Lepton® is a complete long-wave infrared (LWIR) camera module designed to interface easily into native mobile-device interfaces and other consumer electronics. It captures infrared radiation input in its nominal response wavelength band (from 8 to 14 microns) and outputs a uniform thermal image with radiometry to provide a temperature image with measurements. Easy to integrate and operate, Lepton is intended for mobile devices as well as any other application requiring very small footprint, very low power, and instant-on operation. Lepton can be operated

in its default mode or configured into other modes through a command and control interface (CCI).

Working Principle

An infrared camera is a non-contact device that detects infrared energy (heat) and converts it into an electronic signal, which is then processed to produce a thermal image on a video monitor and/or to perform temperature calculations. Heat sensed by an infrared camera can be very precisely quantified, or measured, allowing you to not only monitor thermal performance, but also identify and evaluate the relative severity of heat-related problems [35]. Recent innovations, particularly detector technology, the incorporation of built-in visual imaging, automatic functionality, and infrared software development, deliver more cost-effective thermal analysis solutions than ever before.

The FLIR Lepton IR camera was used to record a video of the process of phase change of all the samples on a well plate. The infrared videos recorded contain temperature data at every pixel on the frame in each frame. The temperature data was extracted from the video frames to obtain a temperature profile of all the samples on the well plate.

Configuration of camera modes and features

The camera has various features and modes that were configured to obtain consistent, clean and accurate data.

1. FFC States

Lepton is factory calibrated to produce an output image that is highly uniform when viewing a uniform-temperature scene. However, drift effects over long periods of time degrade uniformity, resulting in imagery which appears grainier and/or blotchy. Columns and other pixel combinations may drift as a group. These drift effects may occur even while the camera is powered off. Operation over a wide temperature range (for example, powering on at -10 °C and heating to 65 °C without performing an FFC) will also have a detrimental effect on image quality and radiometric accuracy. For scenarios in which there is ample scene movement, such as most handheld applications, Lepton is capable of automatically compensating for drift effects using an internal algorithm called scene-based non-uniformity correction (scene-based NUC or SBNUC). However, for use cases in which the scene is essentially stationary, such as fixed-mount applications, scene-based NUC is less effective. In stationary applications and those which need

highest quality or quickly available video, it is recommended to periodically perform a flat-field correction (FFC). FFC is a process whereby the NUC terms applied by the camera's signal processing engine are automatically recalibrated to produce the most optimal image quality. The sensor is briefly exposed to a uniform thermal scene, and the camera updates the NUC terms to ensure uniform output. The entire FFC process takes less than a second.

Lepton provides three different FFC modes:

• External (default for shutter-less configurations)

• Manual

• Automatic (default for configurations with shutter)

All the experiments were recorded by setting the camera in the Automatic FFC mode. In automatic FFC, the Lepton camera will automatically perform FFC under the following conditions:

• At start-up

• After a specified period of time (default of 3 minutes) has elapsed since the last FFC

• If the camera temperature has changed by more than a specified value (default of 1.5 Celsius degrees) since the last FFC[36]

For melting point measurements, the time trigger setting was changed to 30 minutes from 3 minutes. When FFC is performed in automatic mode, the shutter serves as the uniform source of temperature and includes a temperature sensor with automatic input for radiometric measurements. Lepton closes its integral shutter throughout the process which takes up to a second. The radiometry readings are thus not recorded, and this causes a gap in the temperature profile as shown in Figure 3.2. This was obtained by setting the Trigger time to 30s to show the effect of FFC on the experimental data that is obtained.

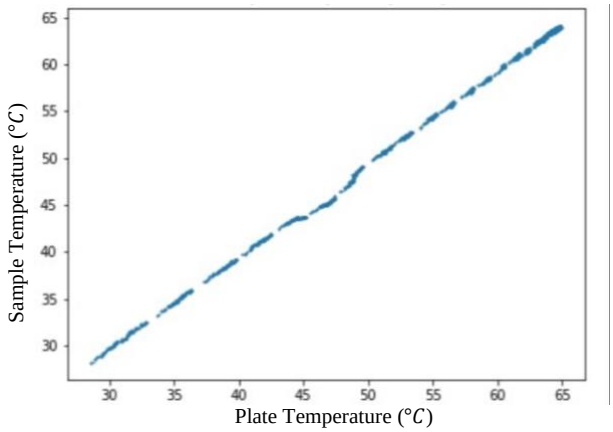


Figure 3.2. Temperature profile obtained with FFC performed every 30 seconds

The effect of changing the FFC trigger time was determined by extracting the temperature profile of the same sample at various trigger times as shown in Figure 3.3 (A) and (B) comparing the results. The experiment was repeated at trigger times of 3, 10, 15, 20, 25 and 30 minutes. The results obtained at various times didn't vary much and showed inflection at the same temperature. Hence, the FFC trigger time was set to 30 minutes.

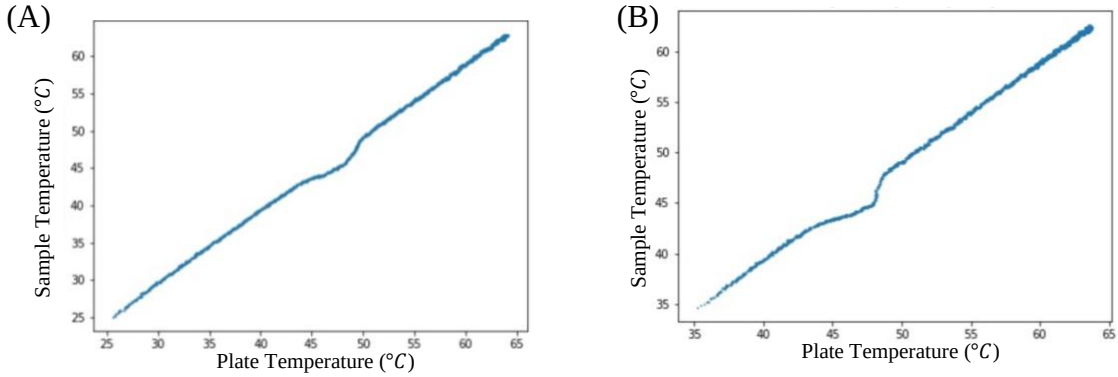


Figure 3.3. (A) Temperature profile performed with FFC performed every 3 minutes; (B) Temperature profile obtained with FFC performed every 30 minutes

2. Gain states

NEDT (noise equivalent differential temperature) is the key figure of merit which is used to qualify *midwave* (MWIR) and *longwave* (LWIR) infrared cameras. It is a signal-to-noise figure which represents the temperature difference which would produce a signal equal to the camera's temporal noise. It therefore represents approximately the minimum temperature difference which the camera can resolve. It is calculated by dividing the temporal noise by the response per degree (responsivity) and is usually expressed in units of *millikelvins*. The value is a function of the camera's f/number, its integration time, and the temperature at which the measurement is made. The Lepton camera can be configured to operate in a high-gain state (the only available state in other versions of Lepton) or a low-gain state. The high gain state provides lower NEDT and lower intra-scene range and the low gain state provides higher NEDT (noise equivalent differential temperature) but achieves higher intra-scene range [37]. The intra scene temperature range is typically -10°C to 140°C in high gain mode and -10°C to 450°C in low gain mode. Since some of the samples might melt at a temperature higher than 140°C, all the experiments were recorded in low gain mode.

3. Radiometry and TLinear

For applications in which temperature measurement is required, radiometry must be enabled to access the related calibration and software features, such as *TLinear*, which support these measurements. In radiometry enabled mode, enabling the corresponding *TLinear* mode changes the pixel output from representing scene flux in 14-bit digital counts to representing scene temperature values in Kelvin (multiplied by a scale factor to include decimals). For example, with *TLinear* mode enabled with a resolution of 0.01, a pixel value of 30000 signifies that the pixel is measuring 26.85°C (300.00K – 273.15K). Thus, all the experiments were recorded with *Radiometry* and *TLinear* enabled [38].

4. Accuracy of the camera

Lepton camera module radiometric accuracy in low gain mode is ±10°C @ 25°C against a 35°C blackbody for a Lepton camera module (using a simple test board with no significant heat sources) at equilibrium and 1” blackbody at 25cm, corrected for emissivity, and in a normal room environment [39].

3.2.2 *Setup*

Camera and well plate mount

The camera and the well plate are both encased in 3D printed mounts. They are both fixed to a T slotted aluminum extrusion to maintain their position relative to each other. The 3D encasing of the well plate acts as an insulator to the T slotted aluminum extrusion and prevents the transfer of heat to the camera.

Peltier hot/cold plate

A Peltier hot/cold plate shown in Figure 3.4 was used so that the samples can be frozen if required before heating them to determine the melting point.

Figure 3.4. Peltier hot/cold plate used to freeze and/or heat the samples on the well plate

Raspberry Pi

A Raspberry Pi 3 Model B was used to control the Peltier plate and the FLIR Lepton IR camera. The camera was controlled using the **Parabilis Thermal** imaging software built for Raspberry Pi [40].

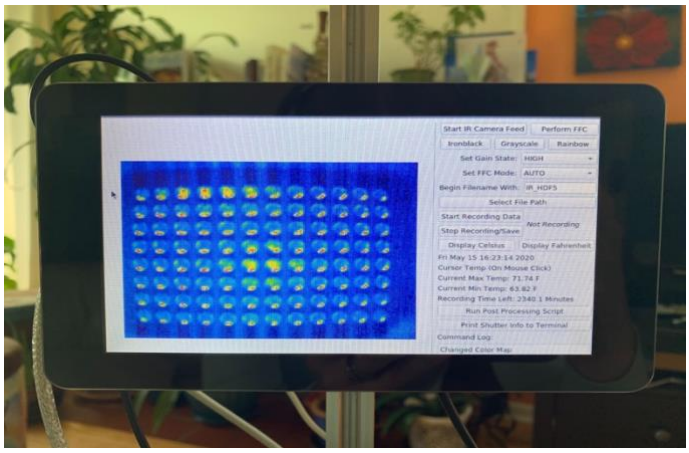


Figure 3.5. The GUI on raspberry pi used to record IR videos

The imaging software outputs the recorded file in the **HDF5** or **TIFF** file format. The acquired data was processed using Python 3 on a different computer.

3.3 Data Extraction: Musical Robot

Musical Robot is a python package developed for high throughput measurement of deep eutectic solvents' melting point using infrared bolometry. The infrared videos which capture the process of melting samples on a well plate in TIFF format are used as input. The package has the following modules to convert the TIFF file into an array of images, crop all the images to include only the well plate, detect and extract temperature profiles of sample and plate locations, and detect the inflection point which is regarded as melting point from the obtained temperature profiles. The following scientific open source python libraries were used majorly to build the musical robot package and visualize the results:

- *Numpy*
- *Pandas*
- *Matplotlib*
- *Scikit-image*
- *SciPy*

3.3.1 *File input*

The TIFF or HDF5 video files are input and converted to a multidimensional array of arrays by using the function *io* from the python library *skimage*. The output array contains arrays equal to the number of frames in the video. Hence, each frame in the video can be accessed and visualized by indexing the array. For example, in python, the array index starts with zero and hence the array representing the first frame in the video can be accessed by using the index 0, second frame by index 1 and so on as shown in the Figure 3.6 (B). The code snippet in the image uses the function *input_file* from the module *edge_detection* which is part of the *musicalrobot* library to input a TIFF file. The videos are also cropped to include only the well plate as shown in Figure 3.6 (A) before they are processed further.

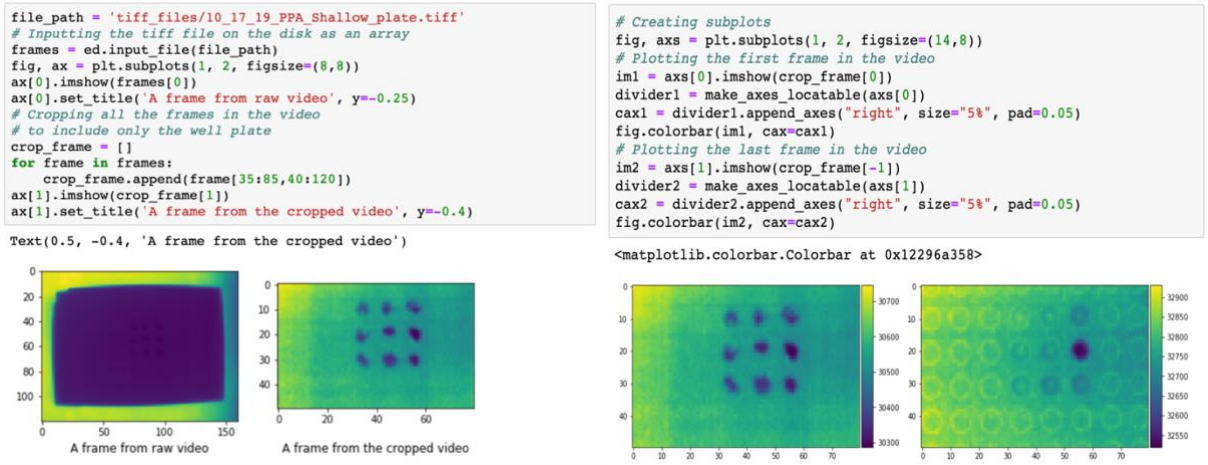


Figure 3.6. (A) Cropping all the frames to retain only the well plate; (B) Visualizing the first and the last frame in the infrared video

3.3.2 *Extracting temperature profiles*

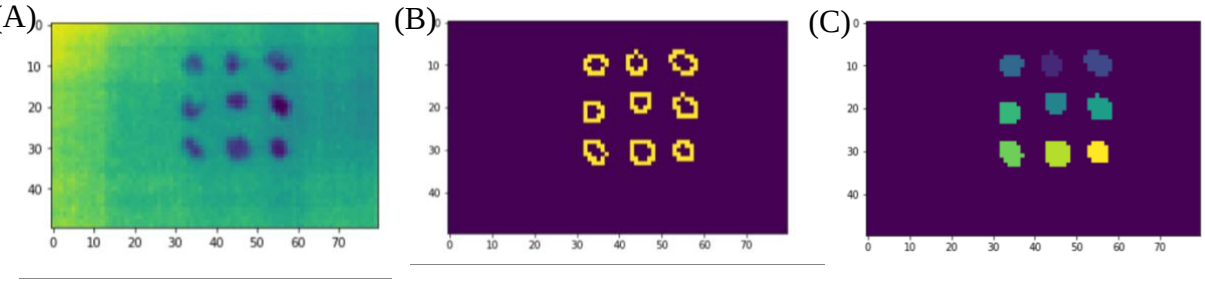
The python package includes the following two modules which use two different techniques to obtain the temperature profile of the samples and well plate locations to determine the melting point of the samples:

- **Temperature profile through edge detection(*musicalrobot.edge_detection*)**

This module can be used for images (video frames) with high contrast and minimal noise as shown in the Figure 3.7 (A) which will allow for detection of edges of just the samples. The temperature profile of the samples and plate is determined by detecting the edges, filling and

labeling them, and monitoring the temperature at their centroids. The edges of the samples are detected as shown in the Figure 3.7 (B) using the function *canny* from the python module *skimage.feature*. The detected sample edges are filled to calculate their mean properties by using the function *binary_fill_holes* from the module *scipy.ndimage.morphology*.

