Deep Learning-Based Prediction of Ppb-Level Metal Ions in Drinking Water with Explainable AI

Shaokun Yu

Marquette University

DEEP LEARNING-BASED PREDICTION OF PPB-LEVEL METAL IONS IN DRINKING WATER WITH EXPLAINABLE AI

by

Shaokun Yu

A Thesis Submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science

Milwaukee, Wisconsin

August 2022

ABSTRACT
DEEP LEARNING-BASED PREDICTION OF PPB-LEVEL METAL IONS IN DRINKING WATER WITH EXPLAINABLE AI

Shaokun Yu

Marquette University, 2022

In the research on detecting different ppb-level metal ions in drinking water in multiple areas. Through the block loop gap resonator (BLGR), this kind of metal ion sensor can continuously measure the concentrations of different metal ions in the water sample. The BLGR can classify different ions and measure the Cu and Pb ions in different ppb-level water in the city. According to the different magnitude and phases in continuous frequency in the testing. A better model can be trained by using deep learning, the convolution neural network.

A neural network is often regarded as a black-box model because its very strength in modeling complex interactions also makes its operation almost impossible to explain. However, the neural network is an effective tool to make evaluations and predictions, they can perform very well in this kind of area. The neural network can play a better model learning ability in the case of processing a large number of data, especially in the detecting of the metal composition of water quality in multiple areas, more data points can be better evaluated and predicted.

In this project, we are going to improve the explainability of neural networks applied in detecting ppb-level water systems. We propose to apply the Layer-wise relevance propagation (LRP) algorithm to explain the relevance of different frequency features. This algorithm can highlight the features that contribute most to deep network learning. We can effectively explain which frequencies are more decisive by the high contribution characteristics to the LRP output. We believe that being able to explain machine learning-based decisions greatly improves our analytical capabilities for ppb-level water detecting in this project.

ACKNOWLEDGMENTS

Shaokun Yu

I would like to sincerely thank my major advisor, Dr. Ye Dong Hye, for his guidance and his patient devotion to my research. I also want to thank other committee members, Dr. Lee Chung Hoon and Dr. Frigo Fred, for their support and help during the reviewing of my thesis. Thanks to the entire department for giving me the funding and guiding me during my graduate period. Special thanks to Dr. Ye for giving me the opportunities of research assistant and teaching assistant during the last year at Marquette University. Also special thanks to Dr. Lee, for guiding me and offering me help during the research. Your bits of help were invaluable and hope you can make more achievements in your research area. Finally, I want to thank my family, my girlfriend, and my friends for their continued help and love for me.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Background

To quantify metal ion concentrations in drinking water, microwave measurements and machine learning techniques are provided. As a microwave probe, a new block loop gap resonator (BLGR) is built. The BLGR probe makes no physical contact with the water sample, allowing for on-site/in situ continuous measurements of ppb-level metal ion concentrations. Using a support vector regression approach, the S11 raw data (magnitude and phase) from the BLGR is utilized to identify and estimate metal ion concentrations [1]. In addition, a deep learning regression algorithm was designed. Therefore, it can make better predictions and evaluations compared with support vector regression, making the R square value of deep learning reach above 0.97. Metal ions in water have a profound effect on biological and environmental processes[2], [3]. While metal ions such as sodium (Na), iron (Fe), copper (Cu), and zinc (Zn) are required for a variety of biological functions, metal ions such as lead (Pb), mercury (Hg), and cadmium (Cd) are very dangerous in even trace levels. The concentration of these metal ions in the water influences its hardness. Additional metals, on the other hand, may contribute to the hardness of some waters. Calcium and magnesium are non-toxic and are often absorbed more rapidly than other metals by living organisms. As a consequence, hard water decreases the toxicity of a dangerous ingredient at a given concentration. On the other hand, in soft, acidic water, the same metal concentrations may be more harmful [4].

The article addresses the development of a metal ion sensor using a microwave block loop gap resonator (BLGR). The BLGR analyzes metal ion concentrations in drinking water systems continuously. Because liquid samples of interest are contained inside a glass tube within the resonator, as seen in Fig. 1(d), the BLGR does not come into physical contact with the sample, allowing on-site/in situ monitoring of metal ions in water. The studies demonstrate that the BLGR is capable of categorizing several ion solutions (Na^+, K^+, and Mg^{2+}) in deionized (DI) water and of detecting the Pb ion concentration in city water with a resolution of 1 ppb.

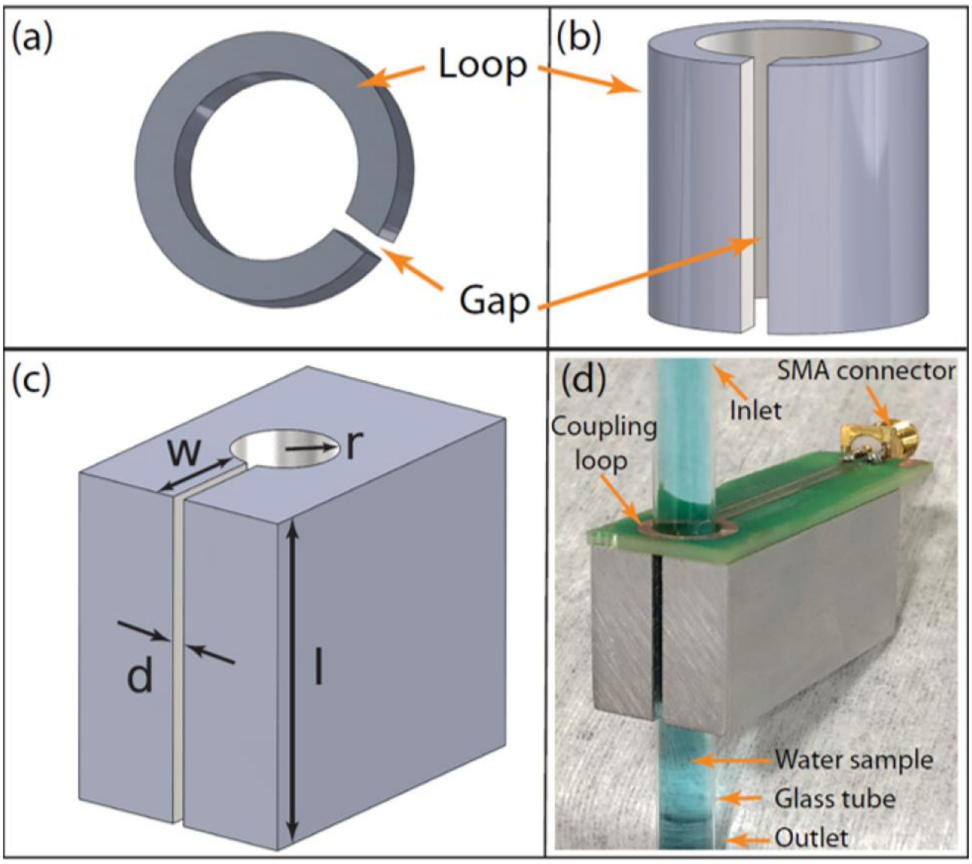


Fig. 1.1 Different resonators are shown schematically: (a) split-ring resonator, (b) slotted tube cavity resonator, (c) BLGR with dimensions, and (d) integration of a BLGR, coupling loop, and water sample for S11 measurement. In the BLGR loop, a 6.3-mm OD glass tube is put. A drop of water is inserted into the glass capillary. The BLGR does not come into direct touch with the water sample. The water sample in the image has been dyed blue to aid with viewing.

