COMP390

2021/22

A low-dimensional map of high-dimensional data

Student Name:	You Wu
Student ID:	201521317
Supervisor Name:	Matthew Bright

DEPARTMENT OF
COMPUTER SCIENCE

University of Liverpool
Liverpool L69 3BX

Acknowledgements

First of all, I'd like to thank my parents for their support and accompany, which gave me the chance and ability to pursue the purposes of my life.

I want to thank my supervisor Matthew Bright for patiently and kindly supporting me throughout the project. I want to thank Daniel Widdowson and Vitaliy Kurlin from the research group for their enormous help during the project.

COMP390

2021/22

A low-dimensional map of high-dimensional data

Student Name:	You Wu
Student ID:	201521317
Supervisor Name:	Matthew Bright

DEPARTMENT OF
COMPUTER SCIENCE

University of Liverpool
Liverpool L69 3BX

Abstract

We developed a neural network model that is capable of predicting the element type of an atom within a crystal structure given its k-NN distances as a representation of its local topological environment. Trained by data retrieved from crystal structure databases, the model was able to predict the element type with an accuracy of over 80% on 20 elements. The model could be applied in crystal structure prediction (CSP) as a fast filter to prevent unnecessary expensive computations by suggesting the likelihood of the generated structure. In addition, the relation between the elements and their k-NN distances, which is the foundation of the model's predictive power is analysed by examining the distributions of k-NN distances between different pairs of elements and extracting the bonds formed between the nearest neighbours. A conjecture of how the k-NN distances might be conveying chemical information and how the model's performance may be influenced by k was proposed.

Table of Contents

1. Introduction

1.1 Crystals and k-NN distances

- Crystals

Crystals are solid materials with periodic arrangements of atoms [1, Ch. 1, pp. 1-4]. The *lattice*, being a grid of infinite points in space, describes the spatial periodic repetition of identical structural units. Once a lattice is defined, we associate each lattice point with a finite set of atoms (called *motif*) to represent the crystal structure in space (See Figure 1.1) [1, Ch. 2, pp. 1-4]. A Crystal can thus be expressed by the sum of a lattice and a motif.

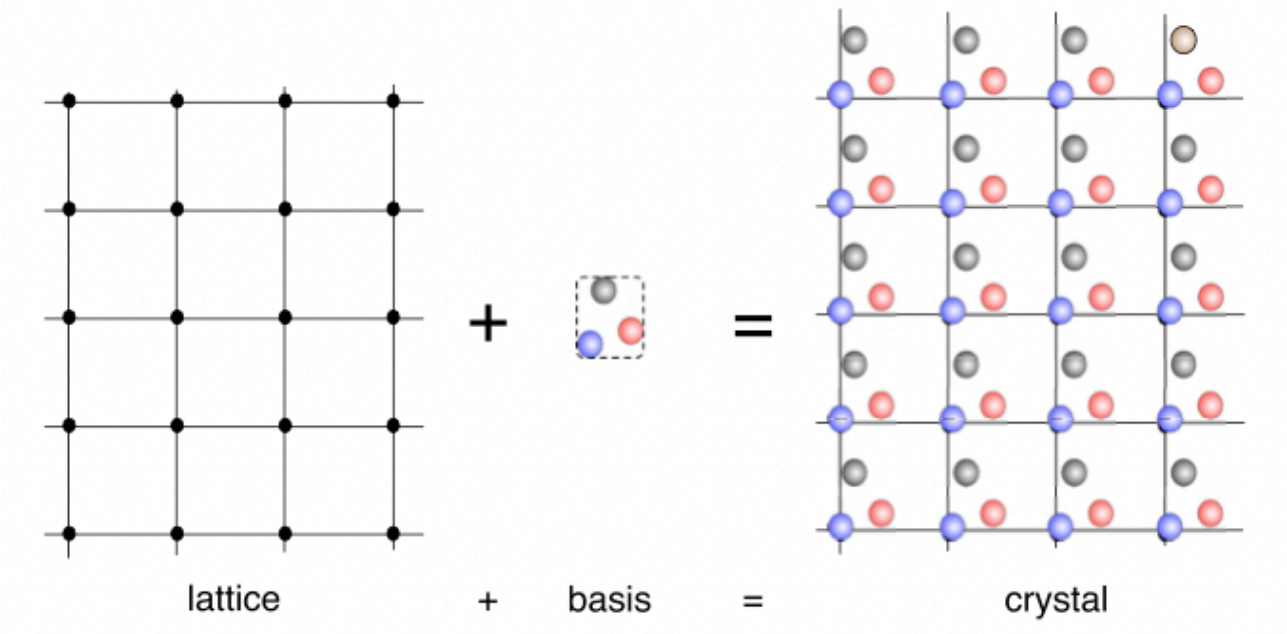

Figure 1.1 An example showing a two-dimensional crystal can be represented by a lattice and a basis (motif) [1, Fig 2.2].

Mathematically, a lattice can be given by the integer linear combinations of a set of basis vectors that span a *unit cell*, which is the smallest repeating unit. The point set contains all integer linear combinations of these basis vectors. A *periodic point set* is the Minkowski sum $\Lambda + M = \{\vec{u} + \vec{v} : \vec{u} \in \Lambda, \vec{v} \in M\}$ of a lattice Λ and a motif M [2, p. 530].

For instance, Sodium chloride (NaCl), widely known as salt, is a very common type of crystal. It has a cubic unit cell, where sodium ions and chloride ions alternate with each other in each of the three dimensions.

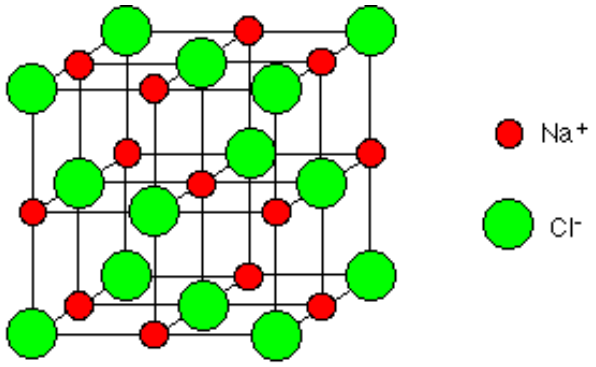

Figure 1.2 A diagram showing the unit cell of sodium chloride [3].

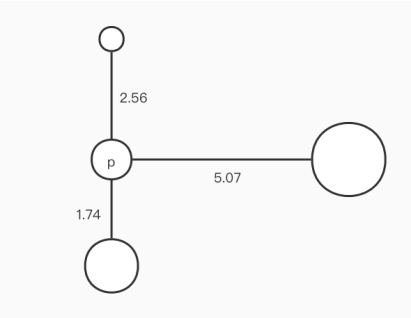

Figure 1.3 A diagram showing 4 atoms with their interatomic distances.

• k-NN distances

Suppose the particular atom we are interested in is p. S is the increasingly ordered set of the distances from all other atoms in the crystal to p. k-NN distances is then defined as a vector containing the first k elements in the set S.

For example, as shown in Figure 1.3, there are in total 4 atoms, with one of them, p, being the atom we are interested in. First, we sort the distances of the atom p to all other atoms in ascending order: S=[1.74, 2.56, 5.07]. Then the k shortest distances of the atom p is the vector consisting of the first k elements in the set S. When k=2, the 2 shortest distances of atom p is [1.74, 2.56].

1.2 Crystal Structure Prediction

Crystal structure prediction (CSP) is the problem of predicting, by computational methods, how a molecule will crystallise given only its chemical diagram and crystallisation conditions [4, p. 107]. Since many properties of the material are fundamentally controlled by crystal structures, with proper prediction methods, it would be possible to design molecules that will crystallise with desired structural and physical properties, which will lead to enormous implications in industries, for example developing organic crystalline materials (e.g. pharmaceuticals). Alternatively, CSP can be used to assess the likelihood of any other undiscovered polymorphs to prevent generating undesirable products [4, 5].

Cambridge Crystallographic Data Centre (CCDC) has been organising CSP Blind Tests since 1999 to gather scientists and researchers from industry and academia to test their methods against real examples. According to the report on the latest CSP Blind Test [5, 6], the main approach to CSP has remained largely unchanged since the earliest published attempts. Generally, the process of crystal structure prediction can be broken down into 2 steps:

(i) Generating plausible crystal-packing structures of the given molecules based on their chemical properties.

(ii) Ranking the likelihood of resulting crystal structures using some form of scoring or fitness function.

For the second step, the evaluation of generated crystal structures has been dominated by approximating the structures' DFT energies (or lattice energies) using quantum-mechanical methods, which would become prohibitively computational-intensive and time-consuming with dense and/or complex crystal structures [6, 7]. The search for the stable crystal structure, which is the one corresponding to the global minimum of the

free energy surface, is made extremely difficult by the fact that the free energy surface is high-dimensional (having a degree of 3N+3, where N is the number of atoms in the unit cell), and typically have an enormous amount of local minima separated by high barriers [8].