(A) (B) (C)



(D)

unique_index	Row	Column	Plate_temp(cK)	Radius	Plate_coord	Area	Perim	Sample_temp(cK)
75.0	9.0	34.0	30564.0	2.820948	39.820948	25.0	15.656854	30406.0
54.0	21.0	33.0	30536.0	2.705758	38.705758	23.0	14.828427	30366.0
3.0	30.0	34.0	30544.0	2.931615	39.931615	27.0	17.071068	30376.0
84.0	9.0	43.0	30528.0	2.705758	48.705758	23.0	15.071068	30430.0
33.0	18.0	45.0	30534.0	2.705758	50.705758	23.0	14.828427	30350.0
38.0	30.0	45.0	30497.0	3.241022	51.241022	33.0	18.242641	30358.8
29.0	9.0	55.0	30528.0	3.141275	61.141275	31.0	18.485281	30398.0
30.0	19.0	55.0	30526.0	2.985411	60.985411	28.0	17.656854	30293.0
97.0	30.0	55.0	30526.0	2.646284	60.646284	22.0	14.242641	30321.0

Figure 3.7. (A) IR video frame with minimal noise and high contrast between the samples and the plate; (B) Sample edge detection; (C) Filling the sample edges to calculate sample region properties; (D) Dataframe showing all the calculated sample region properties

The filled samples regions as shown in Figure 3.7 (C) are labeled by using the function *label* and the properties of the labeled image regions are measured using the function *regionprops* from the module *skimage.measure*. The function *regionprops* measures properties such as the coordinates of the centroid, mean intensity, area and perimeter of the labeled region. These properties are stored in a *pandas dataframe* as shown in the Figure 3.7 (D)**.** The temperature at each sample region is obtained from the mean pixel intensity as surface temperature is stored as pixel values in infrared videos and images. The properties of all the samples are obtained from each frame in the video. The centroid locations of the samples are used to calculate the plate locations for each sample as shown in the Figure 3.8 (A). The temperature at each of these plate locations is also monitored to plot them against the corresponding sample location. When the sample melts, there

is a sudden increase in the sample temperature due to increase in the sample **thermal conductivity** and **surface area of contact between the plate and the sample** but the temperature of the plate changes steadily. The temperature profile of each sample is plotted against the temperature profile of its corresponding plate location to observe the inflection in sample temperature as shown in the Figure 3.8 (B).

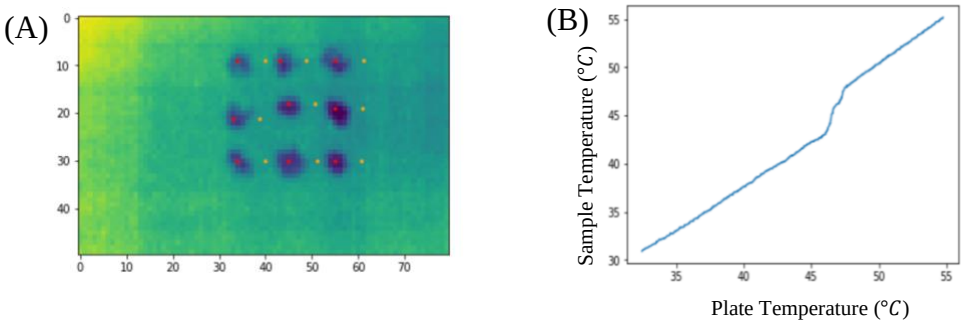


Figure 3.8. (A) Determined sample and plate locations; (B) Inflection in sample temperature observed in the temperature profile

- **Temperature profile through Pixel Analysis**

Sometimes, all the sample edges are not detected precisely, or some samples go undetected as shown in the Figure 3.9 due to various aberrations. In some situations, the contrast between the image and sample may be too low for edge detection, even with contrast enhancement. It could also be due to camera effects like fisheye lens and vignetting. The first module developed (*musicalrobot.edge_detection)* is unfit to use in such situations.

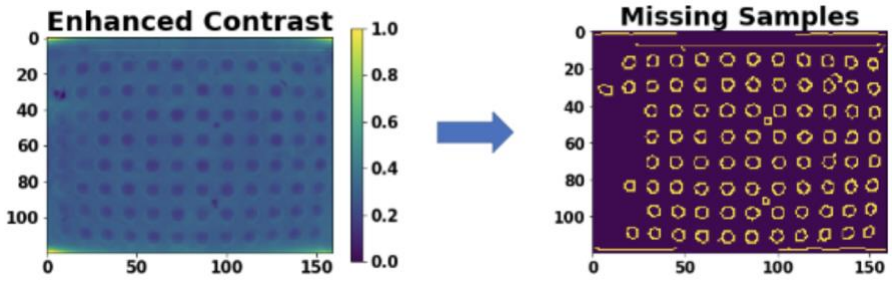


Figure 3.9. Missing edges due to low contrast between the well plate and samples

Thus, a second module was developed which uses a more flexible and reliable technique to detect all the samples on a well plate. It uses pixel analysis as a tool to detect all the samples in the following way. Image equalization is done by adding the corresponding pixel value in all the image frames present in the video and divided by the number of frames. This produces an

equalized image with enhanced contrast from which the sample and plate locations are determined. The location of the samples is determined by summing the pixels over all the rows and over all the columns of the image array. When the summed value of pixels over the rows is plotted, the y-coordinate values of all the rows of wells can be obtained from the troughs of the plot. Similarly, the plot of summed values of pixels over the columns gives the values of x-coordinates of all the columns of wells. This is because the value of pixels representing wells are different as compared to pixels representing the plate. The obtained plots are inverted to convert the troughs into peaks as shown in Figure 3.10. The coordinate value at the peaks is determined by using the *findpeaks* module from the *SciPy* library.

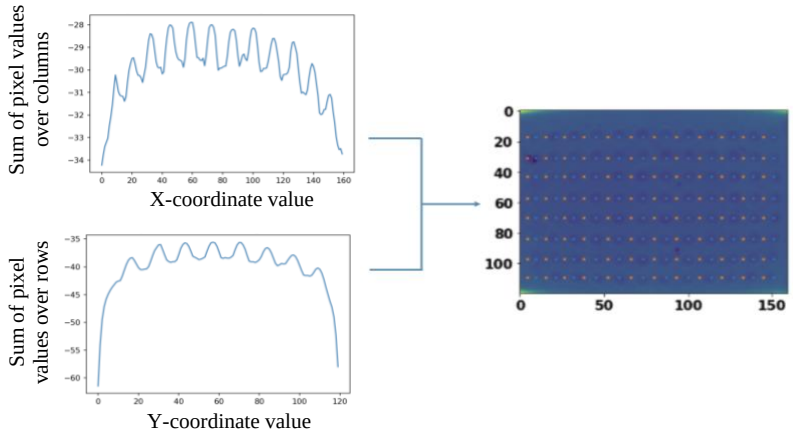


Figure 3.10. The x and y coordinate peaks are used to determine sample and corresponding plate locations

The row and column coordinates are then used to determine the coordinate of centroids of all the samples. The surface temperature at each sample is determined by calculating the mean of pixel values in a circle with centroid of the sample as the center and 1 pixel as the radius. The plate locations are also calculated with respect to the centroid location of the corresponding sample. The temperature profiles at the sample and plate locations are monitored over all the frames in the video and are plotted to determine the melting point of the samples.

3.3.3 *Inflection point determination*

The inflection point is determined by finding the derivative of the curve obtained by plotting sample temperature against plate temperature. The derivate of the curve has a peak at the beginning of the inflection due to the sudden increase in slope and the function *find_peaks* from

the library *scipy.signal* is used to detect the highest peak and the temperature at which the peak occurs is regarded as the melting point.

3.4 DATA CLASSIFICATION

3.4.1 *Need for data classification*

The melting point is determined from the temperature profiles using an analytical approach. The accuracy of this technique relies on the fact that the temperature profile has an inflection and its derivate produces a curve with the highest peak at the inflection point. This was not always the case due to reasons such as:

- Noise in the data caused due to errors in the IR Camera signal processing
- Missing observations in the data
- Temperature profiles with no inflection due to sample behavior

Hence, the data obtained was classified into the following categories using neural networks as shown in the Figure 3.11.

Noisy data: Temperature profile curves with several bumps/peaks with or without an inflection.

Noiseless data: Temperature profile curves with no noise with or without an inflection.

Inflection data: Temperature profile curves with an inflection.

No Inflection data: Temperature profile curves without an inflection.

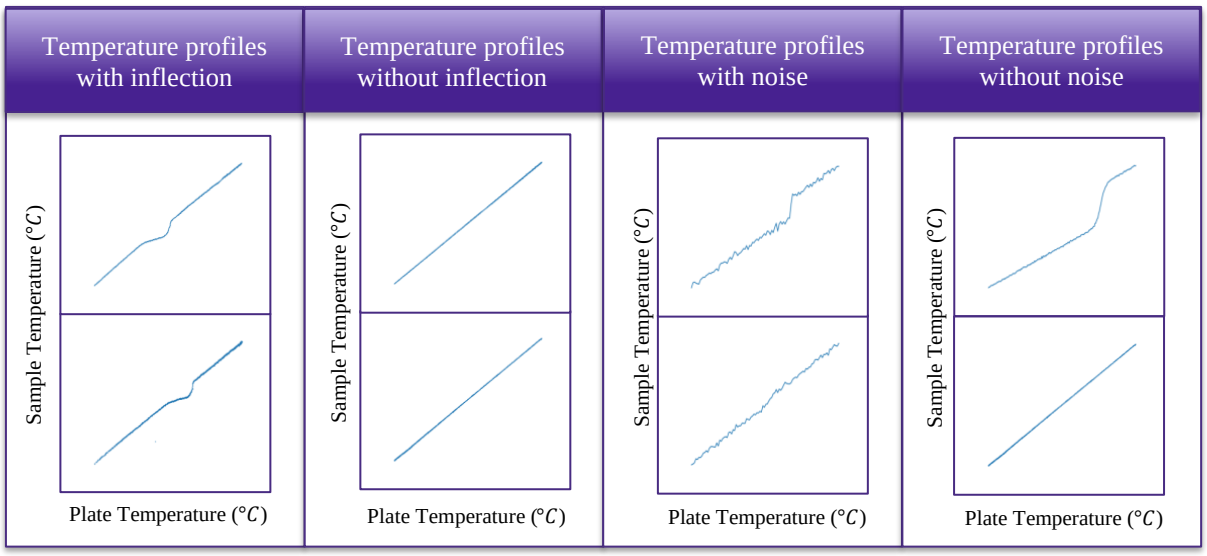


Figure 3.11. Temperature profile curves classified into four categories

The data obtained from experimentation and synthetic generation was manually classified into the above categories and used as a dataset to train neural networks. As mentioned in chapter 2, two types of neural networks namely *Convolutional neural network (CNN)* and *Long Short Term Memory (LSTM) network* were trained to classify the data. The data classification hierarchy followed is shown in the Figure 3.12**.**

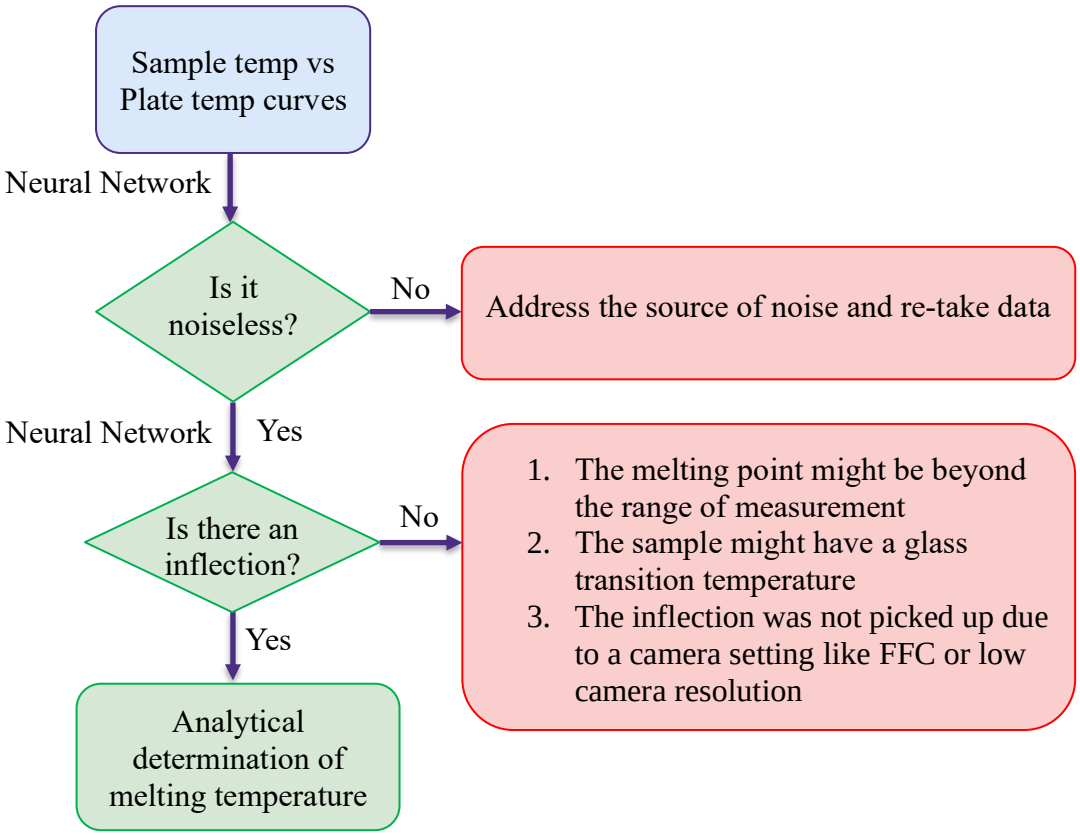


Figure 3.12. Hierarchy of data classification

3.4.2 *Generation of synthetic data*

As discussed in chapter 2, supervised learning neural nets require a vast amount of training data to achieve high accuracies. Since experimentally generating all the data would be tedious and might prove to be the bottleneck in the process of high throughput measurement, synthetic data belonging to each class of data was generated using an analytical approach. The following function was used to generate data.