One kind of loop gap resonator is an LGR. It's also called a split ring resonator. The gadget is an LC lumped circuit with an inductor and a capacitor [5]. A magnetic field is generated around the loop when an EM pulse is present [6]. The gap carries charges that produce an electric field. The LGR structure is resonant when the magnetic and electric stored energy in the loops are equal. Unusable passive LGR with high filling factor, middling Q, and virtually uniform magnetic field [7], [8]. It is less sensitive to environmental signals than the cavity resonator. The sensor is cheap due to mass production. LGRs are widely used to measure solid electrodynamic properties. A single-loop multiple-gap LGR for 1–10 GHz electron spin resonance spectroscopy has been presented [9], [10].

Unwanted radiation from a single-loop LGR is possible. Wood *et al.* [9] employed a multiloop multigap resonator to decrease radiation loss. Each gap adds a series capacitor to the equivalent circuit, allowing higher frequency ESR spectroscopy. As seen in Fig. 1(a) and (b), the majority of LGR structures are cylindrical, with the loop acting as an inductor and the body sliced into parallel slots to provide a capacitor. Capacitance in a cylindrical LGR is proportional to the difference between the inner and outer radii, while inductance is proportional to the inner radius. Adjust the resonance frequency by varying the inner radius (inductance) or by increasing the distance between the inner and outer radii (capacitance). In the suggested sensor, the outside perimeter is replaced with a rectangular body, as seen in Fig. 1. (c). This is referred to as a BLGR sensor. Not only does the block form body offer mechanical stability for the LGR, but it also provides a stable mounting location for the coupling loop necessary to link RF signals to the LGR. The resonance frequency of the BLGR may be adjusted by altering the loop diameter and gap size (d).

Additionally, the BLGR collects S11 data from ions in a water sample. The BLGR can detect the presence and concentration of certain ions through a change in the reflection coefficient. Ions in water cause the resonator's resonance frequencies to change. The frequency shift is dependent on the number of charges in the liquid sample [10]-[12], its concentration, and its background. Water samples containing a variety of ions and concentrations of a single ion (e.g., Pb) display amplitude and phase fluctuations similar to those seen in S11 data. Variation is seen throughout the whole frequency range but is most evident at the BLGR's resonance frequencies. Gramse et al. [13] investigate just two distinct forms of dopant ions that are substantially stationary in the sample. However, a water sample may include a large number of metal ions.

Additionally, the mobility of these ions might vary significantly depending on the sample's properties. As a result, determining the properties of various ions and concentrations of a single metal ion in a water sample requires a considerably more complicated analysis of the reflection coefficient.

As a result, a machine learning method is utilized to extract features, categorize and quantify various ions. SVR and deep learning regression are used in this project. The comparison between them is shown below:

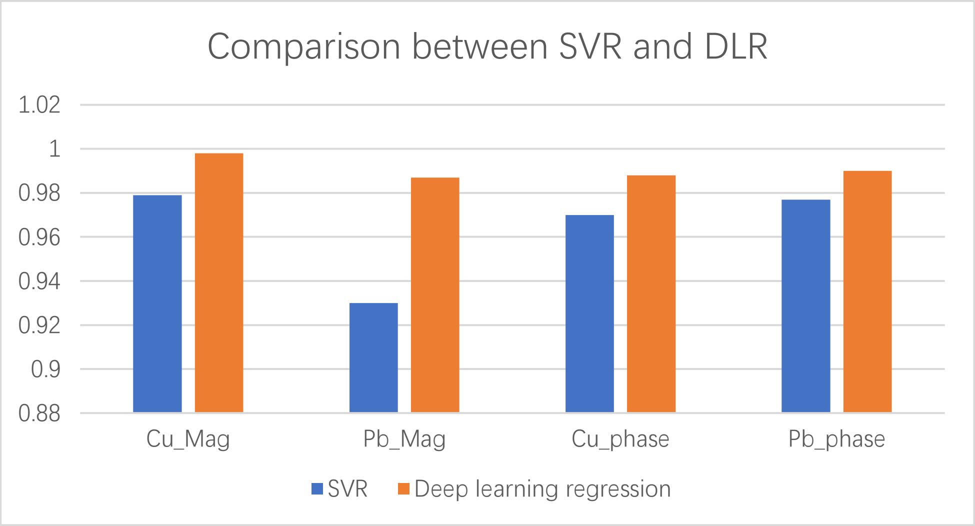


Table 1.1 Comparison between SVR and DLR

More details about deep learning regression will be introduced later. Previously, the S11 raw data is processed using a support vector machine (SVM) or support vector regression (SVR) method. The SVR model is trained on a range of target ion concentrations (e.g., Pb). As a consequence, the algorithm learns how the frequency and phase response components of S11 data vary in response to the presence of certain target ions in a water sample [1]. To make the data more stable and accurate, water quality samples at different times are needed. In this project, 18000 data points are used as training set data, among which 9000 are Cu metal and the other 9000 are Pb metal. In addition, 2,400 data points were tested to assess whether the model was sufficiently accurate. Furthermore, it is necessary to combine data from different dates and retrain the model to make it more stable. Large datasets are needed to train deep neural networks to prevent overfitting and fine-tune parameters (see Fig. 2). The performance of deep neural networks generally improves with increasing amounts of data. In conclusion, deep learning works bette r than ordinary machine learning by learning from large amounts of data.

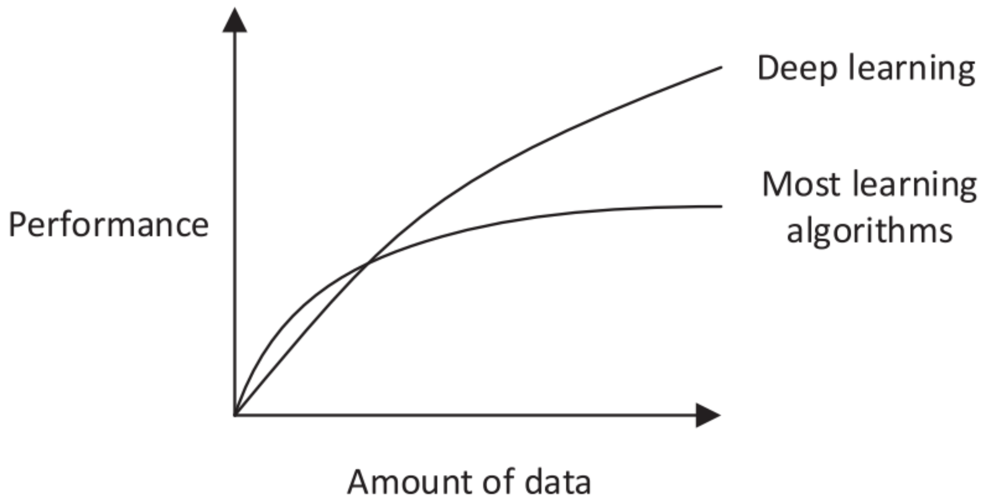


Fig. 1.2 Deep learning and machine learning

The LRP method is used for feature optimization in the deep learning process. Features with greater correlation are picked to improve the current neural network model, which is achieved via the usage of LRP. Specifically, the objective of LRP is to offer an explanation for the output of any neural network in the domain of its input. Using a backwards propagation of the prediction in the neural network, layer-wise relevance propagation (LRP) was used to score features according to their significance to the CNN's predictions, and the results were shown in Table 1. In the model prediction, frequencies with a high magnitude mean relevance score are significant, while frequencies with a magnitude mean relevance score around 0 are less important to the model. This score was used to determine which important elements should be included in the training process.