A typical ranking algorithm will take days or even weeks on a single crystal structure optimisation. For example, the method DFT-D, with the highest success rate in fourth and fifth blind tests, would typically require one day of CPU time for one crystal structure optimisation on a state-of-the-art PC at that time [17, 18]. Taking the development of computational power into consideration, it still indicates a high computational complexity.

1.3 Literature Review

Providing solutions to the ranking problem mentioned above has long been an active area of research. Many algorithms evaluating the energy with higher accuracy and lower complexity were developed and proved to be improving the success rate of the prediction [4-6, 8, 17-18].

Machine learning algorithms, especially neural network models, have seen increased application in chemistry-relevant problems. [24] constructed a structural overview of emerging applications of deep neural networks in chemistry, for example, drug design, protein contact prediction and compound properties prediction. As an example, a model based on convolutional neural networks (CNN) was introduced for the classification of powder X-ray diffraction (XRD) patterns in [26]. It trained CNN with an overwhelming amount of XRD pattern data without feature engineering involved, which turned out to enable the prediction of crystal systems of totally unknown materials. In particular, there is research that embeds deep learning algorithms into the ranking of generated candidates for material prediction. [25] introduced deep neural networks in traditional evolutionary algorithms as a model of evaluation; [7], in particular, constructed a deep neural network to learn representation from normalised Atomic Fingerprints (AFP), which is a 2-dimensional representation of the 3-dimensional topological environment around the atom. The model culminated in having an average error rate of 31% on the validation set.

1.4 Problem Statement

The popular approaches to CSP ranking problems, of which the vast majority are based on energy calculations, are introduced above to be significantly compute-intensive and may possibly contain unnecessary computations of crystal structures that are highly improbable.

This project aims to develop a neural network model that is able to generate predictions of the atom types given their k nearest neighbour distances. Since the overall likelihood of a crystal structure can be approximated as the product of the likelihood of every atom in the *asymmetric unit*[1], the overall likelihood would decrease significantly to some value close to zero when the model detects an atom being highly implausible at the suggested position and outputs a very low likelihood. For instance, suppose we have a generated crystal structure with n atoms $a_1, a_2, \ldots, a_n$ within the asymmetric unit. The n atoms are of types $t_1, t_2, \ldots, t_n$ and have k-NN distances $\overrightarrow{d_1}, \overrightarrow{d_2}, \ldots, \overrightarrow{d_n}$ respectively. The likelihood of the structure can be evaluated as the product of the likelihood of these atoms:

$$L(structure \mid topological\ representation) = \prod_{i}^{n} L(t_i \mid \overrightarrow{d_i}),$$

1 *Asymmetric unit*: the smallest part of the unit cell that can generate the complete unit cell by symmetric operations

where $L()$ represents a likelihood function, $L(t_i|\overrightarrow{d_i})$ is generated by the model through the output of the neuron corresponding to the type t_i, and the topological representation of the structure is given by k-NN distances $\overrightarrow{d_1}, \overrightarrow{d_2}, \ldots, \overrightarrow{d_n}$. Note that, the likelihood of the structure would be noticeably decreased if any of the atoms have a low likelihood. Therefore, although the model cannot generate a ranking of suggested crystal structures, it could serve as a quick filter that efficiently eliminates highly unlikely crystal structures from further computationally expensive evaluations, given generating predictions for 40,000 atoms spent only 5 seconds on a modern 8-core desktop.

1.5 Aims & Objectives

The aims and objectives of this work remained largely the same as they were in the project proposal, with one objective not tested and one new objective added.

1.5.1 Aims

The aim of this project is to design a general neural network model that is capable of predicting atom types from k nearest neighbour distances after being trained. In particular, the model shall be adaptable to a larger k by reusing previously trained parameters. Additionally, an intent was developed during the project process to analyse how the k-NN distances are carrying chemical information.

1.5.2 Objectives

- To develop a neural network model that is capable of predicting chemical elements given their k-NN distances.
- To develop an algorithm for a trained model to adapt to a larger k.
- To develop a method to evaluate the improvement of performance and the amplification of complexity of the model by increasing k.
- To suggest a best k value of complexity-accuracy tradeoff for the designed model.
- (New) To extract the distribution of the kth nearest neighbour distances on various k and pair of atom types.
- (New) To extract the nearest neighbours forming chemical bonds and plot the distribution of bonds within nearest neighbours.
- (New) To propose a conjecture on how the k-NN distances are conveying chemical information and how the performance of the model may be affected by k based on the data extracted above.

1.6 Outline

This article will describe a neural network model trained to predict the atom type of an atom given its k-NN distances and analyse how k-NN distances might be conveying chemical information.

All subsequent sections, except sections 5 and 6, are divided into two parts, presenting contents concerning the neural network model and analysis of k-NN distances respectively. This paper first introduces the methodology it applied by describing the development, architecture and training process of the neural network, explaining how training and testing data were retrieved in section 2.1, and demonstrating the methods used to extract the distribution of the kth nearest neighbour distances and the nearest neighbours

forming chemical bonds aiming to explain the model's predictive power in section 2.2. Then, the paper illustrates the performance of the model and the distribution extracted in section 3. Finally, in section 4.1, an optimum k value is proposed and possible applications of this work are explained in section 4.1. In Section 4.2, a conjecture of the reason for the model's predictive power was presented based on previous data analysis. Note that since the project focuses more on research, there is less testing and evaluation of the software. The descriptions of how the project is fulfilling objectives were placed in the Discussion section rather than having a separate section.

2. Methods

2.1 Constructing the Neural Network Model

Software Libraries

The project was developed with Python. The software libraries used include: average-minimum-distance (AMD) [2, 9], Cambridge Structure Database and corresponding Python API [10], NumPy [11], Pandas [12], TensorFlow [13], and Matplotlib [14]. The AMD package was used for calculating nearest neighbour distances. The TensorFlow library was used for constructing, training and evaluating the neural network model. The NumPy and Pandas libraries were used for data processing. The confusion matrix was computed using a function provided by Scikit-learn [15]. Later in the project, the CSD API was used for the analysis of the nearest neighbour by bonds.

Crystal Structure Dataset

The dataset used for k-NN distances calculation was provided by Widdowson within the research group. The dataset is a filtered subset of the CSD, where crystals with the following characteristics were removed from the dataset [22]:

(i) Crystals which have missing data on atomic coordinates or unit cells. For example, some crystals may have known composition, but the coordinates, which are crucial for computing distances are missing or not recorded.

(ii) Crystals which have some form of *disorder*[2]. The k-NN distances are therefore not applicable over these atoms, as they may vary among unit cells.

The filtered crystals are then stored as periodic point sets (`periodicset.PeriodicSet` objects in [9], which records the element types and 3-dimensional coordinates of the atoms within the asymmetric unit) in a compressed .hdf5 file.

Input Data preprocessing

Widdowson provided the scripts for generating the k-NN distances of given interested types from the preprocessed dataset based on his work AMD [9] (see `/scripts/get_type_distance_data.py`[3] for source code and Figure 2.3 for pseudocode). The script was based on a function in AMD that returns k-NN distances

[2] A crystal has *disorder* if the atomic positions are not perfectly periodic. For example, the same atom may have a slight shift among different motifs.

[3] Figures or files contained in the archive are referred to with a relative path.

and the type of each neighbour for every atom in the asymmetric unit given a crystal structure (`amd.nearest_neighbour()`). It first shuffles the entries in the crystal structure list, then iterates through every crystal to generate k-NN distances and neighbours' types list for every atom within the interested

Attribute(s)	Data type	Description
type of centre atom	String	Element type of the centre atom
crystal id	String	id of the crystal the centre atom is in
k-NN distances	integer	k-NN distances of the centre atom
k-NN types	String	element types of the k nearest neighbours

Figure 2.1 The structure of one record in the k-NN dataset

types. For each atom of interested types, the id of the crystal structure it belongs to, which is the identifier of the crystal in CSD [10], and the type of the centre atom were concatenated to the list of k-NN distances and types as one record. The script stops when the number of records for each type reaches the preset upper limit, or every crystal structure has been visited.

atom_type	ID	dist1-dist10	nn_type1-nn_type10
C	KUSXEB	1.3287, 1.4471,…	C,C,H,H,O,C,…

Figure 2.2 A sample record in the k-NN dataset

At first, the model was tested on 10 element types: Li, C, N, O, Na, Al, Si, P, S, K. During the extraction, the upper limit of each element type was set to 10,000. The total count of records was 92,963. Note that the

```
shuffle(dataset)
# initialise the element types interested
interested_elems=[...]
# initialise upper limit of count for every element type
upper_limit=...
sample_count_for_every_element={elem:0 for elem in interested_elems}
result=[]

for crystal in dataset:
  list_of_atom_types=get list of atom types in crystal
  for atom_type in list_of_atom_types:
    if atom_type in intersted_elems and sample_count_for_every_element[atom_type]<upper_limit
      add atom_type, crystal_id, nn_distances, nn_types to result
      sample_count_for_every_element[atom_type]++

save result as csv file
```

Figure 2.3 pseudocode of the extraction of k-NN distances

number is less than $10 \times 10{,}000$ because some element types may have a total number of records less than 10,000.