$$y = \frac{a}{1+e^{-cx}} + dx \tag{3.1}$$

Plots belonging to different categories of data were generated by randomly varying the value of parameters and the range of x values as shown in the Figure 3.13. A random amount of noise was also added to some plots to produce noisy data. Noise was generated using the function *numpy.random.normal* which draws samples from a normal (Gaussian) distribution. It was then scaled to 0.1 times the value of the data point it was applied to.

$$Probability\ density(p(x)) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad \mu: Mean \quad \sigma: Standard\ deviation \qquad (3.2)$$

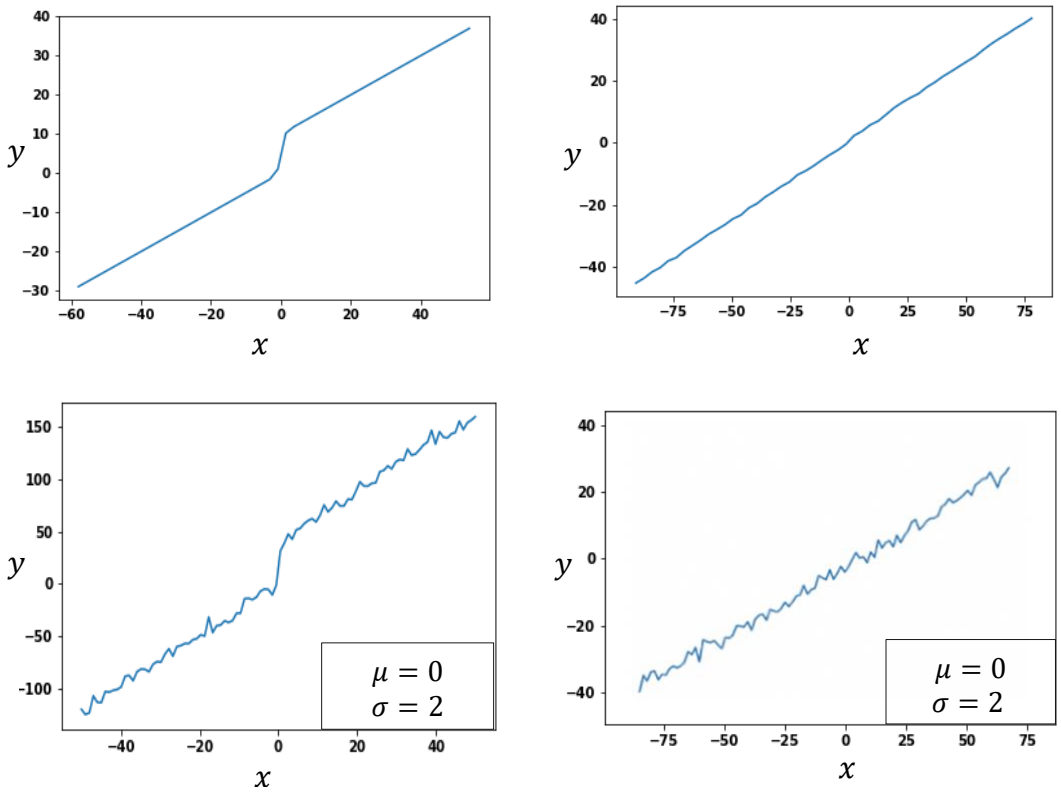


Figure 3.13. Four categories of temperature profiles

Root mean square deviation (RMSD) and Multidimensional scaling (MDS)

As mentioned in chapter 2, *root mean square deviation* and *multidimensional scaling* are performed to model the similarity or dissimilarity in datasets, determine the variance in a dataset and among various datasets in comparison, and visualize the similarity or dissimilarity using a distance metric.

The steps mentioned in chapter 2 were performed to discretize and standardize the datasets before calculating root mean square deviation and performing multidimensional scaling. The evaluation

was performed on the following datasets and the results obtained are visualized in the Figures 3.14**,** 3.15 and 3.16 using histograms and scatterplots:

1. Temperature profiles with an inflection obtained using experimentation
2. Temperature profiles with an inflection obtained by synthetic generation
3. Temperature profiles without an inflection obtained by experimentation
4. Temperature profiles without an inflection obtained by synthetic generation

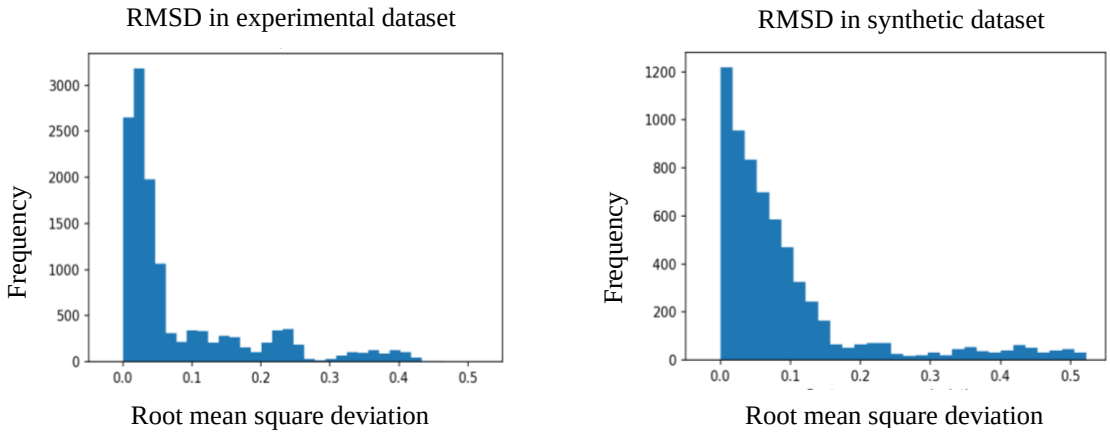


Figure 3.14. Root mean square deviation of data with an inflection

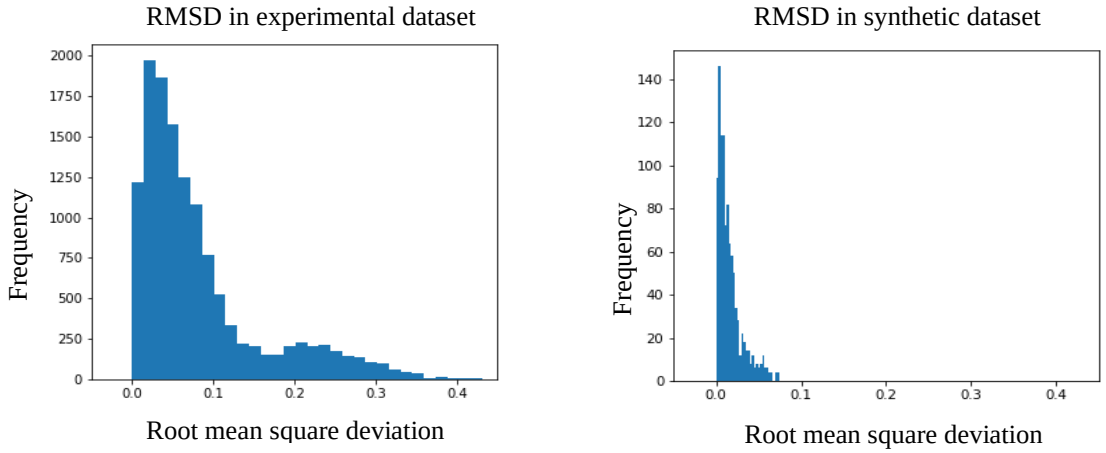


Figure 3.15. Root mean square deviation of data without an inflection

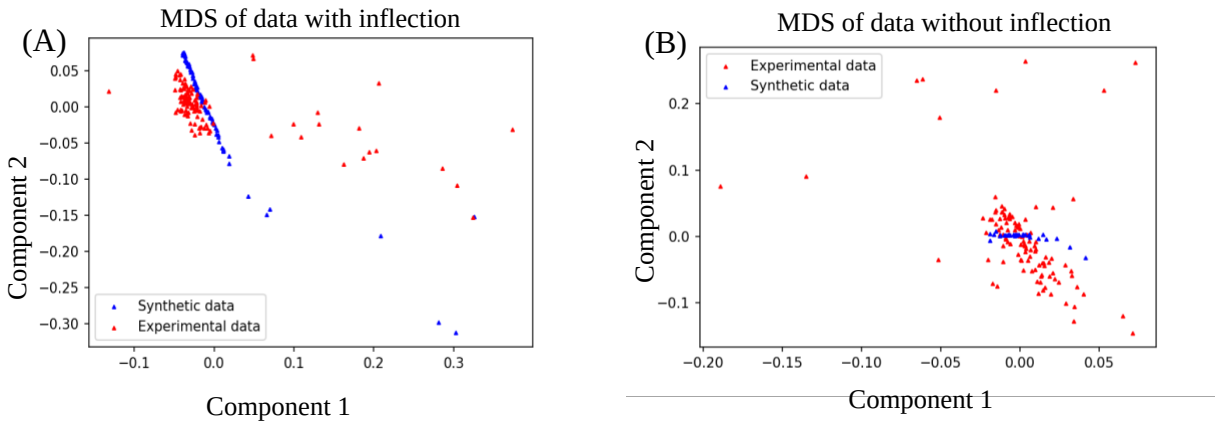


Figure 3.16. (A) Multidimensional scaling of data with inflection; (B) Multidimensional scaling of data without inflection

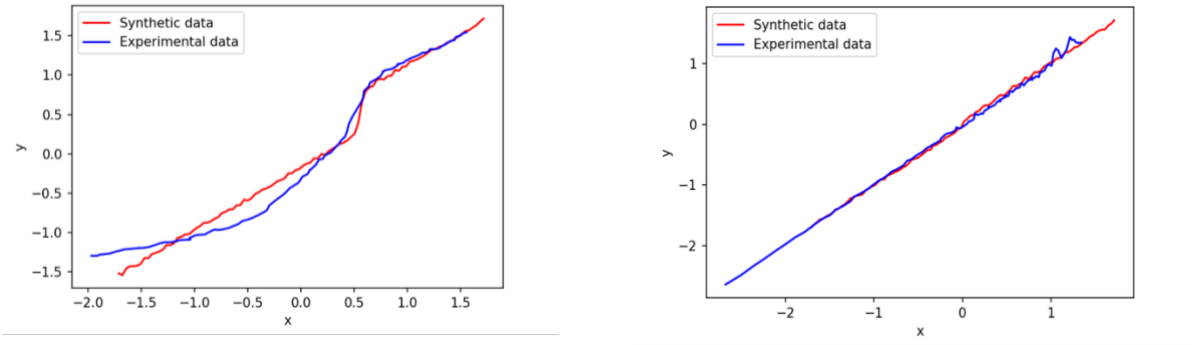


Figure 3.17. Comparing an experimental curve with high MDS Component (>0.1) and a synthetic curve with low MDS Component (<0.1)

In Figure 3.17, the outliers in the experimentation data (MDS Component > 0.1) observed in the Figures 3.16 (A) and (B) are plotted with a synthetic curve (MDS Component < 0.1) to observe the characteristics of the experimentation curves which increase the variance in the dataset. The results obtained show that the variance or dissimilarity obtained in both the experimental and synthetic datasets for both temperature profiles with and without an inflection is quite low and has the same range. Hence, the synthetic data produced was regarded similar to experimental data and was used to train the neural networks.

3.5 Data Classification: Results

As shown in Figure 3.12, data classification was performed to eliminate all the temperature profiles with noise and without an inflection. The noisy temperature profiles were identified by

training a convolution neural network with noisy and noiseless images. The noiseless temperature profiles were classified as plots with and without an inflection using two types of neural networks: Convolutional neural networks and long short term networks. The results obtained from all the neural networks are discussed below.

3.5.1 *Noise Net: CNN*

A convolutional neural network was used to classify the temperature profile as noisy or noiseless. Grayscale images of size (150, 150) of the temperature profiles devoid of auxiliary plot data such as axes, axes labels, and title were used as model input. The existing dataset of images was split into training, validation and testing datasets for all of which the true class labels are known. The weights used for evaluating the performance of the model on the testing dataset are from the epoch with maximum validation accuracy. The training and validation metrics obtained at each training epoch are shown in the Figures 3.18 and 3.19 and Tables 3.2 shows the metrics obtained from the epoch with maximum validation accuracy.

Table 3.1. Performance of noise net on holdout dataset

	Actual : Not noisy	Actual: Noisy
Predicted: Not noisy	36	3
Predicted: Noisy	0	153

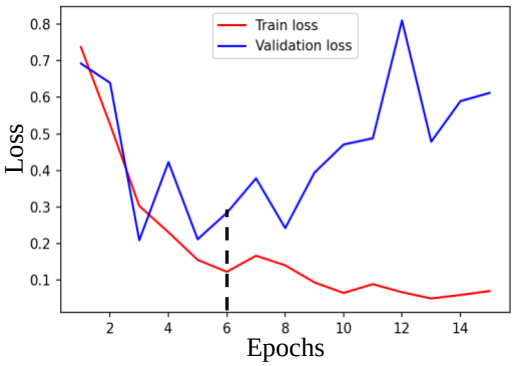


Figure 3.18. Training and validation loss over epochs

Table 3.2. Performance metrics of noise net

	Binary Accuracy	Binary Loss
Train	0.9647	0.1226
Validation	0.9356	0.2840

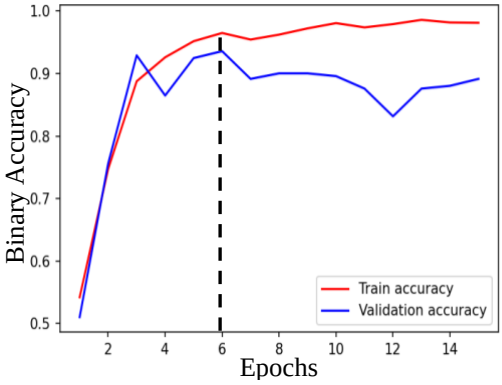


Figure 3.19. Training and validation accuracy over epochs

The confusion matrix shown in Table 3.1 describes the performance of the model on the testing dataset. The metrics and the confusion matrix indicate that the neural network is efficient for screening temperature profiles with noise. The three false positives (noisy plots classified as not noisy) are shown in the Figure 3.20. Two of these plots were manually labelled as noisy because they were obtained from empty wells, but they do not have as much noise as compared to other noisy plots used to train the model. The third false positive is a unique plot and the training dataset does not contain any plot similar to this. Hence, it can be inferred that the trained model is proficient at correctly classifying plots similar to the ones in its training dataset and more training can further lower the error.