1.2 Objectives

The objectives of this thesis are shown in the followings:

- Re-adjust the structure of S11 data, and make it adjustable for deep learning.
- For the testing of detecting the metal ions' ppb-level in drinking water, generate a better model and its prediction using deep learning regression.
- Optimize the model and use deep learning algorithms to explain the artificially intelligent.
- Find the most significant frequency range through the dataset and optimize the result.
- Retrain the model and provide a better solution to help the experiments save time and cost of sensors.

1.3 Thesis Organization

- Chapter 2 pre-processing the S11 data split into 2 independent datasets as our input. Then introduce the different kinds of deep learning algorithms used in the project. In addition, use different algorithms to analyze the model and make an evaluation and comparison. Using different parameters in the regression will generate different results. Use explainable AI algorithms to analyze the internal contributions of the model and make a comparison to different explainable AI algorithms in the INNvestigate package.
- Chapter 3 shows the different metal ions in the magnitude and phase conditions. Generate plots to evaluate the model and show experiments results of deep learning.

- Chapter 4 discusses the optimization after LRP is used in the model, the relevance score will guide the important partial range of the frequency features. For different conditions, the result may vary. Also, the new solution may occur after the LRP result shows.
- Chapter 5 concludes the thesis with a summary and some ideas for future works, including the application of 2-D CNN, this solution will merge the datasets in multiple metal ions and have the opppotunitie to find the relationships between ions.
- Chapter 6 shows the reference of the thesis.

2 METHODOLOGY

2.1 Data Structure

2.1.1 Data Extraction and Integration

Data collecting is a significant bottleneck in machine learning and a hot issue of study in several fields. There are primarily two reasons why data gathering has become a crucial concern in recent years. To begin, as machine learning becomes more extensively employed, we are seeing new applications with insufficient labeled data. Second, unlike standard machine learning approaches, deep learning techniques build features automatically, which reduces feature engineering costs but may need higher volumes of labeled data. Notably, modern data collecting research originates not only from the machine learning, natural language processing, and computer vision disciplines but also from the data management field, owing to the critical nature of managing vast volumes of data. Data collection includes acquiring data, labeling it, and improving current data or models. We provide a research landscape for these procedures, offer suggestions for when to employ either approach and propose intriguing research issues. In this project, the metal concentration data we tested in drinking water may be mismatched in in-depth models, which explains the importance of pre-processing data. What's more, dimensionality reduction of data dimension is beneficial to studying the changes of each metal under different conditions, and it will be more expansible in future polymerization studies. The combination of machine learning and data management for data collecting is part of a bigger trend of Big data and Artificial

Intelligence (AI) integration, which creates several new research possibilities.

Four steps of data pre-processing [15] are needed:

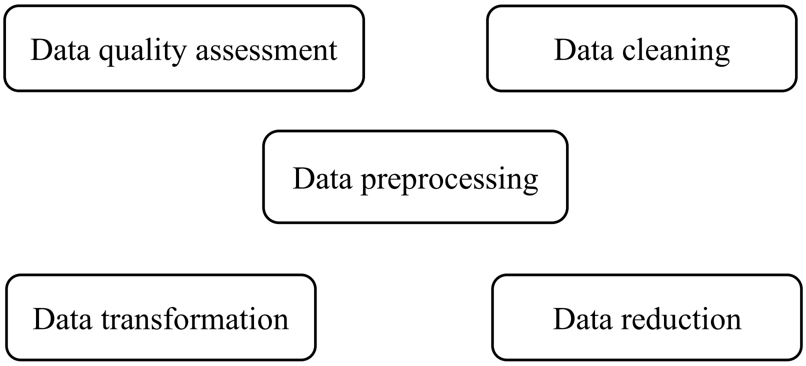


Fig. 2.1 Data preprocessing steps

Several elements influence the effectiveness of Machine Learning (ML) on a particular assignment. The first and most important consideration is the representation and quality of the instance data. In the presence of a large amount of irrelevant and redundant information as well as noisy and inaccurate data, knowledge discovery during the training phase is more difficult to achieve. It is commonly known that data preparation and filtering stages in machine learning issues consume a significant amount of processing time. It is necessary to clean and normalize data before processing it. It is also necessary to extract and pick features from data before processing it.

The data structure in S11 data has three kinds of pieces of information in each datapoint. The brief plot of the structure is shown below:

Frequency	Magnitude	Phase
Frequency _1	Magnitude _1	Phase _1
Frequency _2	Magnitude _2	Phase _2
…	…	…
Frequency _20000	Magnitude _20000	Phase _20000

Table 2.1 Data structure

In this project, data quality inspection and quality cleaning are not necessary, because the data measured with BLGR are very accurate and the probability of error is reduced to a small enough to be ignored. In the first machine learning experiments, the errors in the data were negligible and the model performance was very good. Therefore, data cleaning can also take no action during data preprocessing.

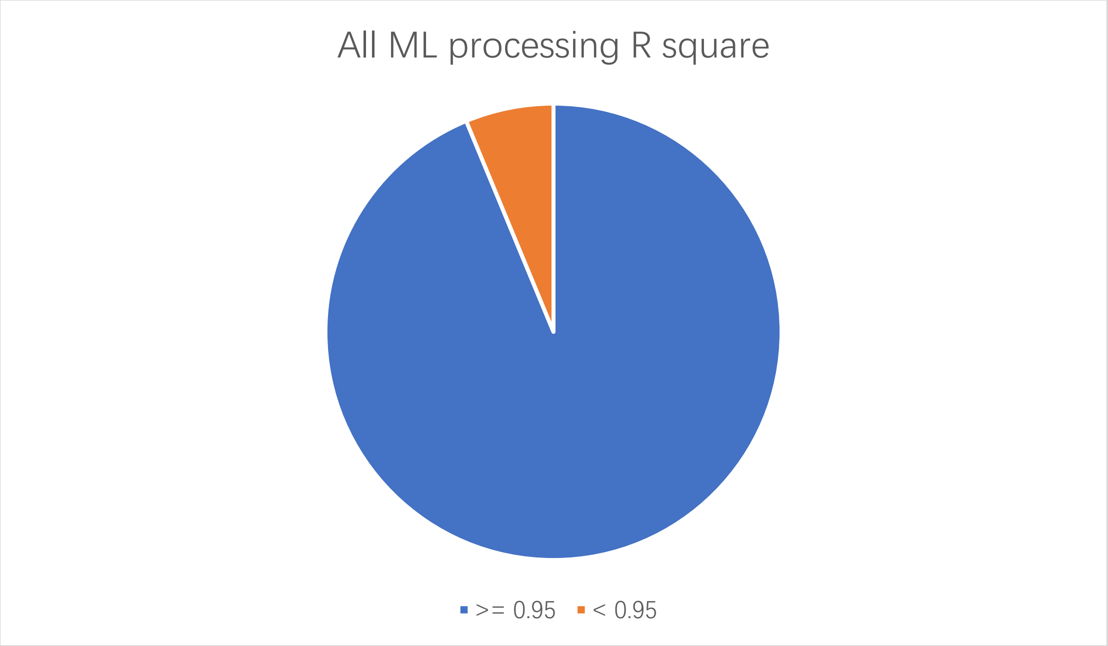


Fig. 2.2 All ML processing R square value

For data transformation and data reduction [16], the downsampling of data is set to an integer value of 8. The sensor may provide both amplitude and phase data for each specified frequency value in the full data point if the sensor is capable of doing so. If the value of each frequency, as well as its accompanying amplitude and phase, were to be used as input to the learning model, it would result in a total of 20,000 frequency characteristics being contributed to the process of learning.