Then the model was extended to the top 20 most abundant elements in the CSD [10]. See the count of every element type in the CSD in `/data/atomic_types_counts.csv`. The upper limit was set to 100,000. The total number of records was 1,365,173.

To test the model, the test dataset is a separate set generated randomly given the interested elements and the upper limit of 2,000.

Note that the dataset contains only selected element types, there exists an inherent class imbalance, which means that the frequency of classes within the dataset is highly imbalanced. This leads to a lower precision on infrequent classes, as the model can improve its performance on the training set by simply increasing the frequency of predicting the majority class, rather than learning the characteristics from the data. In our case, we fix the class imbalance problem in order to reuse the trained parameters on expanded element types i.e. to achieve the objective of model reusing.

Model Architecture

The structure of the model (see Figure 2.4) was mainly based on the idea from [7], where the input data was another representation (Atomic Fingerprints, AFP) of the local topology around each atom. The model consists of two parts. The first part is a five-layer sigmoid classifier, which is meant to solve the class imbalance by letting the classifier learn which elements are not presented in the dataset. The input of this part is a vector of length k, representing the k-NN distances of the atom. The classifier has 118 output neurons, each representing a chemical element in the periodic table. In this part of the model, a class weight of 118 was applied to the true classes. Since the unobserved classes are never presented in the training set, the model thus learns that these elements have zero likelihood to occur on any input [7, Methods]. Note that, these class weights need to be adjusted when new classes are added. Then, the output, which is a vector of length 118, was concatenated with the original k-NN distances and fed into a softmax classifier, which outputs a normalised probability distribution on 118 elements.

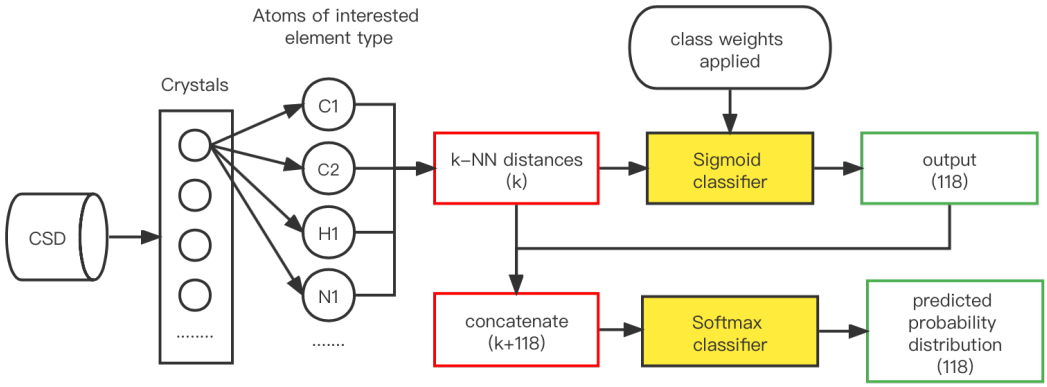

Figure 2.4 The architecture of the neural network model

Training Notes

The model was trained on Barkla [16], the university's high-performance computing cluster. The two parts of the model were trained sequentially, with a typical total training time of 5-6 hours. The layers are fully and sequentially connected, with each connection having a weight and each neuron having an excitation threshold, both trainable. The model gets trained by a method called batch gradient descent, where the gradient of the loss function of the model is evaluated on the basis of batches, and the parameters of the

model are adjusted in the opposite direction of the gradient, where the value of the high-dimensional loss function decreases the fastest.

The model was trained using the Adam optimizer with a learning rate[4] of 0.001 for 60 epochs[5]. During the training of the model on the top 20 most abundant elements in the CSD, the improvement of the model's performance over epochs, measured by categorical accuracy[6], was recorded and plotted (it will be presented in section 3.1).

The script for training can be found at `/scripts/training.py`.

2.2 Analysing the nearest neighbour distances chemically

The model showed an over 80% accuracy after training (see section 3.1). To explain the model's predictive power, the distributions of nearest neighbour distances for each element type were extracted and analysed. For simplicity, the analysis was restricted to four element types: C, H, O, and N.

Extracting the distribution of NN-distances by order and element type

For analytic purposes, we would like to assign each atom in the asymmetric unit of a crystal a unique identifier. Fortunately, the CSD [10] provided `ccdc.molecule.Atom.label`, which is of the form type + id (e.g. 'C1') and uniquely identifies each atom in the asymmetric unit.

Although the k-NN distances extraction script mentioned before generated the centre atom's atom type, it cannot generate the index for the atom, since the `PeriodicSet` objects recorded only types of the atoms in the asymmetric unit, as mentioned before in **Crystal Structure Dataset** section. Fortunately, since the list of types corresponds to the list of labels within CSD [10] (both referring to the atoms within the asymmetric unit in the same order) and the `amd.nearest_neighbour()` method traverses the list in order, we are able to obtain the indices by substituting the types with labels retrieved from the CSD.

Therefore, a script was written to extract the list of labels within the asymmetric unit for each crystal present in the dataset from the CSD (See `/scripts/label_extraction.py`). For each crystal, a file was generated to store the labels list. The file name is the id of the crystal structure.

Then, a program was developed to iterate through the k-NN distances file and substitute the centre atom types with labels extracted from the CSD. See `scripts/label_replace.py` for the source code and Figure 2.5 for pseudocode.

However, indexing neighbours would be ambiguous as some neighbour(s) of an atom could be in another asymmetric unit, which has the same label as its counterpart in this asymmetric unit.

Then, the distributions of the kth nearest neighbour distances are plotted by iterating through the dataset to generate a dictionary of vectors, each storing the distances of a particular pair of element types. See `/scripts/nn_distance_distribution_plotter.py` for source code. The distributions were extracted and plotted for k=1-6. These distributions were further split by pair of atom types and plotted for further analysis.

---

[4] Learning rate: the learning rate is the step size of each modification of the parameters.

[5] Epoch: an epoch indicates one thorough iteration of the training dataset.

[6] Categorical accuracy: with the model's prediction being the corresponding element type of the neuron having the maximum output value, the categorical accuracy is the proportion of records where the model makes a correct prediction.

```
# initialise interested element types
interested_elems=...

# initialise a list storing data for current crystal being visited
current_dataframe=[]

for row in csv file:
  if row[id] is the same as the last row[id]:
    current_dataframe.append(row)
  else:
    get labels by last row[id]
    add a column of labels to current_dataframe
    write current_dataframe to file named with last row[id]
    clear current_dataframe
```

Figure 2.5 Pseudocode for indexing atoms by labels retrieved from CSD [10].

Extracting bonds within nearest neighbours

The k-NN distances might be carrying chemical information as there are neighbours forming a bond with the atom, whose length is dependent on bond type and the atoms forming it while being nearly the same across various molecules [19, Chap. 7, p. 222]. For example, experimental bond lengths of carbon-carbon single bonds[7] probed in 7 molecules [19, Table 7-1, p. 222] ranged between 1.53 and 1.54Å ($10^{-10}m$), being equal within the probable errors. It is also shown that covalent bond lengths tend to behave in an addictive manner, meaning that the bond length of A-B equals the arithmetic mean of bond lengths of A-A and A-B of the same bond order [19, p. 223]. As given in [19, Table 7-2], the covalent radii[8] showed a periodic dependence on atomic numbers. In particular, the covalent radii for carbon-carbon single, double and triple bond is 0.772, 0.667 and 0.603Å respectively. Note that the covalent radius decreases as the bond order increases, which indicates a stronger bond. A special type of bond, known as aromatic bond can also be formed between carbon and carbon in a ring structure called an aromatic ring, which contains six carbon atoms. The electrons are known to be delocalised and shared equally between each of the carbon atoms. The resulting bond length, being 1.40Å [21], is greater than a double bond while shorter than a single bond. A new set of covalent radii [20, Fig. 2 and Table 2], deduced from the latest crystallographic data and technologies, showed similar results and complied with the theory stated above. Therefore, by extracting the distribution of bond lengths of various atom pairs within the nearest neighbour distances, we might be able to visualise how chemical information is carried by k-NN distances.

The CSD holds registered information of bonds that the atom forms for every atom within the asymmetric unit, including bond type, the label of the other atom forming the bond, and the length of the bond, which enabled our extraction of bonds within the nearest neighbours. However, since the indices of the neighbours are missing, and there typically exists a slight error between the registered bond length and the calculated neighbour distance, we cannot identify bonds immediately from the list of neighbours.