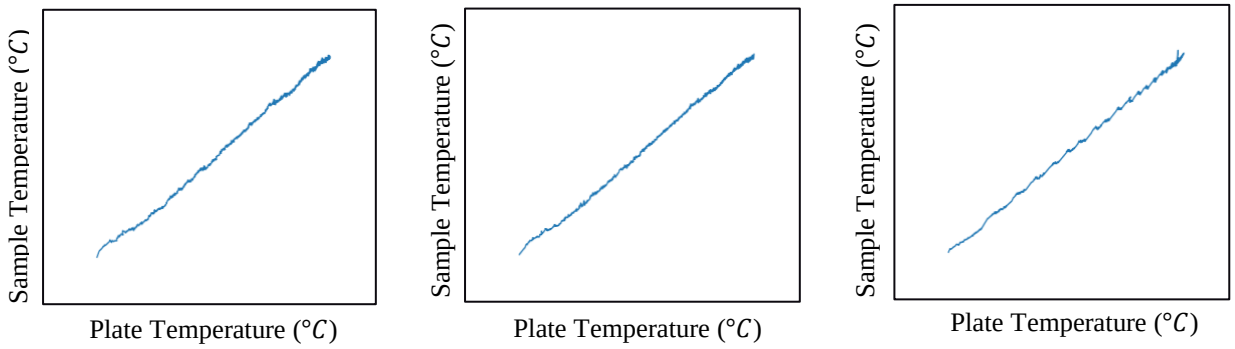


Figure 3.20. Temperature profiles incorrectly classified as noiseless by the noise net

3.5.2 *Inflection net: CNN*

A convolutional neural network was used to classify noiseless temperature profiles as plots with and without an inflection. The image arrays input to the model are in the RGB format and thus contain three channels. The size of the input arrays is (3, 150, 150). The first channel of the image array contains temperature profile data. The second channel of the image array contain the derivate of the temperature profile and the third channel contains an empty white image. Examples of the input images are shown in Figure 3.21**.** It can be seen that the temperature profile with an inflection has a discernible peak in its derivative.

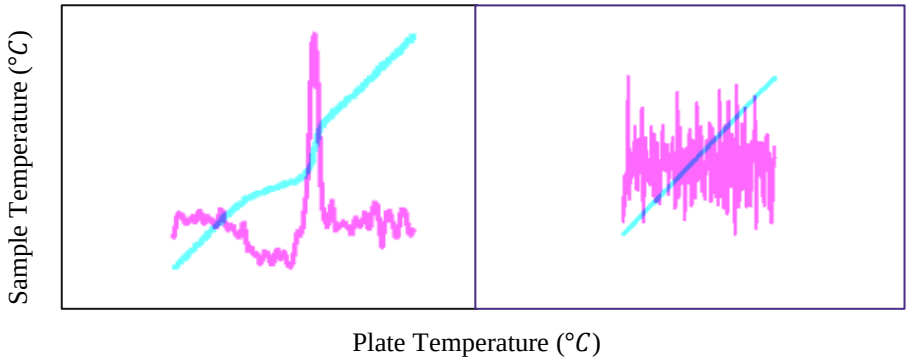


Figure 3.21. Examples of the input provided to the inflection net

The existing dataset of images was split into training, validation and testing datasets for all of which the true class labels are known. The weights used for evaluating the performance of the model on the testing dataset are from the epoch with maximum validation accuracy. The training and validation metrics obtained at each training epoch are shown in the Figures 3.22 and 3.23and

the Table 3.3 shows the metrics obtained from the epoch with maximum validation accuracy. The confusion matrix in Table 3.4 describes the performance of the model on the testing dataset.

Table 3.3. Performance metrics of the inflection net

	Binary Accuracy	Binary Loss
Train	0.8820	0.2838
Validation	0.8808	0.5260

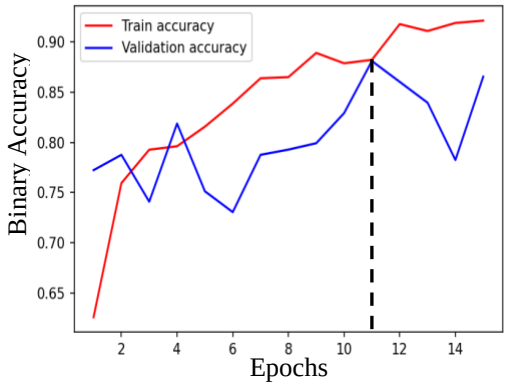


Figure 3.22. Train and validation accuracy over epochs

Table 3.4. Performance of the inflection net on the holdout dataset

	Actual : Inflection	Actual : No inflection
Predicted: Inflection	50	1
Predicted: No inflection	1	45

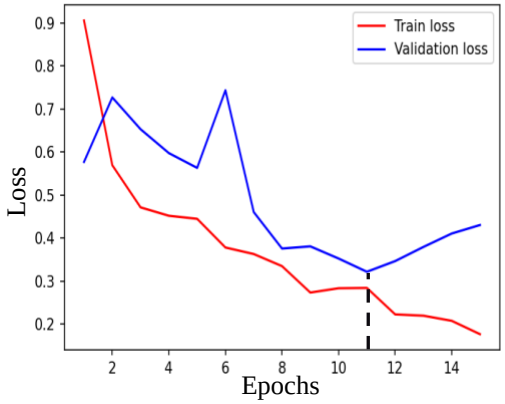


Figure 3.23. Train and validation loss over epochs

Although the performance metrics are not very high, the confusion matrix has only one false negative and one false positive. The Figure 3.24 shows the plots incorrectly classified by the model. The false negative has an inflection at which the slope decreases as opposed to the usual increase. This gives rise to a negative peak in the derivative curve which led to the model classifying it as a plot with no inflection. The temperature profile in the false positive has no inflection, but the derivative has a clear peak which led to the model classifying it as a plot with an inflection. The peak might be due to inconsistencies in data not visible in the temperature profile plot. These results again indicate that the model is proficient in classifying plots similar to

the plots in the training dataset and training it with more datasets consisting of different relevant features can lower the error.

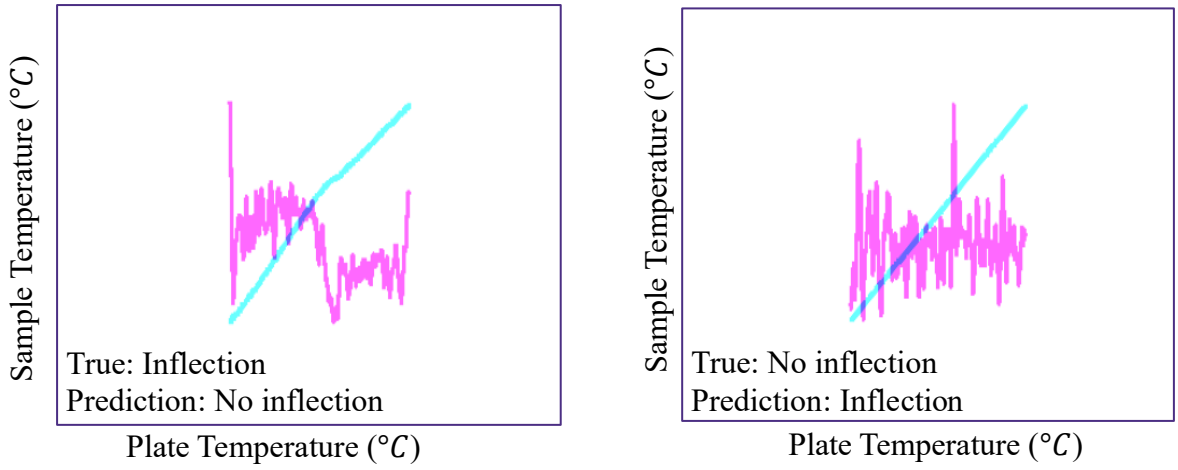


Figure 3.24. Incorrect predictions made by the inflection net

3.5.3 *Inflection net: LSTM*

The sample temperature and the plate temperature of each temperature profile were input as a 3-dimensional *numpy* array to the LSTM model as shown in the Figure 3.25**.**

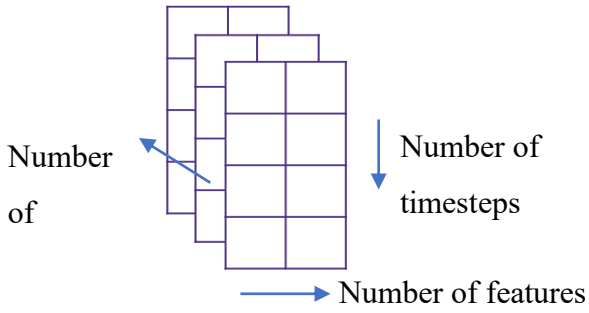


Figure 3.25. Input array shape for the LSTM inflection net

The existing dataset was split exactly like it was for the convolutional neural network. The training and validation metrics obtained at each training epoch are shown in the Figures 3.26 and 3.27 and the Table 3.5 shows the metrics obtained after training the network. The confusion matrix in Table 3.6 describes the performance of the models on the testing dataset.

Table 3.6. Performance metrics of the LSTM inflection net

	Binary Accuracy	Binary Loss
Train	0.8337	0.4289
Validation	0.8318	0.3891

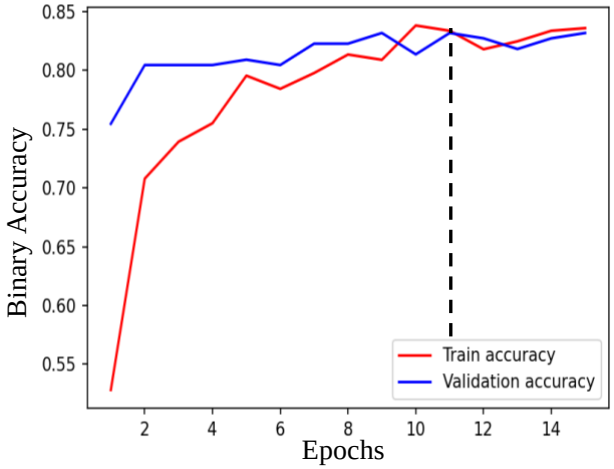


Figure 3.26. Training and validation accuracy over epochs

Table 3.5. Performance of the LSTM inflection net on the holdout dataset

	Actual : Inflection	Actual: No inflection
Predicted: Inflection	100	13
Predicted: No inflection	0	71

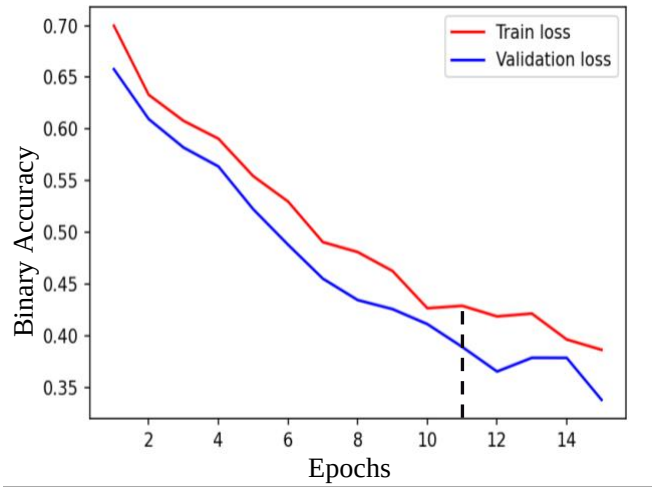


Figure 3.27. Training and validation loss over

The performance metrics and the confusion matrix indicate that the model needs more training. The performance of the model on the test dataset is also poor when compared to the CNN model. Hence the **CNN model** was used to classify the plots with and without an inflection.

3.6 CONCLUSION

- CNNs and LSTMs are both algorithms that can be used to effectively classify images and time series sequential data respectively.
- Since both the CNN and LSTM models are trained and tested on the same data, it can be inferred that the CNN model performed better than the LSTM model.
- Assessment of the performance metrics and the confusion matrix obtained from all the models denotes that the models need to be trained with an enormous volume of data consisting of relevant features.

Chapter 4. HIGH THROUGHPUT ELECTROCHEMICAL TESTING SYSTEM

Redox Flow Batteries (RFBs) are an economical and reliable alternative to large scale grid storage of energy due to their inherent scaling advantages. Increasing the storage capacity and power density of a system can be achieved both independently and with relative ease, just by increasing the volume of anolyte and catholyte stored in the external tanks for the former and increasing the length of the power stack for the latter [41]. Vanadium flow batteries are the most commercially used batteries as they exploit the ability of vanadium to exist in four different oxidation states in solution thus requiring only one electroactive element, instead of two, to complete the battery. Although vanadium flow batteries have many inherent benefits, vanadium is an expensive metal and it drives up the overall cost of the system and hence more viable materials are required for large scale deployment. The majority of the redox flow batteries use either aqueous or non-aqueous solvents as electrolyte media. Electrolytes based on non-aqueous solvents have a wider electrochemical window and can therefore support higher voltage, and therefore greater energy density, as compared to aqueous chemistries, but non-aqueous solvents are expensive when compared to water, have low solubility of ionic species and have potential safety concerns. Deep Eutectic Solvents (DESs) are promising alternative solvents for RFBs due to their low cost, molecular versatility, ease of synthesis and environmentally benign qualities. They are also known to be great solvents and thus promising electrolytes for salt-based and organic redox flow batteries.

The design space for deep eutectic solvents is vast and high throughput electrochemical testing has to be employed to select the optimum DES for use in redox flow batteries. Cyclic voltammetry is the most widely used experiment to determine the redox electrochemical properties of materials such as electrochemical potential window, peak cathodic and anodic currents and electrochemical cyclic stability. This chapter describes the high throughput

electrochemical testing system setup to conduct cyclic voltammetry and classify solvents as having reversible or irreversible cyclic voltammetry curves.