Deep learning research often requires that the quantity of data used by 10 times or more than the number of features to produce a better model. There is no question that 20,000 features is a significant amount of data and is unnecessary for this investigation; thus, suitable feature reduction is required. Researchers did downsample behavior on data in the previous several other learning processes concerning water quality detection, for SVR and SVM to play a greater role in the proper number of features in the appropriate number of features. In this experiment, the value of downsampling is set to 8, which indicates that out of a total of 20000 rows of data, one row out of eight is selected for machine learning. As a result, 20,000 data points will be reduced to 2,500 data points. The following are some of the benefits of using this method:

1. The number of features should not be excessive for the suitable model to be computed successfully and fitted as quickly as feasible after being created.

2. Significantly lower the program's execution time. The more the amount of data you have to deal with, the more difficult it will be to evaluate it, even after cleaning and changing it. Data reduction not only makes the analysis simpler and more precise but also reduces the amount of data that must be stored.

Frequency	Magnitude	Phase
Frequency _1	Magnitude _1	Phase _1
Frequency _8	Magnitude _8	Phase _8
…	…	…
Frequency _20000	Magnitude _20000	Phase _20000

Table 2.2 Dataset after downsampling

For high-dimensional data, the amount of computation required may be too large if the pattern is not preprocessed before being entered into the classifier. In this article, it is necessary to divide the data set into two different evaluation factors. Doing so reduces the complexity of regression and improves performance.

In the process of attribute selection, take frequency as the feature and magnitude as the value under different features. In this way, a new data set is generated, and the process is to reduce the two-dimensional data into two one-dimensional data sets for deep learning. Since the objective of the experiment is to discuss the concentration of specific metal ions at different frequencies, it is necessary to discuss different metal ions separately. At the same time, we can also see the trend of ion concentration of different metals at different frequencies.

The new magnitude dataset is shown below:

	Frequency _1	Frequency _8	…	Frequency _20000
Datapoint_1	Magnitude _1	Magnitude _8	…	Magnitude _20000
Datapoint_2	Magnitude _1’	Magnitude _8’		Magnitude _20000’
…	…	…	…	
Datapoint_9000	Magnitude _1”	Magnitude _8”	…	Magnitude _20000”

Table 2.3 Magnitude dataset after preprocessing for deep learning

From table 2.2, 9000 data points are included in the dataset. This means, there are 9000 testing samples are participated in the training. Also, this kind of format can be applied to different metal ions, such as Cu or Pb.

In this way, another phase table after split (dimensionally reduction) is shown below:

	Frequency _1	Frequency _8	…	Frequency _20000
Datapoint_1	Phase _1	Phase _8	…	Phase _20000
Datapoint_2	Phase _1’	Phase _8’		Phase _20000’
…	…	…	…	
Datapoint_9000	Phase _1”	Phase _8”	…	Phase _20000”

Table 2.4 Phase dataset after preprocessing for deep learning

2.1.2 Normalization

Normalization is an essential approach that is often conducted as a pre-processing step before many Machine Learning models normalize the range of features in the input data set. It is used to normalize the range of features in the input data set [17].

They are employed in the data preparation stage, during which the data is prepared for further processing by one of the data mining and machine learning methods, such as the support vector machine or neural networks, among others [18].

When characteristics of an input data set have considerable disparities across their ranges, or when they are measured in various measurement units (e.g., pounds, meters, miles, and so on), normalization is required to be applied. To illustrate this with an example in this article:

	0.1GHz_magnitude	0.1GHz_phase	0.1003GHz_magnitude	0.1003GHz_phase
Datapoint_1	-3.32592515E-02	1.64681048E+02	-3.33737438E-02	1.64623704E+02

Table 2.5 Data details in “2_1ppb_2000” file

It is necessary to merge the two data sets when considering magnitude and phase together (as in a simple extension study). It was discovered that the particular value of phase was around 160, and the magnitude was approximate -3 in magnitude. No matter what model run on this data set, the phase feature will always take precedence over the magnitude feature and will make a greater contribution to the calculation simply because it has larger values when compared to the magnitude feature in this data set. As a result, normalization is the answer for preventing this issue by translating characteristics into equivalent scales via standardization.

Generally, When it comes to machine learning, feature scaling is one of the most critical data preparation steps. If the data is not scaled, algorithms that calculate the distance between the features are biased towards numerically greater values.

The size of the features has little effect on the performance of tree-based algorithms. Additionally, feature scaling aids in the training and convergence of machine learning algorithms, as well as deep learning algorithms.

The most often used feature scaling strategies are normalization and standardization, which are two of the most widely used procedures.

Normalization (or Min-Max) is a term used to describe the process of Scaling a technique for transforming features into a comparable scale. The new point is computed as follows:

$$x_{new} = \frac{(x - x_{min})}{((x_{max} - x_{min}))}$$

(2.1)

As a result, the range is scaled to [0, 1], or in certain cases [-1, 1]. The n-dimensional data is compressed into an n-dimensional unit hypercube when the transformation is performed geometrically. As a result, it is only beneficial when there are no outliers, as it is incapable of dealing with them.

2.2 Deep Learning Regression

2.2.1 Introduction to CNN

Deep learning approaches such as convolutional neural networks have grown prominent in several computer vision tasks, and they are drawing attention across a wide range of areas, including radiology, due to their ability to learn from large amounts of data.

Using a backpropagation technique, a convolutional neural network is constructed from a variety of building blocks, including convolution layers, pooling layers, and fully connected layers. It is meant to automatically and adaptively learn spatial hierarchies of data.

CNNs are mathematical constructs that are generally constructed of three kinds of layers (or building blocks): convolutional, pooling, and fully connected layers. [19] – [21] Convolutional neural networks (CNNs) are a form of neural network that is built of convolutional neural networks (CNNs). Features are retrieved in the first two layers (convolution and pooling), and the third layer (a fully connected layer) translates the collected features into final output (such as classification) in the third layer (convolution). A convolution layer is critical in CNN, which is made of a stack of mathematical operations, such as convolution, which is a specific sort of linear operation. A convolution layer is also important in image recognition. Because a feature can appear anywhere in an image, CNNs are extremely efficient for image processing. As one layer feeds its output into the next layer, the extracted characteristics may grow increasingly complicated and hierarchical as they move through the layers. Training is the process of optimizing parameters such as kernels. It is carried out in order to reduce the difference between outputs and ground truth labels, and it is accomplished via the use of optimization algorithms like backpropagation and gradient descent, among other techniques.