Therefore, we first iterated through the k-NN dataset to pick out atoms with matching types of the nearest neighbour and the other atom of its shortest bond. Then, the differences between the nearest neighbour

[7] A ‘single’ bond indicates the bond order of the covalent bond is 1. A covalent bond is an interatomic bonding as a result of sharing electron pairs. The bond order of a covalent bond is defined as the number of pairs of electrons that the two atoms are sharing. Similarly, a double bond indicates a bond order of 2, a triple bond indicates a bond order of 3 etc.

[8] Covalent radius: a measure of the size of an atom that forms one part of a covalent bond. The bond length should theoretically equal the sum of the covalent radius (of the corresponding bond order) of the two atoms forming it.

distance and the shortest bond length were calculated and plotted. A threshold was set accordingly, and these atoms with the calculated difference less than the threshold (set to 0.05Å) were extracted and regarded as highly likely to be forming a bond with their nearest neighbours for having such a small difference in distance. Since hydrogen atoms tend to form simple bonds while having a large number of records in the dataset, it was ignored from bond extraction.

More specifically, 3 scripts were developed for the above purposes:

(i) A script (see `/scripts/bond_info_extractor.py`) was developed to extract bond information of atoms within the dataset from the CSD [10]. See Figure 2.6 for pseudocode.

```
# initialise interested element types
interested_elems=['C','H','O','N']

for crystal in dataset:
  dict_of_bond_info=[]
  for atom in crystal.asymmetric_unit.atoms:
    if atom.type in intersted_elems:
      get bonds from CSD by atom.label
      sort bonds increasingly by bond length
      dict_of_bond_info[atom.label]=bonds
  write dict_of_bond_info to file named by crystal_id
```

Figure 2.6 Pseudocode for extraction of bond information from the CSD.

(ii) Then a script (see `/scripts/difference_extractor.py`) iterates through the atoms and calculates the differences between shortest bond lengths and the nearest neighbour distances with matching element types.

(iii) Finally, a threshold was set based on the differences extracted above. We iterate through the distances dataset again, calculate the differences and pick out atoms with the difference between the nearest neighbour distance and the shortest bond length less than the threshold and regard them as atoms forming a bond with the nearest neighbour (see `/scripts/neighbour_bond_extractor.py` for source code and Figure 2.7 for pseudocode).

```
# initialise interested element types
interested_elems=['C','O','N']
# set threshold of difference for extraction
threshold=0.05
# initialise the dictionary storing bond lengths for different pair of atoms (e.g. (C,C), (C,N) etc.)
# of various bond types
bond_lengths={type_pair: {bond_type:[] for bond_type in bond_types} for type_pair in type_pairs}
for atom in dataset:
  if atom.type in interested_elems:
    get shortest_bond_info of the atom.label from the crystal with atom.crystal_id
    # There are some bonds in the CSD having missing bond length information. If the atom have
    # any of its bond length missing, the record is eliminated to avoid possible errors.
    if shortest_bond_info is not None:
      if shortest_bond_info.the_other_atom.type=atom.nearest_neighbour_atom.type:
        if abs(shortest_bond_info.length-atom.nearest_neighbour_distance)<threshold:
          bond_lengths[shortest_bond_info.type].append(shortest_bond_info.length)
save bond_lengths to file
```

Figure 2.7 Pseudocode for extraction of lengths and types of bonds formed by nearest neighbours.

3. Results

3.1 Machine learning model's performance

The performances of the neural network models were visualised with confusion matrices[9].

The neural network model was first tested on the prediction of 10 element types as a demo. As mentioned in **Input Data Preprocessing**, the total count of samples of the training set used to train the demo was 92,963. The confusion matrix when k=5 is shown in Figure 3.1.

As illustrated in the diagram, despite confusing Lithium (79%) with Aluminium (62%), the model showed an accuracy of about 90% for most elements. The demo was trained on k=1-5, see the matrices of other k values in `/neural_network_res/demo`.

With the model structure appearing to be effective, the model was then trained on the top 20 most abundant elements in the CSD [10]. Setting an upper limit of 100,000 for every single element, the total number of samples extracted for the training set was 1,365,173. The model was trained for k=1-10, 60

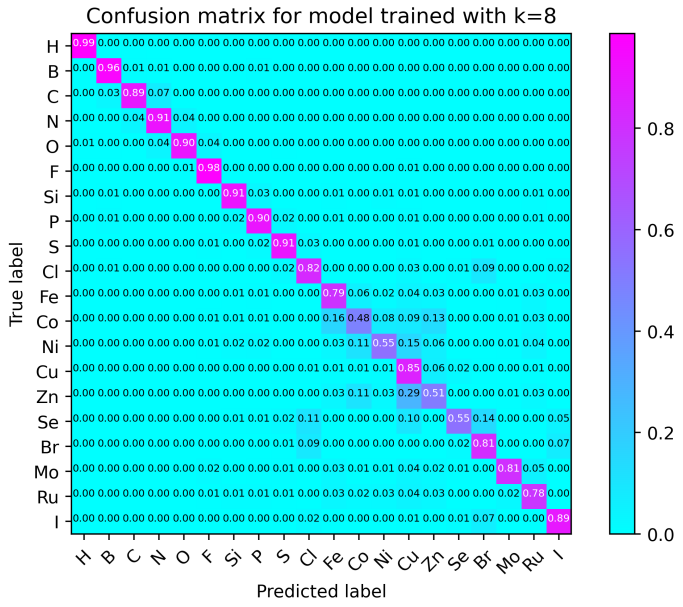

Figure 3.4 (k=8) The confusion matrix of the model trained with 20 elements obtained on the test set.

epochs each. Figures 3.2-3.4 show the confusion matrices for k=2, 5, and 8 respectively.

According to Figure 3.4, when the model is trained with k=8, the prediction of most elements has an accuracy of over 80%. It can be observed from the figures that the predictions tend to be more concentrated on true labels (i.e. the value of diagonal elements increases) as k increases.

[9] Confusion matrix: a confusion matrix is a matrix where each row represents an actual class and each column represents a predicted class. The number in the cell represents the percentage of the corresponding actual class being predicted as the corresponding predicted class (i.e. accuracy). A classifier perfectly identifies all objects would have a confusion matrix with all diagonal elements being 1 and all others being 0.

The improvement of the model's accuracy over epochs was also plotted for k=5 (see Figure 3.5). The accuracy improved significantly over the first few epochs and then steadily increase at a slower rate.

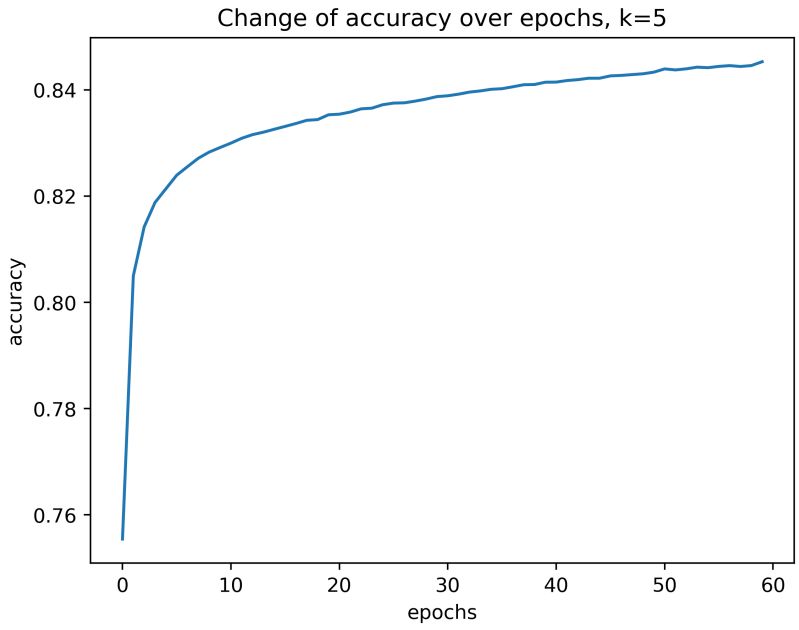

Figure 3.5 The change of accuracy of the softmax classifier over epochs for k=5.

If we evaluate the performance of the model by the sum of the values of diagonal elements in the confusion matrix, the improvement of the model's performance over k is shown in Figure 3.6. The performance of the model monotonically increases over k, but the rate slows down after k=5.