4.1 Introduction To Cyclic Voltammetry

Cyclic voltammetry (CV) is a powerful and popular electrochemical technique commonly employed to investigate the reduction and oxidation processes of molecular species. CV is also invaluable to study electron transfer-initiated chemical reactions. In a cyclic voltammetry experiment, voltage is applied to the working electrode and the current response is plotted as a function of the applied voltage. The applied voltage is scanned linearly from an initial value (V_i) to a predetermined value (V_1) known as the switching potential after which the direction of the scan is reversed. The scan can then be halted at the required value of voltage (V_f) or can be cycled between two preselected values of voltage [42]. A typical voltage signal applied to an electrochemical response and the current response obtained from a redox material are shown in Figures 4.1 (A) and 4.1 (B)**.** Consider ta reversible electron transfer reaction where $\boldsymbol{O}$ is the oxidized form and $\boldsymbol{R}$ is the reduced form of the analyte as shown below:

$$O + e^- \rightleftharpoons R \tag{4.1}$$

The equilibrium between $\boldsymbol{O}$ and $\boldsymbol{R}$ is described by the Nernst equation. The Nernst equation relates the potential of an electrochemical cell ($\boldsymbol{E}$) with the standard potential of the species (E^0) and the relative activities or concentrations of the oxidized (O) or reduced (R) analyte.

$$E = E^0 + \frac{RT}{nF} \ln \frac{(O)}{(R)} \tag{4.2}$$

In the equation, F is Faraday's constant, R is the universal gas constant, n is the number of electrons, and T is the temperature. As the potential is scanned in the negative direction the current rises to a peak and then decays in a regular manner. The current depends on two steps in the overall process, the movement of electroactive material to the surface of the electrode and the electron transfer reaction. The electrolysis of the reactant depletes its concentration near the surface. Since the experiment is performed at a stationary electrode in an unstirred solution, diffusion is the principal means of moving the reactant to the surface. This relatively slow mode of mass transport cannot maintain a steady-state concentration profile in the region close to the electrode. Therefore, the depletion zone grows. In a sense, the average distance that the reactant molecules must travel to reach the surface increases. Consequently, the rate of mass transport

decreases. The dependence on mass transport, and the fact that a finite rate for the reverse electron transfer process is possible, prevent the current from increasing exponentially with potential. Eventually the mass transport step becomes rate determining and the current reaches a maximum. Since the concentration gradient continues to decrease, the rate of mass transport continues to decrease causing the current to decay. An advantage of the cyclic voltammetry experiment is the fact that a significant concentration of product (in this case, the reduced form) has been generated near the electrode on the forward scan. When the scan direction is reversed, the reduced form is oxidized back to the original starting material and the current for the reverse process is recorded.

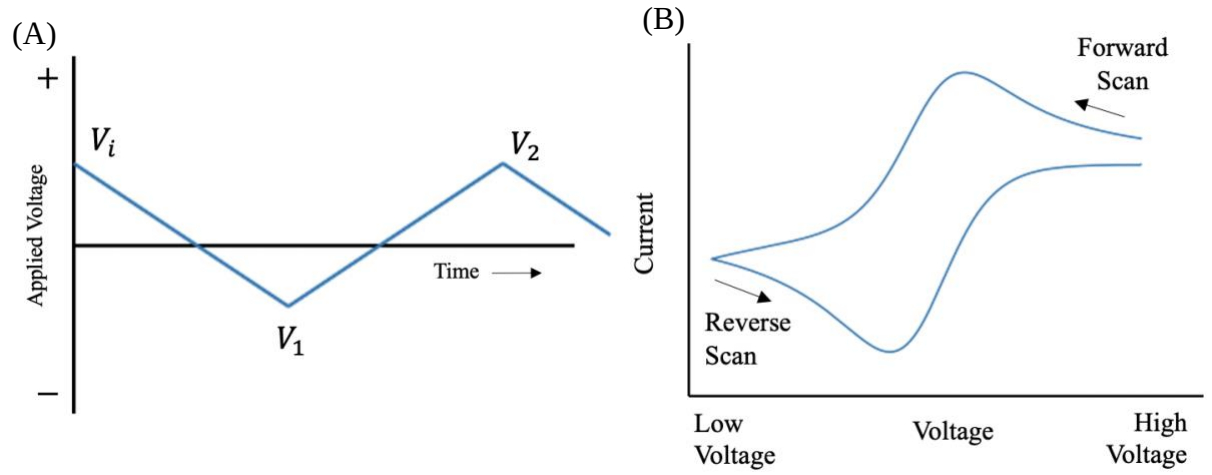


Figure 4.1. (A) A typical voltage signal applied; (B) Current response obtained for the applied voltage signal

4.2 High Throughput Setup To Conduct Cyclic Voltammetry

High throughput measurement of CV data was performed using a screen-printed electrochemical array formed by 96 three-electrode electrochemical cells with carbon-based working electrodes (MetrOhm). This electrochemical array is fixed in the bottom of a standard microtiter plate with 96 wells. The well plate is connected to a gamry potentiostat with runs cyclic voltammetry on all the samples in the well plate. The components of the setup are described below:

4.2.1 *Gamry Potentiostat*

A potentiostat is an electric equipment which controls the voltage difference between the working electrode and reference electrode both contained in an electrochemical cell by injecting current into the cell through an Auxiliary, or Counter Electrode. The conventional setup of the potentiostat and the electrochemical cell is shown in Figure 4.2. The potentiostat measures the current flow between the Working and Counter electrodes. The controlled variable in a potentiostat is the cell potential and the measured variable is the cell current. As the potential of the working electrode is increased (positive scan), the analyte is oxidized, and the analyte gets reduced as the potential of the working electrode is decreased (negative scan). A gamry potentiostat was used to conduct the cyclic voltammetry experiments with the help of the *Gamry* graphical user interface (GUI). The parameters of the experiments such as scan rate, initial voltage, switching voltage and final voltage can be easily set using the GUI. It also stores the correct response and outputs a plot of the current response obtained to the applied voltage and a text file with all the run details.

Figure 4.2. Conventional setup used to perform cyclic voltammetry

4.2.2 *Screen printed 96 well electrodes*

The *DropSens Electrochemical ELISA* plate was used to conduct high throughput cyclic voltammetry experiments. It is a screen-printed electrochemical array formed by 96 three-electrode electrochemical cells with carbon-based working electrodes. Each electrochemical cell consists of:

- Working electrode: Carbon (3 mm diameter)

- Auxiliary electrode: Carbon
- Reference electrode: Silver
- Plastic substrate: L7.4 cm x W11 cm x H0.5 mm
- Electric contacts: Gold

This electrochemical array is fixed in the bottom of a standard microtiter plate with 96 wells. The screen printed well plate is connected to the *DRP-96-Well Plate Connector* from *DropSens*. It is a portable instrument powered by a 12 V adaptor and has 2 mm female connectors in the front panel acting as an interface between the Electrochemical ELISA Plate and the gamry potentiostat. In consists 96x3 gold plated retractile pins in the top panel to connect the screen printed well plate to this system. Electrochemical experiments can be run on up to eight wells of the same column at the same time. Also, independent analysis of any well can be developed. Two manual rotating selectors allow to select the row letter (from A to H) and the column number (from 1 to 12) identifiers of the wells to be analyzed. The *Gamry GUI* can be used to set parameters and run electrochemical tests on all the samples in the screen printed well plate. The complete setup in shown in Figure 4.3**.**

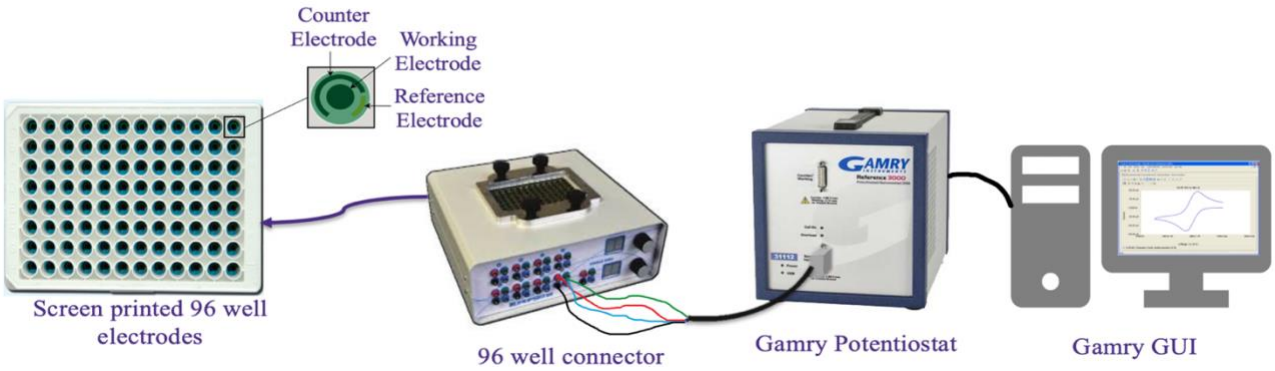


Figure 4.3. Setup used to conduct high throughput cyclic voltammetry

4.3 DATA CLASSIFICATION

High throughput cyclic voltammetry data might differ based on the reduction and oxidation properties of the analyte being tested and the curves can belong to various categories. The analysis of vast number of curves obtained using conventional techniques would take a long

time. Thus curves need to be classified in order to accurately automate the curve analysis and interpretation.

As the potential is scanned in the negative direction the current rises to a peak and then decays in a regular manner. The current depends on two steps in the overall process, the movement of electroactive material to the surface and the electron transfer reaction. Based on the rate of electron transfer and mass transport, the current response varies as shown in Figure 4.4 and the cyclic voltammograms can be classified into the follwing categories [43]:

1. **Electrochemical reversibility**: These are curves obtained when there is a low barrier to electron reactions and are characterized with a peak separation less than 57 mV for a one-electron redox couple.
2. **Electrochemical irreversibility**: These are curves obtained when the rate of electron transfer is slow when compared to the mass transport. Slow electrode kinetics necessitate significantly more negative applied potentials for appreciable current to flow. As a result, peak-to-peak separation is larger than the 57 mV anticipated for an electrochemically reversible one-electron redox couple.
3. **Irreversible chemical reaction**: These are curves obtained when the electron transfer is coupled with a chemical reaction:

$$O + e^- \rightleftharpoons R \tag{4.3}$$

$$R \xrightarrow{k} Z \tag{4.4}$$

A slow homogeneous chemical reaction (compared to the electron transfer) results in a voltammogram that appears reversible, because the amount of **R** that reacts to form **Z** is negligible. As the rate constant (**k**) of the homogeneous reaction increases, the amount of Red consumed in the chemical reaction increases, thereby moving the system from equilibrium and leading to a non-Nernstian response. The ratio of the anodic to cathodic peak currents decreases because the reduced species **R** is consumed by the subsequent chemical reaction, resulting in fewer species to oxidize on the anodic scan. Experimentally, the response can be modified by varying the scan rate. As the scan rate is increased, the time scale of the experiment competes with the time scale of the chemical step, and more **R** is left for reoxidation. For sufficiently fast scan rates, the

electrochemical feature will regain reversibility as oxidation outcompetes the chemical reaction.

4. **Multiple Electron transfers:** Some analytes undergo multiple electron transfers resulting in more than 2 current peaks. Consider an analyte which undergoes two electrochemical steps:

$$O + e^- \rightleftharpoons R \quad (4.5)$$

$$R + e^- \rightleftharpoons Q \quad (4.6)$$

The voltammogram appearance is dependent upon the difference in formal potential of the two electrochemical steps. If the second electrochemical step is thermodynamically more favorable than the first, the voltammogram appears identical to a Nernstian two-electron transfer, and the peak-to-peak separation (ΔE_p) will be 28.5 mV instead of the 57 mV predicted for a one-electron process. As the second electron transfer becomes less thermodynamically favorable, ΔE_p grows until it reaches a maximum value of just over 140 mV; at that point, the wave separates into two resolvable waves, each with ΔE_p = 57 mV.

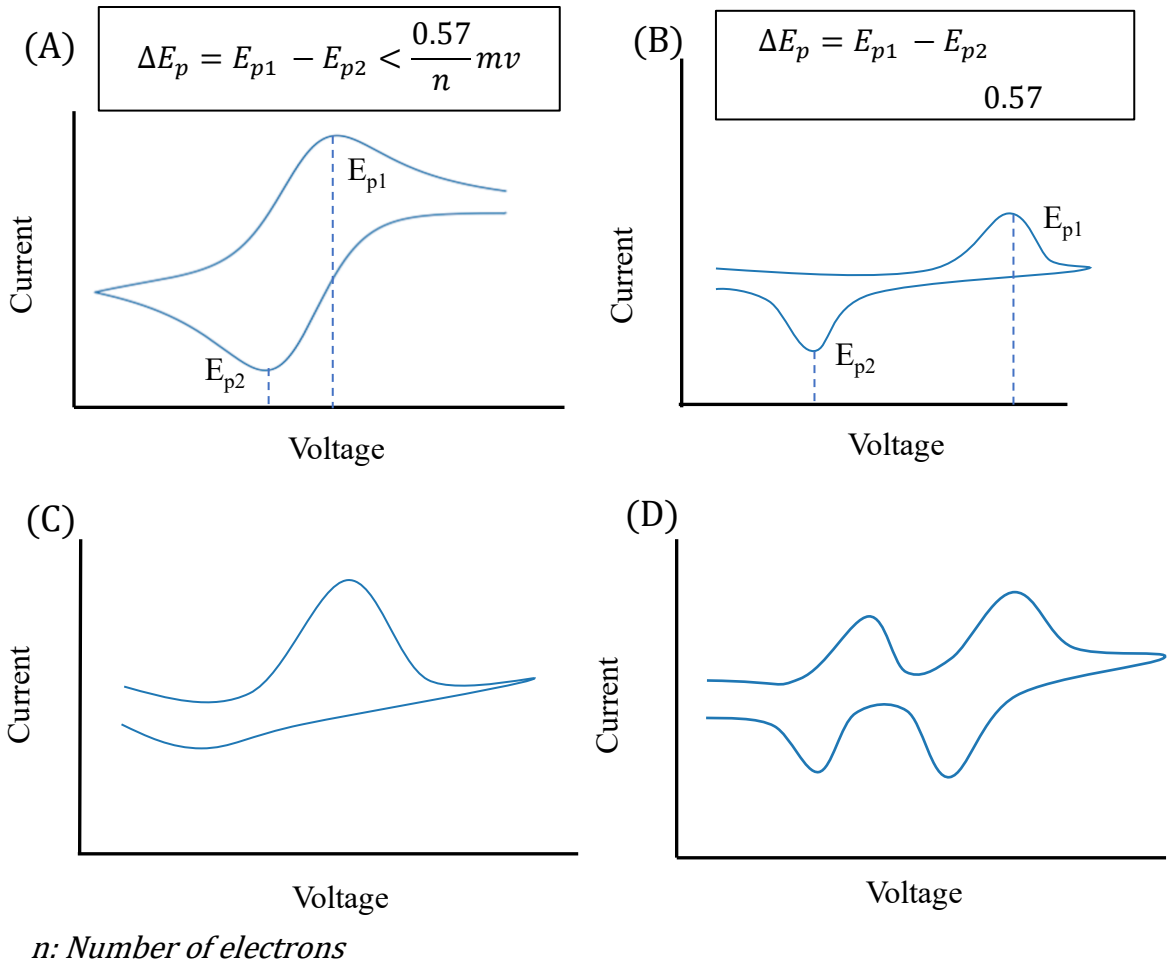


Figure 4.4. Different categories of cyclic voltammograms (A) Reversible one electron transfer; (B) Irreversible one electron transfer; (C) Irreversible chemical reaction; (D) Multiple electron transfer voltammogram

The experimental and synthetic data was manually labelled and classified into chemically reversible and irreversible voltammograms. As mentioned in chapter 2, two types of neural networks namely *Convolutional neural network (CNN)* and *Long Short Term Memory (LSTM) network* were trained to classify the data**.** The hierarchy of data classification is shown in Figure 4.5. All the voltammograms classified as chemically reversible by the neural network models were analytically classified as reversible electron transfer and irreversible electron transfer voltammograms by determining the difference in anodic and cathodic peak potentials.