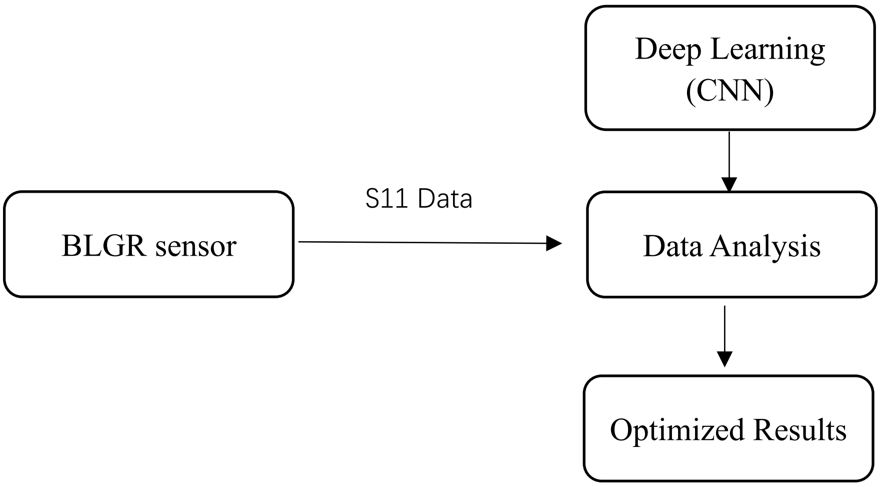


Fig. 2.3 Data processing in machine learning

During this experiment, CNN is employed to assist in the deep learning process. CNN can share the convolution kernel and handle large amounts of high-dimensional input quickly and efficiently. In addition, feature extraction may be carried out automatically in certain instances. When using convolution, features may be extracted from the convolution layer, and the convolution kernel in the convolution layer is very important in extracting the needed features from the convolution layer.

To predict the association between the ppb-level of drinking water and the magnitude and phase values in various metals at different frequencies, a one-dimensional convolutional neural network (CNN) model was developed. When compared to linear models, which struggle with high-dimensional data, this model makes use of the non-linearity of neural networks to better capture the link between the amplitude and phase of the signal and the ppb level of the water in question.

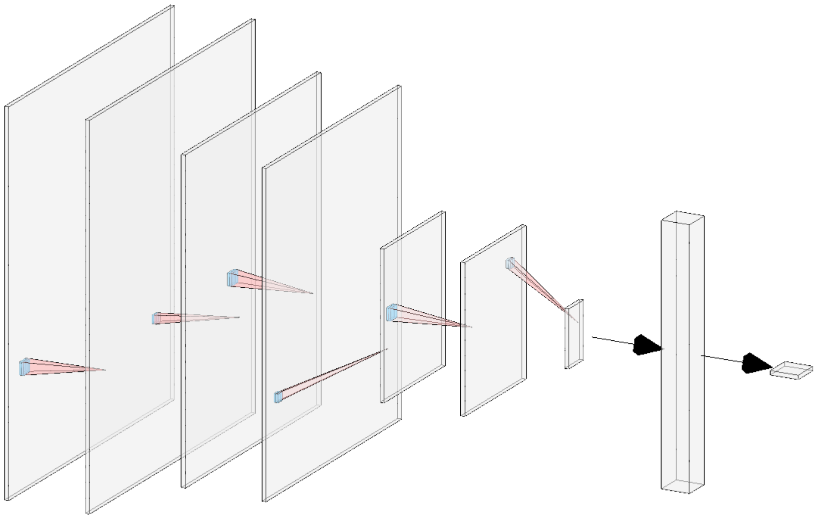


Fig. 2.4 Model architecture

The three major layers of the multilayer convolutional neural network employed are as follows: a 1D convolutional layer with a rectified linear unit (ReLU) activation function and a filter size of 1, followed by a dropout layer with a dropout value of 0.2, and a dropout layer with a dropout value of 0. The number of filters in each of the 1D convolutional layers is 128 in the first two main layers and 64 in the next main layer in the first two main layers of the first main layer. This is followed by a flattening of the model into a 64-neuron dense layer with a ReLU activation function, after which it is completely linked into a single output. The model was built using a mean squared error loss function and an Adam optimizer with a learning rate of 0.001 and a mean squared error loss function. It was decided to use 1D convolutional layers rather than dense fully connected layers because they provided more consistent convergence; however, because the features have no clear spatial or temporal order, a filter size of 3 was used and any type of pooling was used after each layer from the architecture. The architecture was designed to be as simple as possible, and the features have no clear spatial or temporal order. Even attempts to organize the features according to the number of characteristics they include in total and to use a higher filter size failed to generate an improvement in the model's performance. When it came to controlling overfitting, the dropout layers with a 0.2 dropout value were chosen since they outperformed all other strategies in terms of performance.

2.2.2 Comparison between Different Neural Networks

For the most part, ANN relies on weights and an activation function to accomplish its goals. How artificial neural networks (ANN) operate is best explained by the fact that it artificially reconstructs the way a brain's neural network works. When it makes a mistake, it goes back and "modifies" the way it thinks, much as a person would [22].

When using an ANN, the "layers" are rows of data points that are hosted by neurons that are all using the same neural network. When learning, ANN makes use of weights. In ANN, the weights of the neurons are modified after each loop across the neuron. According to the accuracy measured by a "cost function," ANN travels back in time and makes adjustments to the weights.

CNN, on the other hand, does not include any neurons or weights. CNN, on the other hand, applies numerous layers to datasets and filters inputs before analyzing them. The arithmetic layer, the corrected linear unit layer, and the fully connected layer are the three layers in question. The objective of these layers is to comprehend patterns that the network can "see," process data output, and deliver an n-dimensional vector output as a result.

With the help of the n-dimensional output, it is possible to identify different characteristics and relate them to the picture input that was supplied. It may then provide the user with the categorization outcome. However, despite their differences, both approaches rely on measurements of the error to promote learning and both methods create epochs to evaluate the performance of the models that have been constructed.

These characteristics are collected from the input when CNN is used. This makes CNN an excellent choice when thousands of characteristics need to be retrieved. As opposed to needing to measure each and every aspect individually, CNN accumulates these features on its own.

Some picture classification issues become harder to solve when using ANN because 2-dimensional images must be transformed to 1-dimensional vectors before they can be solved. The number of trainable parameters grows exponentially as a result of this. Growing the number of trainable parameters requires more storage and processing power.

To put it another way, it would be prohibitively costly. When compared to its predecessors, CNN's primary benefit is that it automatically determines the most significant characteristics without the need for human intervention. This is why CNN would be an excellent answer for challenges involving computer vision and picture categorization.

After everything is said and done, there are particular occasions in which ANN may be favored above CNN and vice versa. They are both distinct in the way they approach mathematical issues, which allows them to be more effective at addressing certain difficulties.