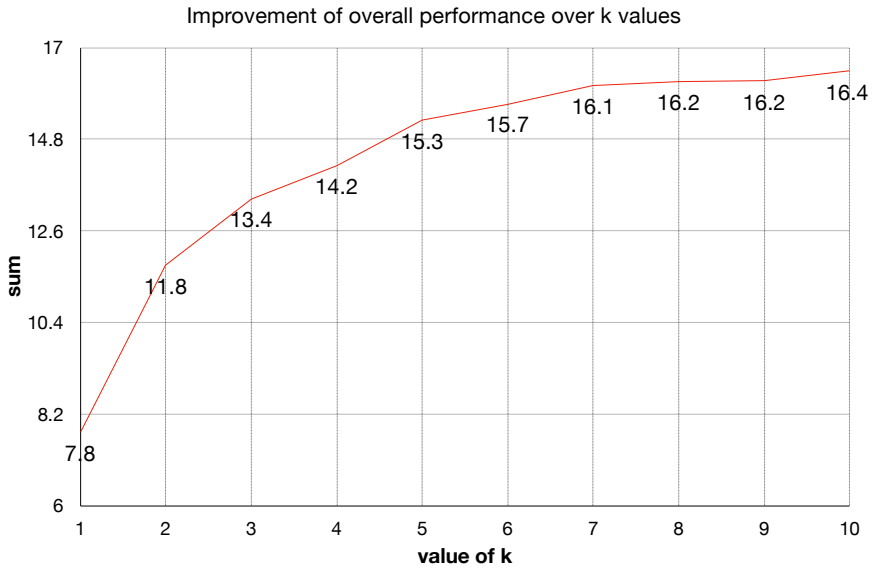

Figure 3.6 The improvement overall performance of the model over k. Note that there are in total 20 elements, so the maximum of this sum would be 20.

The overall accuracy of the model can actually be calculated as this sum divided by 20, which is the total number of classes. At k=7, the overall accuracy of the model would be 80.5%.

3.2 Distribution of nearest neighbours

As mentioned in Methods, the distributions of the kth nearest neighbours were extracted for k=1-6. See `/analysis/distribution_nn_distances/` for all histograms generated.

Figure 3.7 shows a distribution of all shortest nearest neighbour distances with a total number of samples being 1,493,757. Note that most of the shortest nearest neighbour distances fall between 0.5 and 2.0Å, while some extremely high peaks can be observed at somewhere between 0.75 and 1.0Å.

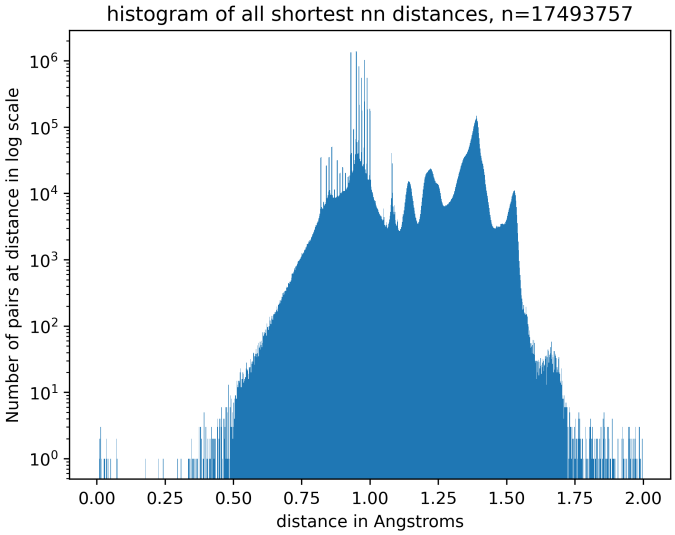

Figure 3.7 A histogram showing the overall distribution of the shortest nearest neighbour distances.

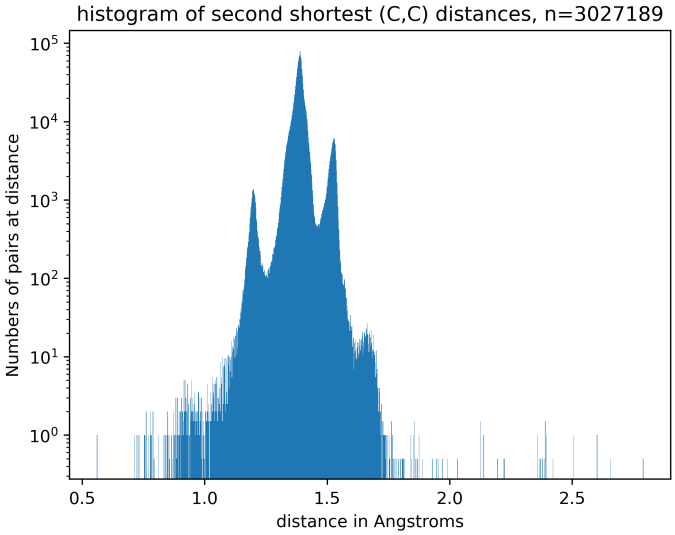

Figure 3.8 A histogram showing the distribution of shortest nearest neighbour distances where the centre atom is a carbon and its nearest neighbour is also a carbon.

Figures 3.8-3.10 illustrate some distributions of distances between particular pairs[10] of elements being the nearest neighbour. All these distributions tend to form peaks at some specific distance, while the locations and magnitudes of the peaks vary among different pairs of elements. For example, the (C, C) pair forms a peak with a distance greater than 1.5Å, while the other two pairs have only a few samples with a distance greater than 1.5Å. Except for some distributions with only a few samples, this fact can be observed similarly in other distribution figures. In addition, following a similar analysis process, the distribution of kth nearest neighbours of a fixed pair of elements appeared to be different depending on the value of k.

3.3. Distribution of bonds within nearest neighbours

The histogram showing the differences between the nearest neighbour distances and the shortest bond lengths of matching types is demonstrated below in Figure 3.11.

Notice that there is an extremely high peak close to zero and if we examine the histogram of the cumulative count of differences (see Figure 3.12), there is more than 5/6 of the differences fall within 0 to 0.05, indicating the vast majority of atoms having a matching type of nearest neighbour and the atom forming the shortest bond, have only a very small difference between the two distances.

[10] Specifically, the 'pair' here is actually an ordered pair of elements. The nearest neighbour of a C being N does not necessarily indicate the nearest neighbour of that N is this C.

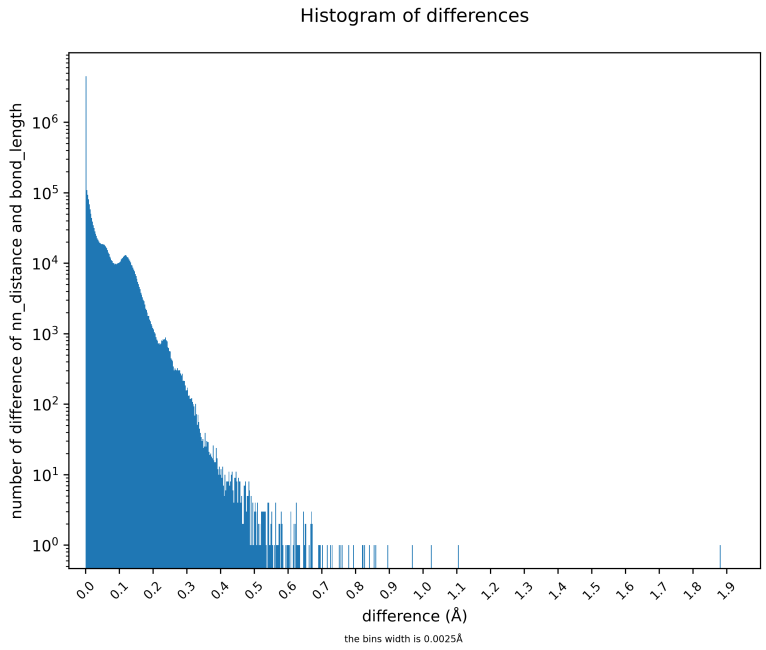

Figure 3.11 The histogram of absolute values of differences between the nearest neighbour distances and the shortest bond lengths

By setting a threshold of 0.05Å as mentioned in **Methods**, a set of bond lengths was extracted by pairs of elements and bond types. Here we take pair (carbon, carbon) as an example. Figures of other pairs can be found in `/analysis/bond_within_neighbours/bonds_distribution`.

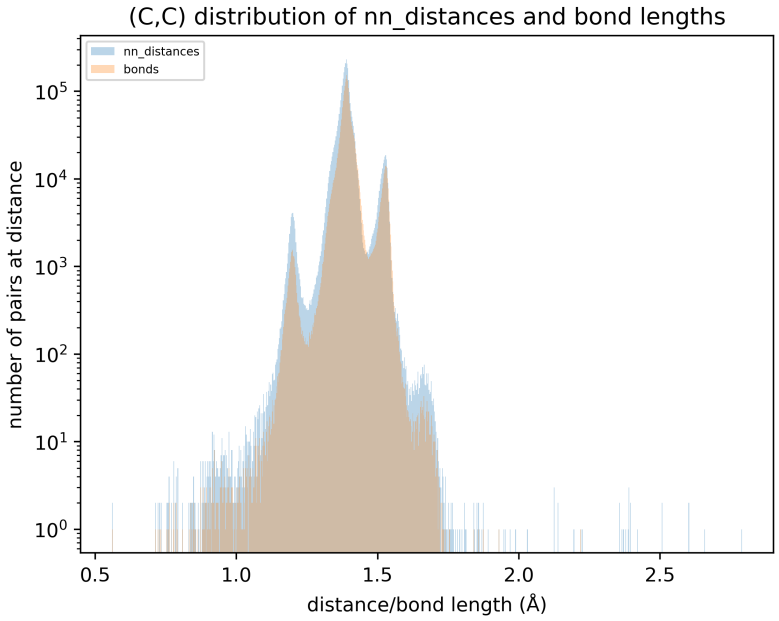

Figure 3.13 The distribution of lengths of all bonds extracted from the centre atom and nearest neighbour type pair (carbon, carbon).