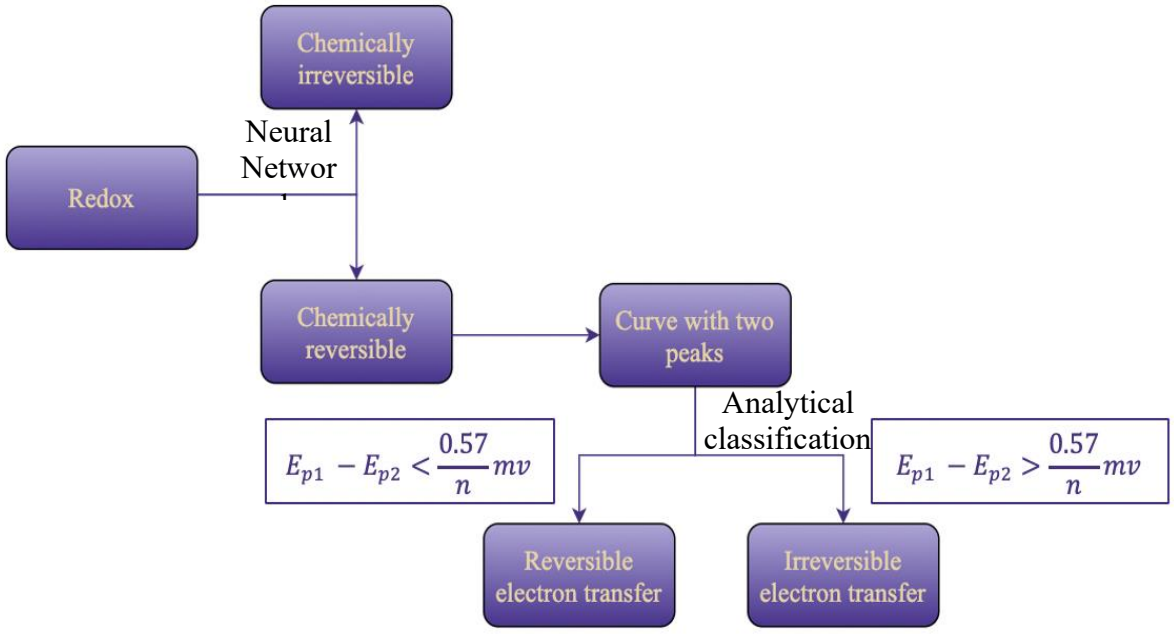


Figure 4.5. Hierarchy of data classification

4.4 Generation of Synthetic Data

Synthetic chemically reversible and irreversible voltammograms were generated using a simulation [44]. This cyclic voltammetry simulation couples a one-electron electrochemical reaction with a subsequent chemical reaction of the reduced species, as below:

$$0 + e^- \underset{k_b}{\overset{k_f}{\rightleftharpoons}} R \xrightarrow{k_c} \mathrm{Z} \tag{4.7}$$

This simulation assumes an aqueous electrochemical system. A metal electrode in solution supplies and extracts electrons from aqueous species O and R, such as in the case of ferro-/ferricyanide. In this simulation, three mechanistic processes are modeled: the electrochemical reaction (charge transfer), the chemical reaction, and diffusion. The interplay between the electrochemical reaction at the surface, the chemical reaction of the reduced product R, and the diffusion of both O and R yield the cyclic voltammetry result. The dimensionless reversibility parameter (Λ) in the simulation defined as the rate of charge transport to the rate of mass transport governs the mechanism of the electrochemical reaction. Electrochemically reversible reactions occur when the rate of charge transport to mass transport (Λ) is high and the inverse is true for electrochemically irreversible reactions. Chemically irreversible reactions occur when

the rate constant of chemical reaction (k_c) is significantly higher than rate constant of electron transfer (k_b).

The following ranges are used to generate curves with varying chemical reversibility as shown in Figure 4.6 (A):

Chemically reversible: $k_c < 10^{-3}s^{-1}$

Chemically irreversible: $k_c > 10^{-2}s^{-1}$

The following range of Λ are used to generate curves with electrochemical reversibility as shown in Figure 4.7(b):

Reversible electron transfer: $\Lambda > 1$

Irreversible electron transfer: $\Lambda < 1$

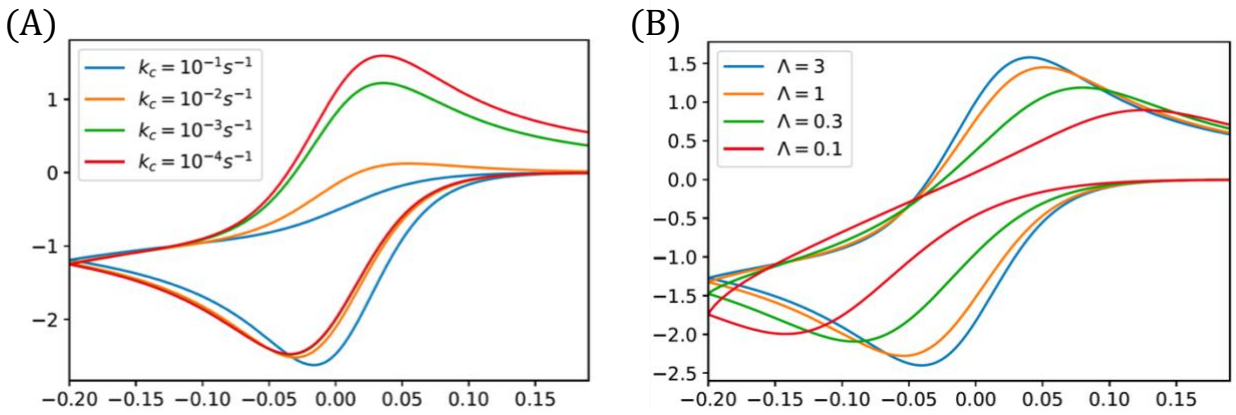


Figure 4.6. (A) Voltammograms moving towards chemical irreversibility as rate constant of chemical reaction (k_c) is increased at constant rate of electron transfer (k_b); (B) Peak-to-peak separation increases leading to electrochemically irreversible voltammograms as the dimensionless parameter (Λ) is decreased.

4.5 Root Mean Square Deviation and Multidimensional Scaling

Root mean square deviation (RMSD) and multidimensional scaling (MDS) were performed on both the experimental data and synthetic data. These techniques enable us to determine the variance in the datasets by modeling the similarity or dissimilarity in data as distance in

geometric space. These techniques are also useful to visualize the variance in a dataset and compare it to the variance of a different dataset.

Root mean square deviation and multidimensional scaling was performed on the following datasets:

1. Chemically reversible data obtained from high throughput experimentation
2. Chemically irreversible data obtained from high throughput experimentation
3. Synthetic chemically reversible data
4. Synthetic chemically irreversible data

Data Treatment

Before modeling the dissimilarity between the datasets, offset correction was performed on all the reversible curves in addition to data reduction and normalizing to establish compatibility between all of them. The positive peak of all the curves were aligned with each other by moving the positive peak to 0 on the x-axis.

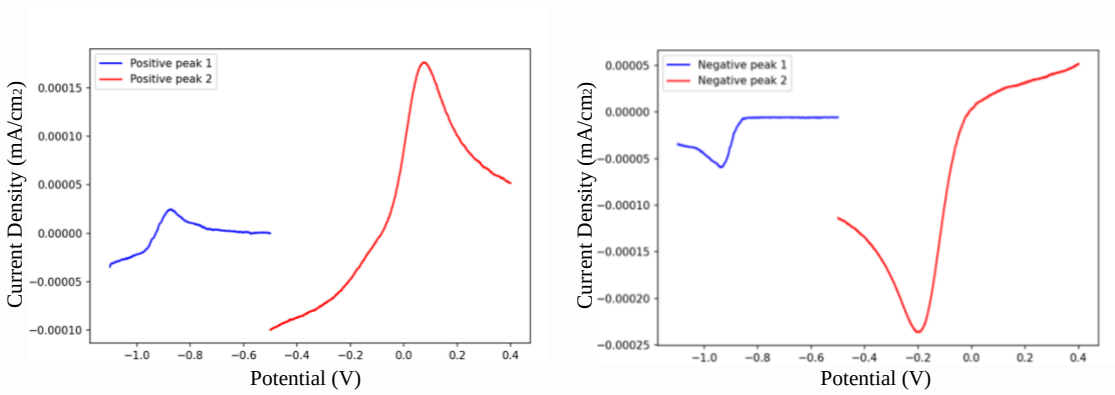


Figure 4.7. Comparing two curves before normalization

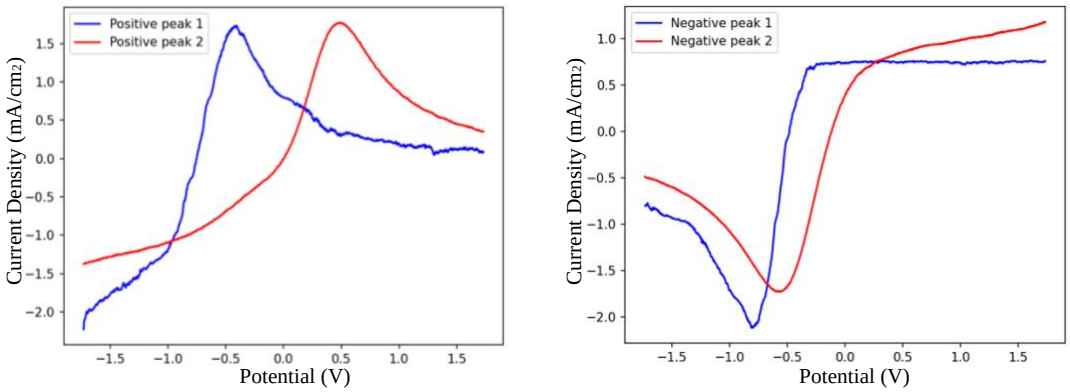


Figure 4.8. Comparing two curves after normalization

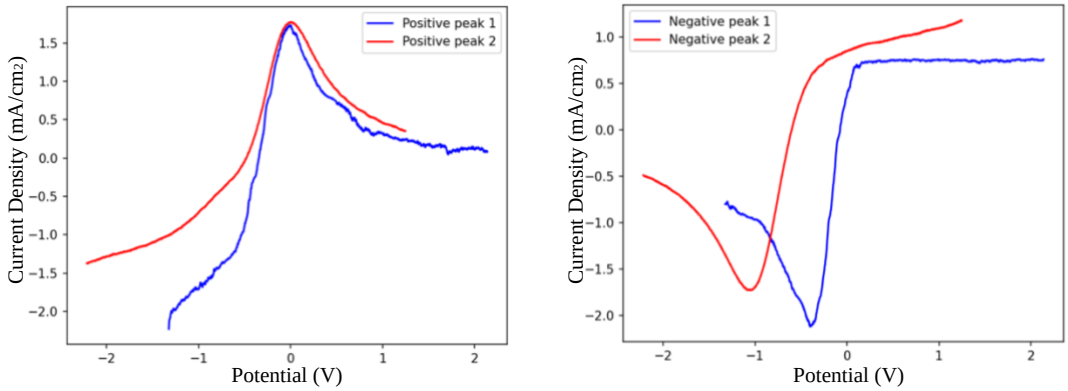


Figure 4.9. Curves after offset correction

The results obtained were visualized using histograms and scatter plots.