In general, CNN is a more powerful and accurate method of handling classification issues than traditional methods. The use of artificial neural networks (ANNs) is still dominating for issues where datasets are restricted and picture inputs are not required. CNN, on the other hand, is the most widely used solution for computer vision and image-dependent machine learning issues because of its ability to treat pictures as data.

	ANN	CNN
The value of the model's convergence epoch	20 epochs	7 epochs

Table 2.6 Comparison between the CNN and ANN on the value of the model's convergence epoch

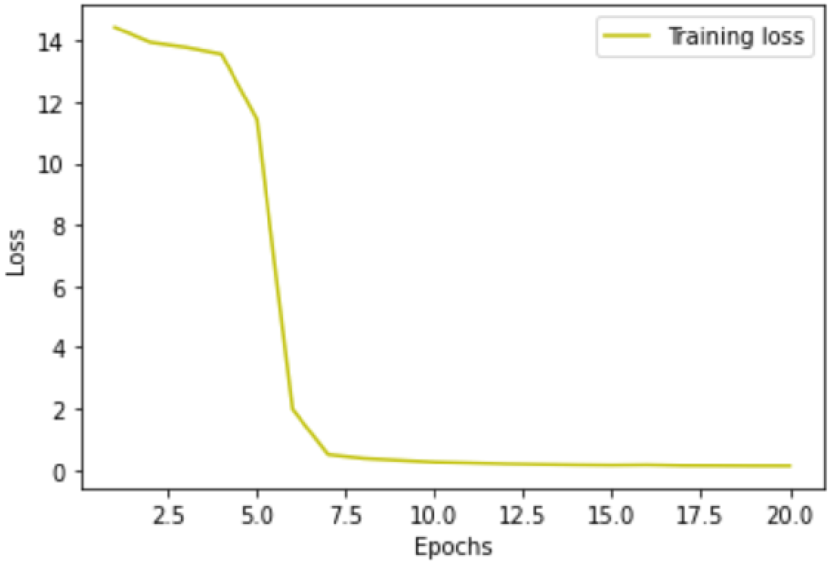


Fig. 2.5 CNN model convergenced after 7 epochs

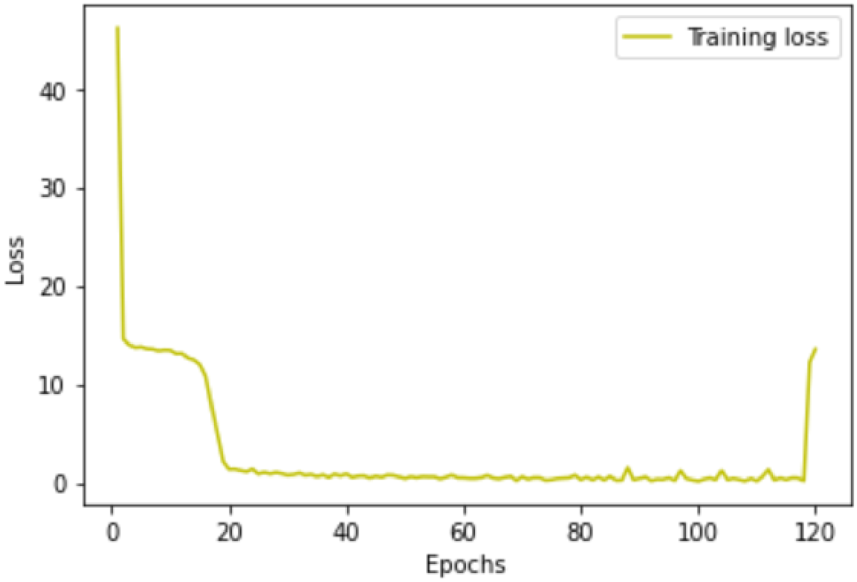


Fig. 2.6 ANN model convergenced after 20 epochs

In the above comparison diagrams of ANN and CNN, it can be observed that, despite the fact that CNN has a long-running period, it can achieve the model convergence state with a minimal number of epochs of learning. The use of artificial neural networks

(ANNs) to solve data regression issues is quite simple. This also implies that it will go through a number of period investigations in order to get experimental data.

2.2.3 Optimize Neural Networks

With the help of the stochastic gradient descent optimization technique, deep learning neural network models are fitted to training data sets.

With the help of the backpropagation of error technique, it is possible to make changes to the weights of the model. When it comes to fitting neural networks, the combination of optimization and weight update algorithm has been carefully selected since it is the most efficient method currently available.

It is, however, feasible to employ other optimization strategies to fit a neural network model to a training dataset in order to improve performance. This may be a great exercise for learning more about the operation of neural networks and the essential role that optimization plays in practical machine learning applications. It may also be necessary for neural networks with non-traditional model designs and non-differentiable transfer functions, among other applications.

Deep learning, often known as neural networks, is a kind of machine learning that is very versatile.

Models built of nodes and layers that are inspired by the structure and function of the brain are used to create these models. For classification or regression predictive modeling, a neural network model operates by propagating an input vector through one or more layers until it produces a numeric output that may be used to make decisions about the input vector.

In order to train a model, it must be regularly exposed to instances of both input and output, with the weights adjusted to reduce the inaccuracy of the model's output when compared to the predicted output. The stochastic gradient descent optimization technique is used to do this. The weights of the model are modified in accordance with a particular calculus rule that allocates error proportionately to each weight in the network, as shown in the figure. The backpropagation algorithm is used to do this.

When it comes to training neural network models, the stochastic gradient descent optimization approach with weight updates done via backpropagation is the most effective method. The use of a neural network for training purposes is not the only option available.

When training a neural network model, it is possible to utilize any arbitrary optimization technique that is available.

Thus, a neural network model architecture may be created, and an optimization technique can be used to identify the set of weights for the model that results in the lowest possible prediction error or the best possible regression assessment for the network.

The Adam optimization approach is a stochastic gradient descent variant that has lately gained traction in computer vision and natural language processing.

Adam is an optimization technique that may be used to iteratively update network weights based on training data instead of the traditional stochastic gradient descent approach.

Adam isn't like traditional stochastic gradient descent. For all weight updates, stochastic gradient descent uses a single learning rate (called alpha), which does not fluctuate throughout training. Each network weight (parameter) has its own learning rate, which is adjusted as learning progresses.

The technique creates an exponential moving average of the gradient and the squared gradient, and the decay rates of these moving averages are controlled by the parameters beta1 and beta2. Moment estimations are biased towards zero when the moving averages' initial value and beta1 and beta2 values are near 1.0 (preferred). This prejudice is eliminated by computing biased estimates first, then bias-corrected estimates.

The alpha factor, also known as the learning rate or step size, is a measure of how quickly a person learns something new. The fraction of time in which weights are revised (e.g. 0.001 in the experiment). In general, higher starting learning rates (e.g., 0.3) result in a quicker learning period before the rate is revised. Smaller values (for example, 1.0E-5) cause learning to be significantly slowed during training beta1, which is the exponential decay rate for the initial moment estimations (e.g. 0.9 in the experiment).

2.3 Explainable AI with using LRP

2.3.1 Introduction

In the field of explainable machine learning, Layer-wise Relevance Propagation (LRP) is one of the most often used approaches (XML) [23]. Specifically, the objective of LRP is to offer an explanation for the output of any neural network in the domain of its input. In a nutshell, what LRP accomplishes is that it makes use of the network weights and neural activations generated by the forward-pass to propagate the output back through the network all the way up to the input layer, as seen in the diagram below: There, you can see which pixels made a significant contribution to the final result. In this case, we'll refer to the magnitude of each pixel's or intermediary neuron's contribution as "relevance" values [24].