Figure 3.13 shows the distribution of all bonds extracted from the nearest neighbours. Note that in some intervals, for example at a distance slightly less than 1.5Å, the count of bond lengths exceeded the count of

nearest neighbour distances. This is because there exists a slight difference between registered bond lengths and nearest neighbour distances.

If we further split the distribution by bond types, there are mainly 4 types (as introduced in section 2.2) composing most of the bonds: single, double, triple and aromatic. See Figures 3.14-3.17 for distributions of them.

There are 3 peaks in the distribution of all bond lengths. All types of bonds tend to mainly have a specific bond length, forming the peak observed in histograms of double, triple and aromatic bonds, and the highest peak in the histogram of single bonds. It seems that the single bonds are exceptionally forming a secondary peak with a count of approximately 1/10 of that of the main peak. The main peak of each type of bond tends to be the main contributor to the corresponding peak observed in the overall distribution, with double bonds being an exception to be forming the second peak in the overall distribution together with the aromatic

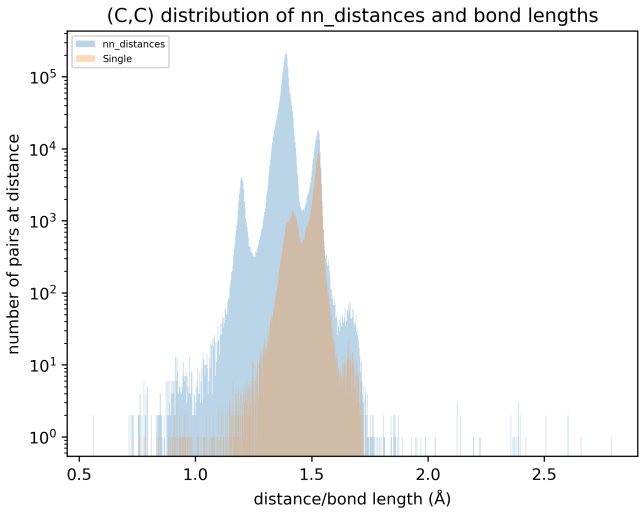

Figure 3.14 Histogram of carbon-carbon single bond lengths

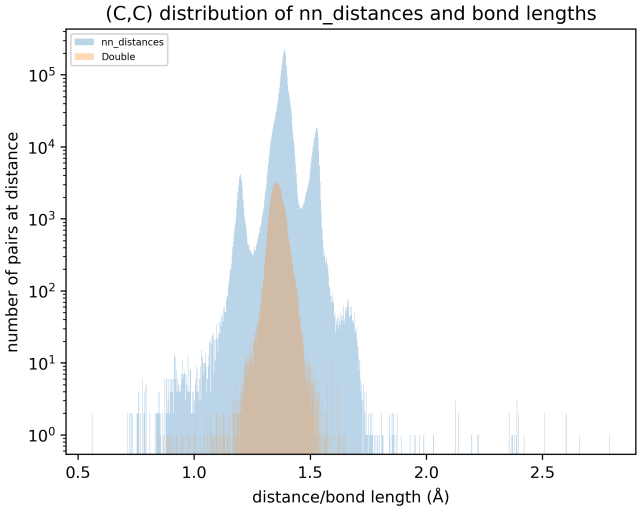

Figure 3.15 Histogram of carbon-carbon double bond lengths

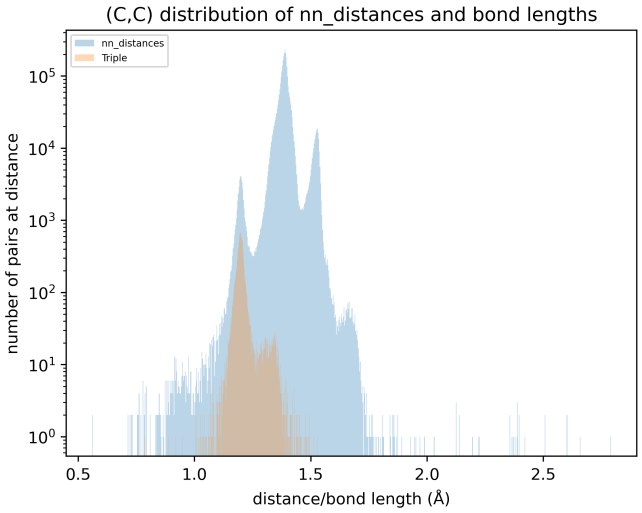

Figure 3.16 Histogram of carbon-carbon triple bond lengths

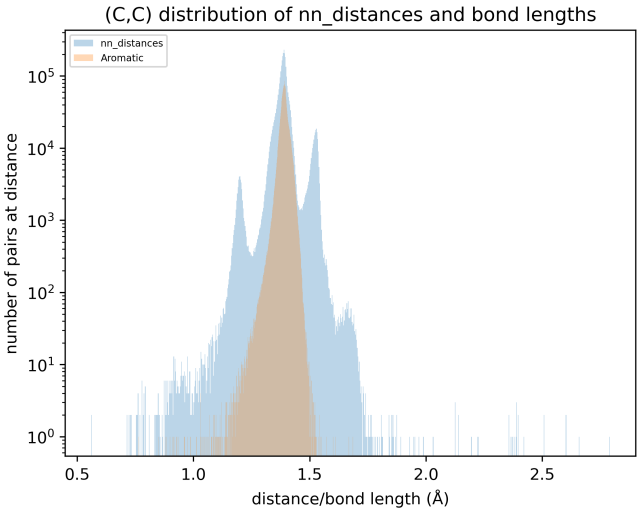

Figure 3.17 Histogram of carbon-carbon aromatic bond lengths

bonds. Note that, the main peaks formed by different types of bonds have the following relationship on the centre of the peak: the typical length of triple bonds is the shortest, followed by double bonds, aromatic bonds and finally single bonds.

In summary, the bond lengths tend to form peaks at some particular distance which is dependent on the pair of elements, and each type of the bond formed between the same pair of elements appears to form typically only one peak, being the main contributor to the corresponding peak in the overall distribution. The same phenomenon can also be observed in the other figures generated.

4. Discussion

4.1 The performance of the model

The model showed an overall accuracy of over 80% on 20 classes, indicating that the model indeed learned chemical knowledge from topological data. Given the high accuracy of the model, the overall likelihood of a crystal structure predicted by the model might show a significant decrease with an implausible atom position, so that it can be distinguished from other structures and eliminated for further computation. In addition, a prediction of 40,000 atoms took 5 seconds to run on a modern 8-core desktop, which suggests the computational complexity of predictions based on this model is remarkably lower than traditional algorithms mentioned in the Introduction. However, the reliability and effectiveness of the model still need to be further tested on actual crystal structure prediction datasets.

The improvement of the model's performance over k was monotonic, but with a reducing rate of increase. The reason for it might be that an increased k indicates an extension of information while a decrease of the relevance of that specific increment of information with the atom because the neighbours are further from the atom and thus less affected by it. This suggests that there should be an optimum value of k even if the computational complexity is neglected, after which the improvement of performance is negligible.

4.2 A possible explanation for the predictive power of the model

With distributions of kth nearest neighbour distances extracted, we found that the pattern, especially the peaks, of the distributions is dependent on k and the pair of elements. The bonds within the nearest neighbours were further extracted, and the distribution of bond lengths of nearest neighbours (carbon, carbon) forming a bond was analysed in depth. It is observed that the distribution of bond lengths tends to form peaks at some distance, depending on the pair of elements. In addition, each type of bond formed between the same pair of elements appeared to have similar lengths, with only one or two adjacent peaks observed in the distribution.

The finding that the bond lengths of the same type of bond formed between the same pair of elements tend to be concentrated in a small interval supports the theory of bonds introduced in the Introduction [19]. Moreover, the peaks formed by different bonds, whose centre indicates a typical length of that specific type of bond, have the same relationship suggested by [19].