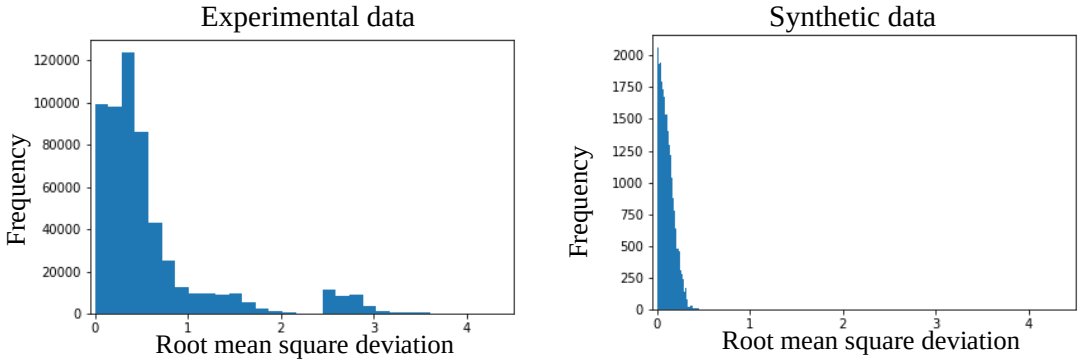


Figure 4.10. Root mean square deviation of reversible cyclic voltammograms

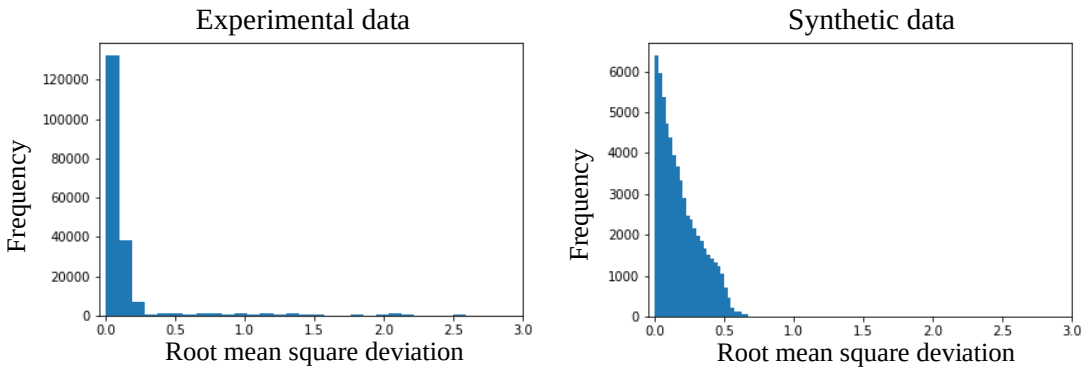


Figure 4.11. Root mean square deviation of irreversible cyclic voltammograms

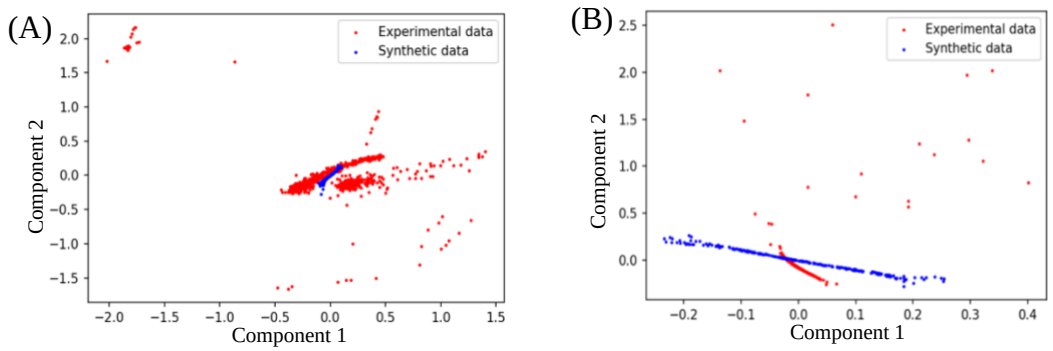


Figure 4.12. (A) Multidimensional scaling of reversible cyclic voltammograms; (B) Multidimensional scaling of irreversible voltammograms

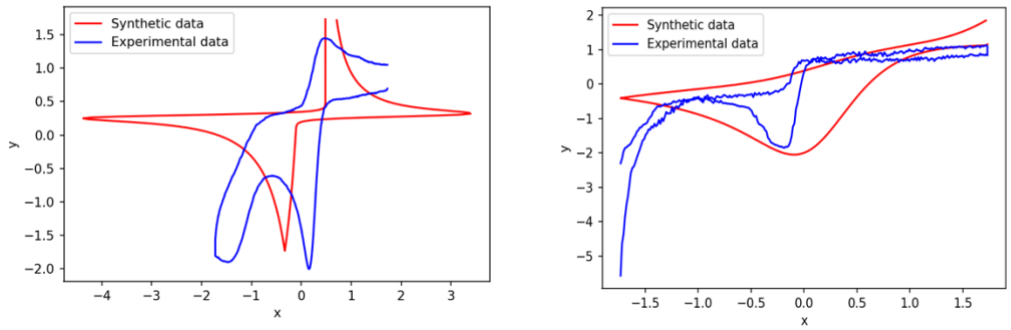


Figure 4.13. Comparing an experimental curve with high MDS Component (> 1) and a synthetic curve with low MDS Component (< 1)

In Figure 4.13, the outliers in the experimentation data (MDS Component > 1) observed in the Figures 4.12 (A) and (B) are plotted with a synthetic curve (MDS Component < 1) to observe the characteristics of the experimentation curves which increase the variance in the dataset.

The results obtained show that the variance or dissimilarity obtained in both the experimental and synthetic datasets for both reversible and irreversible cyclic voltammograms lies in a similar range. Hence, the synthetic data produced was regarded similar to experimental data and was used to train the neural networks.

4.6 DATA CLASSIFICATION: RESULTS

As mentioned in chapter 2, data classification was performed to classify cyclic voltammograms as chemically reversible or chemically irreversible using two types of neural nets: Convolutional neural networks (CNNs) and Long short term memory networks (LSTMs). The results obtained from implementing the CNN and LSTM architecture shown in Figure 2.6 with cyclic voltammetry data is given below.

4.6.1 *Convolutional neural network*

Grayscale images of the cyclic voltammograms of size (150, 150) and type *numpy 2-dimensional arrays* were used as input. The input images are devoid of auxiliary data such as axes, axes labels and, image titles and contain only the cyclic voltammogram. The existing dataset of images was split into training, validation and testing datasets for all of which the true class labels are known. The training and validation metrics obtained at each training epoch are shown in Figures 4.14 and 4.15. The Tables 4.1 and 4.2 show the metrics obtained after training the network. The confusion matrix describes the performance of the models on the testing dataset.

Table 4.1. Performance metrics of CNN

	Binary Accuracy	Binary Loss
Train	0.9969	0.0160
Validation	0.9823	0.1938

Table 4.2. Performance of the CNN on the holdout dataset

	Actual : Reversible	Actual: Irreversible
Predicted: Reversible	170	3
Predicted: Irreversible	10	88

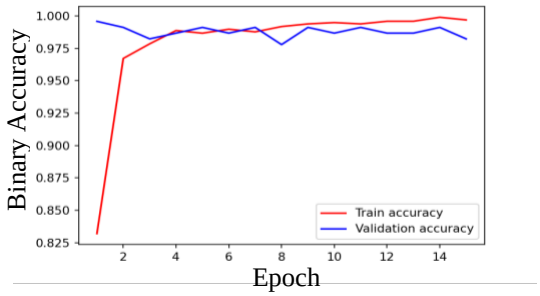


Figure 4.14. Training and validation accuracy over epochs

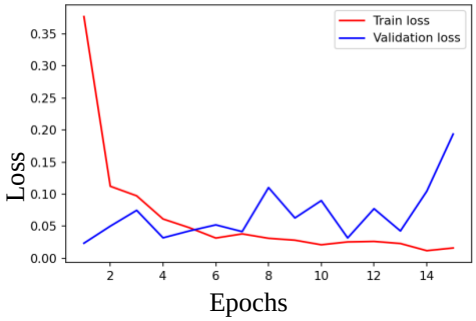


Figure 4.15. Training and validation loss over epochs

The assessment of the confusion matrix indicates that the model needs to be trained with more data consisting of different relevant features that the model might encounter. The cyclic voltammograms are being evaluated by the neural network to determine the reversible curves. Hence, the false positives are irreversible curves classified as reversible and false negatives are reversible curves classified as irreversible by the network. Some examples of incorrectly classified cyclic voltammograms are shown in Figures 4.16 and 4.17. The false positives and false negatives can all be considered outliers when compared to the training dataset and hence it is reasonable that the model classified them incorrectly.

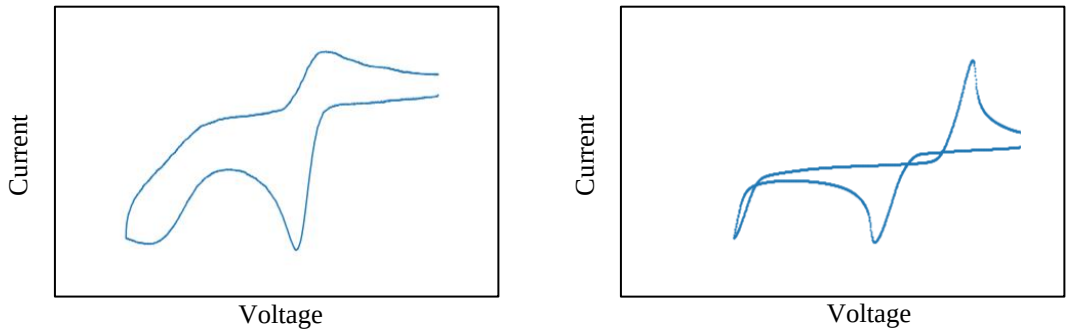


Figure 4.16. Examples of false negatives predicted by the CNN

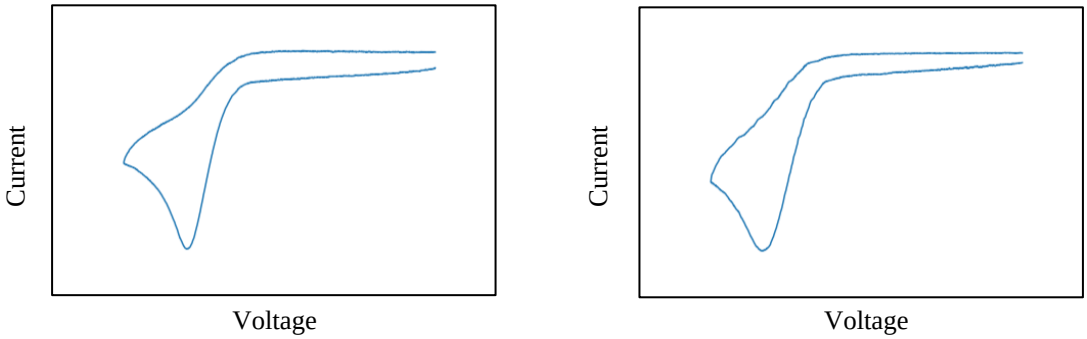


Figure 4.17. Examples of false positives predicted by the CNN

4.6.2 *Long short term memory networks*

The current and voltage values of each cyclic voltammogram are used as features to train the LSTM. The input type is a *3D numpy* array as shown in Figure 4.18. The same number of timesteps was obtained for each voltammogram by selecting a particular number of equally spaced values.

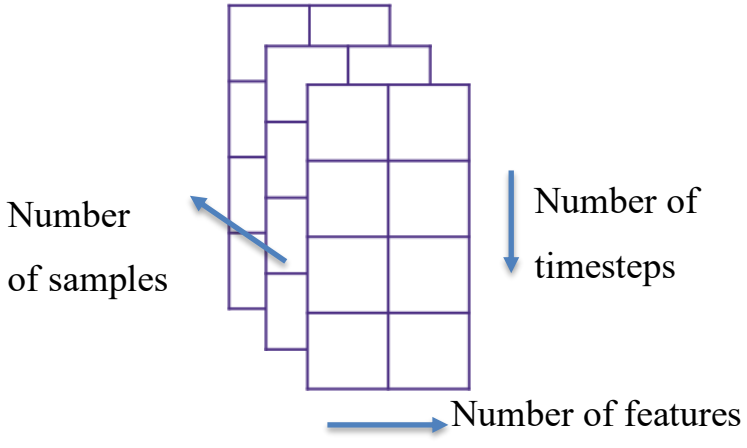


Figure 4.18. LSTM input array shape

The existing dataset was split exactly like it was for the convolutional neural network. The training and validation metrics obtained at each training epoch are shown in Figures 4.19 and 4.20. The Tables 4.3 and 4.4 show the metrics obtained after training the network. The confusion matrix describes the perform ance of the models on the testing dataset.

Table 4.3. Performance metrics of the LSTM

	Binary Accuracy	Binary Loss
Train	0.8655	0.1941
Validation	0.8730	0.2780

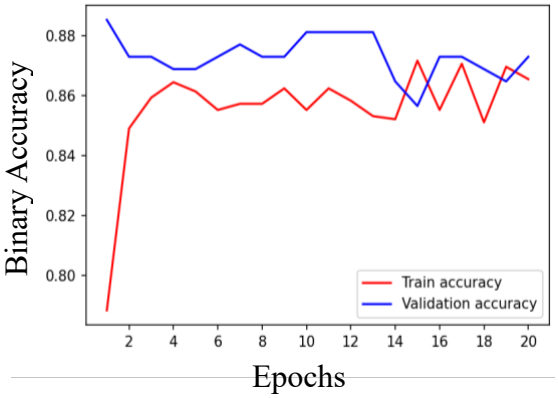


Figure 4.19. Training and validation accuracy over epochs

Table 4.4. Performance metrics of the LSTM on the holdout dataset

	Actual : Reversible	Actual: Irreversible
Predicted: Reversible	183	25
Predicted: Irreversible	2	66

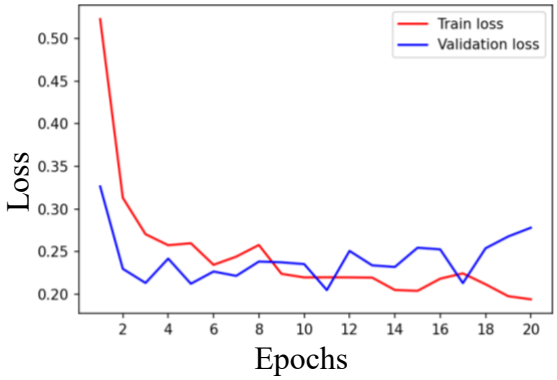


Figure 4.20. Training and validation loss over epochs

Assessment of the performance metrics and the confusion matrix indicates that the model requires more training. The model evaluation on the testing dataset gives a greater number of false positives and false negatives when compared to the CNN. The cyclic voltammograms are being evaluated by the neural network to determine the reversible curves. Hence, the false positives are irreversible curves classified as reversible and false negatives are reversible curves classified as irreversible by the network. Some examples of the incorrectly classified cyclic voltammograms are shown in Figure 4.21 and 4.22**.** Out of the 25 false positives, 12 of the them belonging to the same dataset as Figure 4.22**.** This signifies that the model is not trained on any

data similar to these and requires further training with data consisting of different relevant features the model might encounter.