When using LRP, the magnitude of each output y is preserved throughout the backpropagation process, and it is equal to the sum of the relevance maps R of all the input layers. This condition remains true for any successive layers j and k, as well as for the input and output layers when transitivity is used.

Relevance on a layer-by-layer basis Using deep Taylor decomposition as a foundation, propagation is a technique of interpretation (DTD). In order to assign the correlation of the output backward through a pre-trained network and to assess the contribution of nodes to classification, the approach makes use of DTD technology. In order to acquire the correlation of each layer, it is necessary to propagate the correlation to the next layer, taking into account the activation degree and network weight. Using the same dimensions as the input picture, the interpreter generates a pixel-level heat map, which allows the user to see the most critical portions of the input image that contribute to the chosen category.

2.3.2 Theory

It is the general goal of the pixel-wise decomposition to calculate the contribution of each feature or single pixel to the prediction f. This is accomplished in two ways (x). In this way, it can determine for each dataset which features or pixels are responsible for how much of a positive or negative categorization result is obtained. If you look at this picture, the image has been turned into a feature vector, and the classifier will identify it and assign it to either the cat or the no-cat category. Furthermore, the output f(x) is a summation of the characteristic scores. In this image, the sum of the feature relevance demonstrated that the image shown is that of a cat [25] [26].

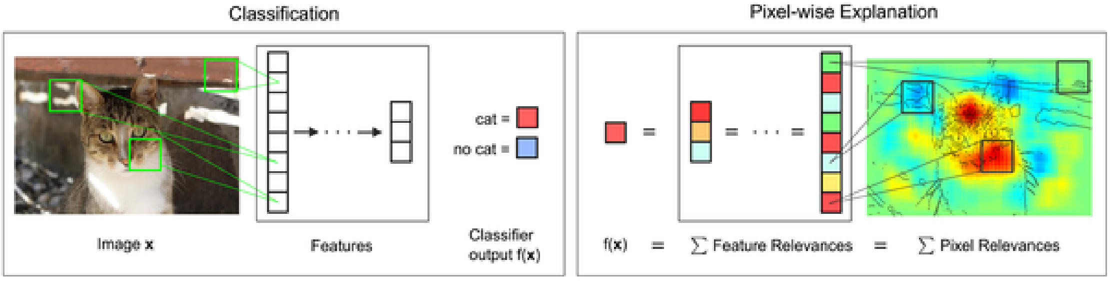


Fig. 2.15 Classification and Pixel-wise Explanation

$$f(x) \approx \sum_{d=1}^{v} R_d$$

(2.3)

Next, we will introduce the LRP. In the general introduction, it can be seen the first layer is the input and the last layer is the output which is the real-valued prediction. The

L means the number of layers and the idea is to find the relevant score $R_d^{(l)}$ in each dimension, and find closer to the first layer, step by step. For example, the figure shows a neuron network with contains the neurons i and weights w, for each neuron we can get the output from the activation function. For the right part of the figure, this is the processing of the LRP, the $R_i^{(l)}$ is the relevance of neuron i which is to be computed. For a better understanding of the figure, we introduce the messages $R_{i\leftarrow j}^{(l,l+1)}$, are messages which need to be computed, such that the layer-wise relevance in this equation is conserved. For example, the R sub 2 super 1 means the input neuron for neurons 4,5,6. It means the R sub 2 super 1 is the sum of the 3 messages from 4,5,6. Similarly, Neuron 3 is an input neuron for 5, 6. Neurons 4, 5, and 6 are the input for neuron 7. For each neuron i, research would like to compute a relevance Ri. We initialize the top layer relevance $R_7^{(3)}$as the function value, thus $R_7^{(3)} = f(x)$. Then we have $R_7^{(3)} = R_4^{(2)} + R_5^{(2)} + R_6^{(2)}$. Also, $R_4^{(2)} + R_5^{(2)} + R_6^{(2)} = R_1^{(1)} + R_2^{(1)} + R_3^{(1)}$.

$$f(x) = \cdots = \sum_{d\in l+1} R_d^{(l+1)} = \sum_{d\in l} R_d^{(l)} = \cdots = \sum_{d} R_d^{(1)} \tag{2.4}$$

Where $R_d^{(l)}$ is the target relevance score.

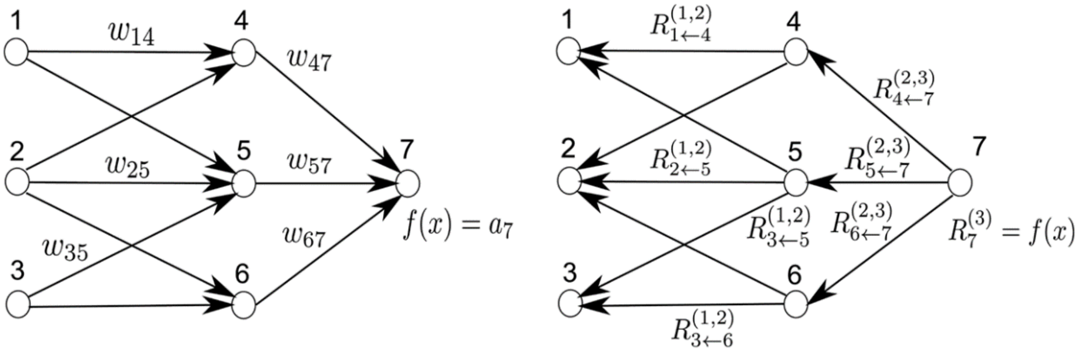

1
w_{14}
4
w_{47}
2
w_{25}
5
w_{57}
7
w_{35}
w_{67}
$f(x) = a_7$
3
6
1
$R^{(1,2)}_{1\leftarrow 4}$
4
$R^{(2,3)}_{4\leftarrow 7}$
2
$R^{(1,2)}_{2\leftarrow 5}$
5
$R^{(2,3)}_{5\leftarrow 7}$
7
$R^{(1,2)}_{3\leftarrow 5}$
$R^{(2,3)}_{6\leftarrow 7}$
$R^{(3)}_7 = f(x)$
3
6
$R^{(1,2)}_{3\leftarrow 6}$

Fig. 2.16 Layer-wise relevance propagation

Where:

Messages are $R^{(l,l+1)}_{i\leftarrow j}$, e.g. $R^{(3)}_7 = R^{(2)}_4 + R^{(2)}_5 + R^{(2)}_6$

Next, here is the sum of incoming messages:

$$R^{(l)}_i = \sum_{k: i \; is \; input \; for \; neuron \; k} R^{(l,l+1)}_{i\leftarrow k} \tag{2.5}$$

As knows, a concept has been created: the message, which is between the neuron i and j which is in a different layer, and the message can be sent along with the network. Therefore, we define the relevance of any neuron except neuron 7 as the sum of incoming messages(2.5)