Based on the distribution figures discussed above, it might be reasonable for us to speculate that the k-NN distances are carrying chemical information about the centre atom by having a proportion of them composed of chemical bonds, which appeared to be dependent on the element pairs forming the bond, and the type of the bond. While there still exists overlapping features when k=1, we get a combination of bond information

in higher-dimensional space as k increases, which is more likely to be distinguishable from other elements. In addition, it is reasonable to envision that as k increases from 1, the corresponding kth nearest neighbour distances would generally change from a bond length to intra-molecular[11] distances and then inter-molecular distances. Based on the common sense that atoms closer to each other tend to have a stronger interaction, the relevance between the kth-NN difference and the element type of the centre atom is presumably decreasing. It suggests that the improvement of the model's performance might stop after k reach a specific value. This value, being the optimal value of k if we ignore the computational complexity of the model, might seem to be theoretically arguable as it is probably dependent on the number of nearest neighbours that are influential on the centre atom. However, as the number of compounds that could be formed with the atom is too huge to analyse, suggesting it based on experiments is believably to be more efficient and reliable.

5. Conclusion

The neural network model developed in this project demonstrates that neural network models are capable of efficiently predicting the element type of an atom in crystals given its k-NN distances. Despite no chemical a priori knowledge being given, the model learnt the characteristics from completely topological data. Compared with energy calculation approaches, the model efficiently filters out highly improbable structures before further calculation. Additionally, since the model is learning the characteristics solely based on the local topological environment of the atom, it is capable of approximating the likelihood of a previously unknown structure by estimating the likelihood of each element being stable at the given coordination topologies. Based on the experiments, the optimum value of k is suggested as 7, after which the improvement of the model's performance significantly drops. Further testing is needed to estimate the model's reliability and efficiency as a quick filter of crystal structures.

The relationship between k-NN distances and the element was analysed through the extraction of distributions of nearest neighbours, and the bonds formed within the nearest neighbours. A conjecture was proposed that the k-NN distances are conveying information through a combination of potential bond information contained within the kth nearest neighbour distance, which tends to diminish as k increases. There are many directions that further research can possibly explore:

(i) Mapping the k-NN distances and containing bond information of higher-dimensional space (i.e. k>3) would probably be helpful to analyse how they are distinguishable.

(ii) In the **Results** section, it was mentioned that the carbon-carbon single bond is abnormally forming a secondary peak. This might suggest some ubiquitous environment (atoms, molecules etc.) around the single bond that is influencing the bond length.

(iii) It can be observed from the distribution of differences between shortest bond length and nearest neighbour distance that there are some atoms actually not forming bonds with their nearest neighbour, which seems unlikely chemically. It might be worth extracting them for further investigation.

[11] intra-molecular: existing within the same molecule; inter-molecular: existing between molecules

6. BCS Project Criteria

6 outcomes of an honours year project are required by the BCS, the Chartered Institute for IT [23]. They are fulfilled as follows:

- An ability to apply practical and analytical skills gained during the degree programme.
  This project applied machine learning knowledge and programming skills learnt during the programme to construct a neural network model that predicts the type of an atom within a crystal based on their k-NN distances. A further attempt was made to explain the predictive power of the model by analysing the way the k-NN distances may convey chemical information.
- Innovation and/or creativity.
  Despite similar research on the topic, the input data of those research was different from this project. A new model was constructed and tested to solve the problem. In addition, an original conjecture was developed based on an analysis of the data.
- Synthesis of information, ideas and practices to provide a quality solution together with an evaluation of that solution.
  As described in the Introduction section, literature on CSP and solving the problem with machine learning algorithms were reviewed, and the primitive idea of how the solution might work was explained in the Problem Statement. The performance of the model was then evaluated in 3.1. This work further attempted to explain the principle behind the solution through analysis in 4.2.
- Your project meets a real need in a wider context.
  The application of the model was explained in the Problem Statement section.
- An ability to self manage a significant piece of work.
  This project successfully fulfilled the objectives in the project proposal earlier than what was expected by the original plan. Some new ideas were developed during the development of the project and further explored later.
- Critical self-evaluation of the process.
  There are self-reflections on what should have been done and directions for further research explained in the next section.

7. Self Reflection

Project Management

- With original objectives being achieved earlier than expected, the later phase (data analysis) of the project should be properly planned by proposing a new development timeline.
- The demos of each phase should be better documented and version controlled. They are

Programming skills & logical imperfections

- The training and testing set were not rigorously separated in the project. Although the dataset is big enough for a simple model structure, there still might exist overfitting problems.

- The relatively long training time of the model is probably caused by the application of the function that reads the whole k-NN distances dataset into the memory. The training time is likely to be improved by transforming the input pipeline to some iterator.
- Learning chemical knowledge and data processing packages like Pandas helped to develop the scripts effectively and explain the results chemically to propose plausible speculation.

References

[1] L. Colombo, *Solid State Physics*: IOP Publishing, 2021. [Online]. Available: https://dx.doi.org/10.1088/978-0-7503-2265-2.

[2] D. Widdowson, M. M. Mosca, A. Pulido, A. I. Cooper, and V. Kurlin, "Average minimum distances of periodic point sets–foundational invariants for mapping periodic crystals," *MATCH Communications in Mathematical and in Computer Chemistry*, vol. 87, pp. 529-559, 2022.

[3] J. Clark. "Ionic structures." Chemguide.co.uk. https://www.chemguide.co.uk/atoms/structures/ionicstruct.html (Accessed Apr. 22, 2022)

[4] G. M. Day et al., "Significant progress in predicting the crystal structures of small organic molecules–a report on the fourth blind test," *Acta Crystallographica Section B: Structural Science*, vol. 65, no. 2, pp. 107-125, 2009.

[5] G. M. Day and C. H. Görbitz, "Introduction to the special issue on crystal structure prediction," vol. 72, ed: International Union of Crystallography, 2016, pp. 435-436.

[6] A. M. Reilly et al., "Report on the sixth blind test of organic crystal structure prediction methods," *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, vol. 72, no. 4, pp. 439-459, 2016.

[7] K. Ryan, J. Lengyel, and M. Shatruk, "Crystal structure prediction via deep learning," *Journal of the American Chemical Society,* vol. 140, no. 32, pp. 10158-10168, 2018.

[8] A. R. Oganov and C. W. Glass, "Crystal structure prediction using ab initio evolutionary techniques: Principles and applications," *The Journal of chemical physics,* vol. 124, no. 24, p. 244704, 2006.

[9] D. Widdowson. *Average-minimum-distance.* (1.1.6a0). Github. Accessed: 25 Apr, 2022. [Online]. Available at: https://github.com/dwiddo/average-minimum-distance

[10] C. R. Groom, I. J. Bruno, M. P. Lightfoot, and S. C. Ward, "The Cambridge Structural Database," *Acta Crystallographica Section B,* vol. 72, no. 2, pp. 171-179, 2016, doi: doi:10.1107/S2052520616003954.

[11] C. R. Harris *et al.*, "Array programming with NumPy," *Nature,* vol. 585, no. 7825, pp. 357-362, 2020/09/01 2020, doi: 10.1038/s41586-020-2649-2.

[12] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, 2010, vol. 445, no. 1: Austin, TX, pp. 51-56.

[13] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467,* 2016.

[14] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering,* vol. 9, no. 3, pp. 90-95, 2007, doi: 10.1109/MCSE.2007.55.

[15] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research,* vol. 12, pp. 2825-2830, 2011.

[16] *Barkla.* (2020). The University of Liverpool. Accessed: April 26, 2022. [Online]. Available: https://www.liverpool.ac.uk/it/advanced-research-computing/facilities/high-performance-computing/

[17] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, "A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu," *The Journal of chemical physics,* vol. 132, no. 15, p. 154104, 2010.

[18] M. A. Neumann and M.-A. Perrin, "Energy Ranking of Molecular Crystals Using Density Functional Theory Calculations and an Empirical van der Waals Correction," *The Journal of Physical Chemistry B,* vol. 109, no. 32, pp. 15531-15541, 2005/08/01 2005, doi: 10.1021/jp050121r.

[19] Pauling, L. (1960) *The Nature of the Chemical Bond and the Structure of Molecules and Crystals: An Introduction to Modern Structural Chemistry*, 3rd edn. New York: Cornell University Press.

[20] Cordero, B. et al. (2008) 'Covalent radii revisited', *Dalton Transactions*, Issue 32, 2832-2838.

[21] G. Bacon, N. t. Curry, and S. Wilson, "A crystallographic study of solid benzene by neutron diffraction," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences,* vol. 279, no. 1376, pp. 98-110, 1964.

[22] D. Widdowson, private communication, April 2022.

[23] BCS (2021) *BCS, The Chartered Institute for IT*. Accessed April 2022. [Online]. Available at: https://www.bcs.org

[24] G. B. Goh, N. O. Hodas, and A. Vishnu, "Deep learning for computational chemistry," *Journal of computational chemistry,* vol. 38, no. 16, pp. 1291-1307, 2017.