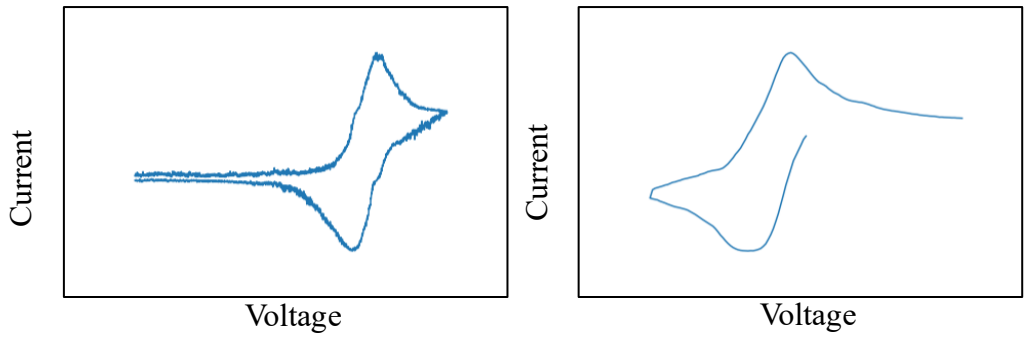


Figure 4.21. False negatives predicted by the LSTM network

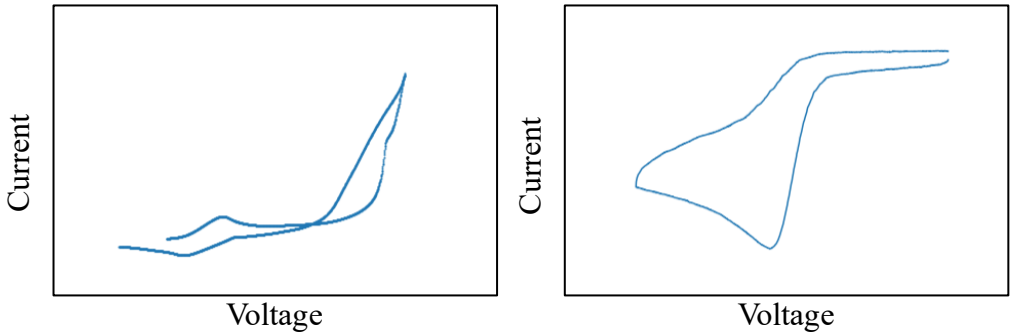


Figure 4.22. False positives predicted by the LSTM network

4.7 CONCLUSION

- CNNs and LSTMs are both algorithms that can be used to effectively classify images and time series sequential data respectively.
- Since both the CNN and LSTM models are trained and tested on the same data, it can be inferred that the CNN model performed better than the LSTM model.
- Assessment of the performance metrics and the confusion matrix obtained from all the models denotes that the models need to be trained with an enormous volume of data consisting of relevant features.

Chapter 5. FUTURE WORK

5.1 FUTURE IMPROVEMENTS TO HIGH THROUGHPUT DETERMINATION OF MELTING POINT

The high throughput setup used to determine the melting point needs the following improvements to integrate all the operations in the process and make it a stand-alone system:

- Integrate the Python package *musicalrobot* and the raspberry pi GUI (Parabilis thermal) used to control the IR camera.
- Increase the code efficiency of the *musicalrobot* package such that it can run on a raspberry pi.
- Automate the video cropping such to include only the well plate.
- Measure the temperature of the well plate with a thermocouple in addition to the camera to evaluate the accuracy of the camera.
- Develop a single Python script which can enable seamless end to end high throughput experimentation, data acquisition, data processing, data classification and data analysis.
- Validate the melting points obtained by making measurements using a Differential scanning calorimetry.
- Develop new methods for synthetic generation of data which capture more features present in the real data.
- Train the neural nets with more data to improve classification accuracy.

5.2 FUTURE IMPROVEMENTS TO HIGH THROUGHPUT ELECTROCHEMICAL TESTING

The high throughput electrochemical testing system needs the following improvements to increase the accuracy of data classification and analysis, integrate all the operations in the process to build a stand-alone system:

- Collect more data belonging to different categories of cyclic voltammograms
- Develop new methods for synthetic generation of data which capture more features present in the real data.

- Train the neural networks with more data to improve classification accuracy.
- Integrate the Python scripts used to read the experimentation files, and process, classify and analyze the data to build a Python package.

5.3 FUTURE STUDIES

There is now a need to discover new materials which can be used to produce and store renewable energy in order to create more efficient energy storage systems so that the consumption of fossil fuels can be decreased. The discovery of an ideal deep eutectic solvent can help increase the energy density and decrease the cost of redox flow batteries leading to remarkable contributions in the field of grid energy storage. Since deep eutectic solvents encompass a vast design space, high throughput experimentation and a closed loop optimization are essential to optimize the number and duration of experiments. Figure 5.1 illustrates the tasks to be carried out to set up a closed-loop optimization.

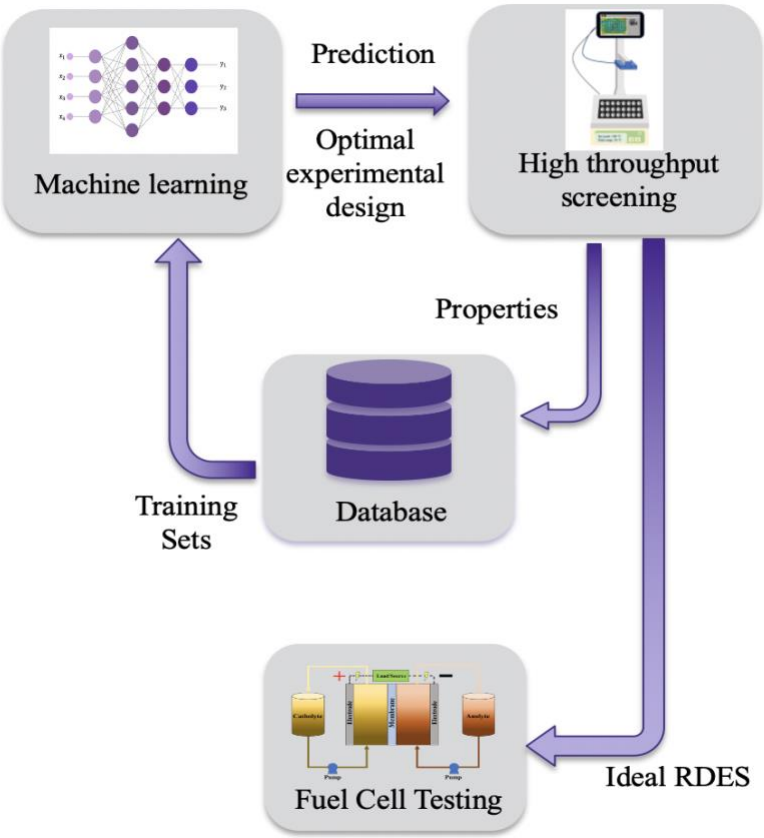


Figure 5.1. Closed loop optimization to discover the ideal redox deep eutectic solvent

The current work can be further developed by carrying out the further studies:

- High throughput experiments have to be developed to measure other relevant properties of the deep eutectic solvents such as viscosity, solubility and to perform further electrochemical characterization. These can be supplemented with data-driven machine learning methods to predict the properties of deep eutectic solvents. Additionally, methods such as Bayesian optimization algorithms can be applied to detect trends and provide feedback to optimize the experimentation design. The predicted properties can be validated via conventional measuring techniques and also by building lab scale redox flow batteries.
- An efficient data pipeline has to be setup to acquire, manage, store and process the data obtained through experimentation. Also, a database has to be built which is easily accessible by and compatible with platforms used to perform high throughput experimentation and data analysis. This protocol can be extended to develop a stand-alone platform by using high level programming languages such as Python to perform high throughput experimentation, acquire and manage the data and perform closed loop optimization.
- It is important to note that a universally acknowledged programming language like Python is especially beneficial to these studies. It can be used to establish coherent communication between various instruments involved in high throughput experimentation, access the database, process the data obtained and develop machine learning techniques. Thus, a single Python application can be used to enable end-to-end high throughput experimentation and closed loop optimization.

BIBLIOGRAPHY

1. Council, N.R., *The National Academies Summit on America's Energy Future: Summary of a Meeting*. 2008, Washington, DC: The National Academies Press. 180.

2. *Monitor.* Proceedings of the Institution of Civil Engineers - Civil Engineering, 2006. **159**(1): p. 7-14.

3. Eckroad, S. and I. Gyuk, *EPRI-DOE handbook of energy storage for transmission & distribution applications.* Electric Power Research Institute, Inc, 2003: p. 3-35.

4. Weber, A.Z., et al., *Redox flow batteries: a review.* Journal of Applied Electrochemistry, 2011. **41**: p. 1137-1164.

5. Abbott, A.P., et al., *Electrodeposition of zinc–tin alloys from deep eutectic solvents based on choline chloride.* Journal of Electroanalytical Chemistry, 2007. **599**(2): p. 288-294.

6. Abbott, A.P., et al., *Solubility of metal oxides in deep eutectic solvents based on choline chloride.* Journal of Chemical & Engineering Data, 2006. **51**(4): p. 1280-1282.

7. Chakrabarti, M.H., et al., *Prospects of applying ionic liquids and deep eutectic solvents for renewable energy storage by means of redox flow batteries.* Renewable and Sustainable Energy Reviews, 2014. **30**: p. 254-270.

8. Bahadori, L., et al., *The electrochemical behaviour of ferrocene in deep eutectic solvents based on quaternary ammonium and phosphonium salts.* Physical Chemistry Chemical Physics, 2013. **15**(5): p. 1707-1714.

9. Box, J.F., *RA Fisher and the design of experiments, 1922–1926.* The American Statistician, 1980. **34**(1): p. 1-7.

10. Steinberg, D.M. and W.G. Hunter, *Experimental design: review and comment.* Technometrics, 1984. **26**(2): p. 71-97.

11. Finney, D.J., *The fractional replication of factorial arrangements.* Annals of Eugenics, 1943. **12**(1): p. 291-301.

12. Olk, C., *Combinatorial approach to material synthesis and screening of hydrogen storage alloys.* Measurement Science and Technology, 2004. **16**(1): p. 14.

13. Jaramillo, T., et al., *Computational High-Throughput Screening of Electrocatalytic Materials for Hydrogen Evolution.* Nat. Mater, 2006. **5**: p. 909-913.

14. Johnson, J.L., H. tom Wörden, and K. van Wijk, *PLACE: an open-source python package for laboratory automation, control, and experimentation.* Journal of laboratory automation, 2015. **20**(1): p. 10-16.

15. Pendleton, I.M., et al., *Experiment Specification, Capture and Laboratory Automation Technology (ESCALATE): a software pipeline for automated chemical experimentation and data management.* MRS Communications, 2019. **9**(3): p. 846-859.

16. Czarnowski, I. and P. Jędrzejowicz, *An approach to data reduction for learning from big datasets: integrating stacking, rotation, and agent population learning techniques.* Complexity, 2018. **2018**.

17. Kim, S.-W. and B.J. Oommen, *A brief taxonomy and ranking of creative prototype reduction schemes.* Pattern Analysis & Applications, 2003. **6**(3): p. 232-244.

18. Jin, R., Y. Breitbart, and C. Muoh, *Data discretization unification.* Knowledge and Information Systems, 2009. **19**(1): p. 1.

19. Liu, H., et al., *Discretization: An enabling technique.* Data mining and knowledge discovery, 2002. **6**(4): p. 393-423.

20. Garcia, S., et al., *A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning.* IEEE Transactions on Knowledge and Data Engineering, 2012. **25**(4): p. 734-750.

21. García, S., J. Luengo, and F. Herrera, *Data preprocessing in data mining*. 2015: Springer.

22. Ramírez-Gallego, S., et al., *Data discretization: taxonomy and big data challenge.* Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2016. **6**(1): p. 5-21.

23. McCulloch, W.S. and W. Pitts, *A logical calculus of the ideas immanent in nervous activity.* The bulletin of mathematical biophysics, 1943. **5**(4): p. 115-133.

24. Warner, B. and M. Misra, *Understanding neural networks as statistical tools.* The american statistician, 1996. **50**(4): p. 284-293.

25. Nicholson, C. *A Beginner's Guide to Neural Networks and Deep Learning*. Available from: https://pathmind.com/wiki/neural-network.

26. Karpathy, A., *Cs231n convolutional neural networks for visual recognition.* Neural networks, 2016. **1**(1).

27. LeCun, Y., et al., *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 1998. **86**(11): p. 2278-2324.

28. Hubel, D.H. and T.N. Wiesel, *Receptive fields of single neurones in the cat's striate cortex.* The Journal of physiology, 1959. **148**(3): p. 574.

29. Rawat, W. and Z. Wang, *Deep convolutional neural networks for image classification: A comprehensive review*, *Neural computing*. 2017, MIT Press Journals.

30. Tompson, J., et al. *Efficient object localization using convolutional networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

31. Krizhevsky, A., I. Sutskever, and G.E. Hinton. *Imagenet classification with deep convolutional neural networks*. in *Advances in neural information processing systems*. 2012.

32. Olah, C. *Understanding LSTM Networks*. 2015
Available from: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

33. Pedregosa, F., et al., *Scikit-learn: Machine learning in Python.* the Journal of machine Learning research, 2011. **12**: p. 2825-2830.

34. Kawakami, K., *Parallel thermal analysis technology using an infrared camera for high-throughput evaluation of active pharmaceutical ingredients: A case study of melting point determination.* Aaps Pharmscitech, 2010. **11**(3): p. 1202-1205.

35. *LEPTON®, F. How Do Thermal Cameras Work?* ; Available from: https://www.flir.com/discover/rd-science/how-do-thermal-cameras-work/.

36. *LEPTON®, F., FLIR LEPTON® Engineering Datasheet*. p. 18-22.

37. *LEPTON®, F.*, ***FLIR LEPTON® Engineering Datasheet***. p. 22-23.

38. ***LEPTON®, F.***, ***FLIR LEPTON® Engineering Datasheet***. p. 30.

39. ***LEPTON®, F.***, ***FLIR LEPTON® Engineering Datasheet***. p. 32.

40. Parks, K., *Raw Thermal Data over UVC from PureThermal 2 - Lepton 3.5 - Python*. 2019.

41. Malik, R., *Flow Battery Solvents: Looking Deeper.* Joule, 2017. **1**(3): p. 425-427.

42. Mabbott, G.A., *An introduction to cyclic voltammetry.* Journal of Chemical education, 1983. **60**(9): p. 697.

43. Elgrishi, N., et al., *A practical beginner's guide to cyclic voltammetry.* Journal of Chemical Education, 2018. **95**(2): p. 197-206.

44. Mattia, P., *A simple cyclic voltammetry simulator, built as a MATLAB app. Based on Appendix B in Bard & Faulkner.* 2018.