[25] T. K. Patra, V. Meenakshisundaram, J.-H. Hung, and D. S. Simmons, "Neural-Network-Biased Genetic Algorithms for Materials Design: Evolutionary Algorithms That Learn," *ACS Combinatorial Science,* vol. 19, no. 2, pp. 96-107, 2017/02/13 2017, doi: 10.1021/acscombsci.6b00136.

[26] W. B. Park *et al.*, "Classification of crystal structure using a convolutional neural network," *IUCrJ,* vol. 4, no. 4, pp. 486-494, 2017.

Appendices

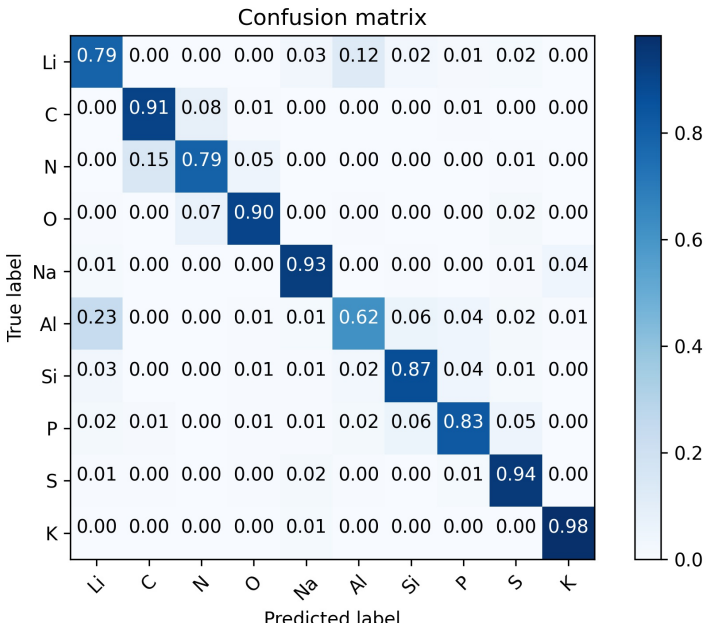

Figure 3.1 The confusion matrix showing the model's performance on predicting the ten element types trained with k=5. Note that the precision calculated is based on the validation set, separated from the training set with a training/validation ratio of 3:1, as it is a demo studying the feasibility of the model structure.

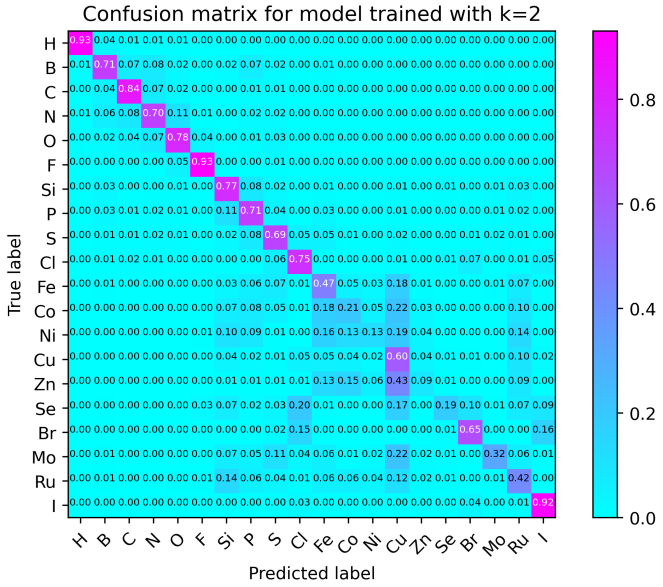

Figure 3.2 (k=2) The confusion matrix of the model trained with 20 elements obtained on the test set.

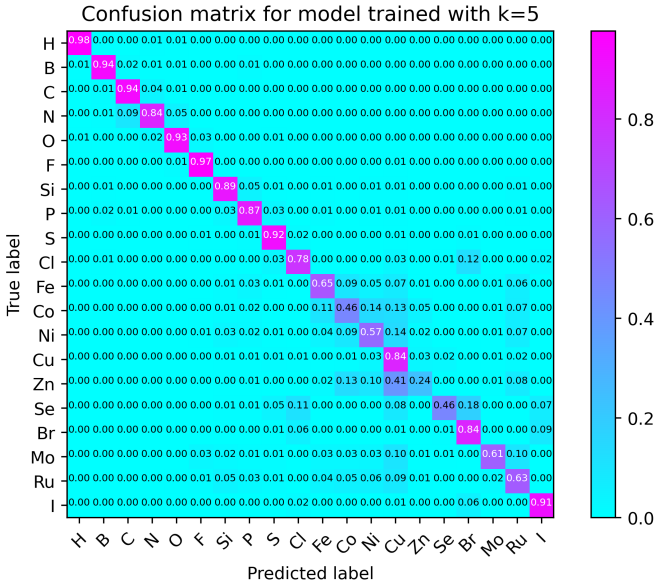

Figure 3.3 (k=5) The confusion matrix of the model trained with 20 elements obtained on the test set.

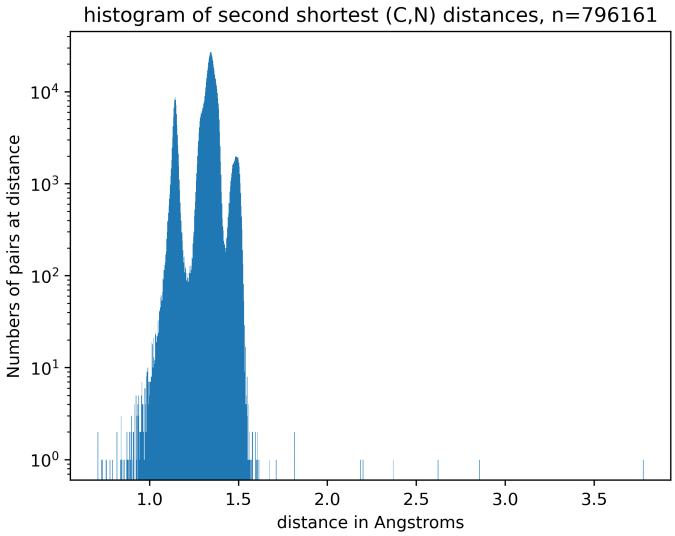

Figure 3.9 A histogram showing the distribution of shortest nearest neighbour distances where the centre atom is a carbon and its nearest neighbour is a nitrogen.

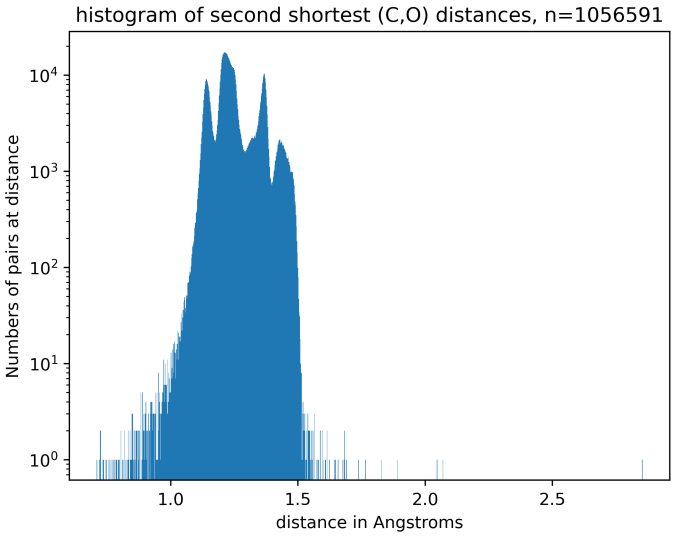

Figure 3.10 A histogram showing the distribution of shortest nearest neighbour distances where the centre atom is a carbon and its nearest neighbour is an oxygen.

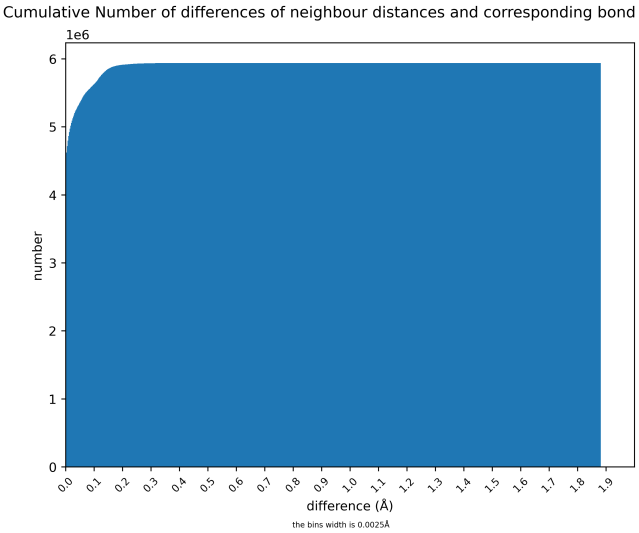

Figure 3.12 A histogram showing the cumulative number of differences falling within the distance so far.