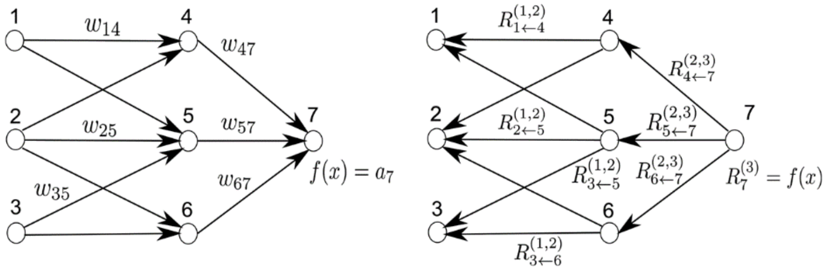

1
w_{14}
4
w_{47}
2
w_{25}
5
w_{57}
7
w_{35}
w_{67}
$f(x) = a_7$
3
6
1
$R^{(1,2)}_{1\leftarrow 4}$
4
$R^{(2,3)}_{4\leftarrow 7}$
2
$R^{(1,2)}_{2\leftarrow 5}$
5
$R^{(2,3)}_{5\leftarrow 7}$
7
$R^{(1,2)}_{3\leftarrow 5}$
$R^{(2,3)}_{6\leftarrow 7}$
$R^{(3)}_{7} = f(x)$
3
6
$R^{(1,2)}_{3\leftarrow 6}$

Fig. 2.17 Layer-wise relevance propagation concept

For example $R_3^{(1)} = R_{3\leftarrow 5}^{(1,2)} + R_{3\leftarrow 6}^{(1,2)}$, because neuron 7 is the last layer so we define $R_7^{(3)} = f(x)$ as above. In this equation, the terms *input* and *source* have the meaning of being input to another neuron in the direction as defined during classification time, For example in the figure, neurons 1 and 2 are inputs and sources for neuron 4, neurons 2 and 3 are inputs and source for neuron 6, the neuron 6 also is the *sink* for neurons 2 and 3. So the LRP also be satisfied by the following statement, such as, $R_7^{(3)} = R_{4\leftarrow 7}^{(2,3)} + R_{5\leftarrow 7}^{(2,3)} + R_{6\leftarrow 7}^{(2,3)}$.

In general, this condition can be expressed as:

$$R_k^{(l+1)} = \sum_{i:\, i\ is\ input\ for\ neuron\ k} R_{i\leftarrow k}^{(l,l+1)} \tag{2.6}$$

The difference between equation (2.6) and equation (2.5) is that in the condition (2.6) the sum runs over the sources at layer *l* for a fixed neuron *k* at layer *l*+1, while in the definition (2.5) the sum runs over the sinks at layer *l*+1 for a fixed neuron *i* at a layer *l*. When using (2.5) to define the relevance of a neuron from its messages, then (2.6) is a sufficient condition to ensure that equation (2,7) holds. Summing over the left side in equation (2.6) yields

$$f(x) = \cdots = \sum_{d\in l+1} R_d^{(l+1)} = \sum_{d\in l} R_d^{(l)} = \cdots = \sum_{d} R_d^{(1)}$$

$$\sum_k R_k^{(l+1)} = \sum_i \sum_{k:i\ is\ input\ for\ neuron\ k} R_{i\leftarrow k}^{(l,l+1)}$$

$$= \sum_i R_i^{(l)}$$

(2.7)

It means the sum of relevance *k* in the *l+1* layer equals the sum of relevance neuron *i* in *l* layer.

Now we can conclude an LRP formula by using the messages $R_{i\leftarrow k}^{(l,\,l+1)}$. The LRP should reflect a relevance score to see which feature is important or not. We know a neuron *i* inputs $a_i w_{ik}$ to neuron *k*, provided that *i* have a forward connection to *k*. Thus, we can rewrite it to this format below, which h means all neurons at the L layer.

$$R_{i\leftarrow k}^{(l,\,l+1)} = R_k^{(l+1)} \frac{a_i w_{ik}}{\Sigma_h a_h w_{hk}}$$

(2.8)

By using the LRP as the essential theory, we can apply it to use different algorithms, like CNN, we can reduce the number of features by using LRP to compute the different relevance scores and pick the most important features to improve our performance in the network.

2.3.3 Application with Using INNvestigate

Recent years have seen deep neural networks revolutionize a wide range of machine learning application fields, and they are now integral to a large number of essential decision and prediction processes [27].

As a result, it is critical that domain experts be able to comprehend and assess the actions and predictions of neural network designs, even those that are very complicated in design. The fact is that neural networks are often considered black boxes, notwithstanding the reasoning presented. There have been several proposals to address this problem, but the absence of reference implementations frequently makes a systematic comparison of the techniques time-consuming and inconvenient to do effectively. A common interface and out-of-the-box implementation for numerous analytic techniques, including the reference implementation for LRP methods, are provided by the presented library iNNvestigate, which tackles this problem.

The presence of a high-dimensional dataset with a small number of samples increases the likelihood of the model is overfitted, which is undesirable (often referred to as the curse of dimensionality). A number of first efforts at feature reduction revealed that deleting low impact frequencies from the dataset might increase CNN performance by a significant margin. Using a backward propagation of the prediction in the neural network, layer-wise relevance propagation (LRP) was used to score features according to their significance to the CNN's predictions, and the results were shown in Table 1. In the model prediction, the frequency with a high relevance score is important, but the frequency with a mean relevance score close to zero is less relevant to the model prediction. This score was used to determine which important elements should be included in the training process.

The LRP assigns recursively to each neuron an input relevance that is proportional to the contribution of the neuron output to the total contribution of the multiple neurons output. The investigate LRP method can rate a total of 2500 frequency characteristics, which is a significant number. This is done in order to examine the contribution to the deep learning network made at various frequencies of operation. A better model may be developed, and the amount of time and money spent on research can be decreased by retaining the frequency of high segmentation while removing the frequency of low scores or zero scores.

2.4 Application of Cross Validation in SVR

The supervised learning technique Support Vector Regression is used to predict discrete values. SVMs and Support Vector Regression are both based on the same premise. SVR's primary concept is to identify the optimum fit line. The best fit line in SVR is the hyperplane with the greatest number of points.

The SVR, unlike other regression models, aims to fit the best line within a threshold value, rather than minimizing the error between the actual and projected value. The distance between the hyperplane and the boundary line is the threshold value. SVR's fit time complexity grows more than quadratically with the amount of samples, making it difficult to scale to datasets with more than a few tens of thousands of samples.

Grid search refers to a technique used to identify the optimal hyperparameters for a model. Unlike parameters, finding hyperparameters in training data is unattainable. As such, to find the right hyperparameters, create a model for each combination of hyperparameters.

Grid search is thus considered a very traditional hyperparameter optimization method since we are basically “brute-forcing” all possible combinations. The models are then evaluated through cross-validation. The model boasting the best accuracy is naturally considered to be the best.

The performance of the models is evaluated using cross-validation. Cross-validation assesses how well a model generalizes to a new dataset. To gain a good assessment of how well a predictive model works, we employ cross-validation.

An independent dataset and a training dataset are created using this technique. After that, divide a single dataset into two sets. Folds are a kind of partition that is the same size as the other partitions. The model in question is trained on all folds except one.

The model is then tested using the excluded fold. This method is continued until all test folds have been utilized. The model's average performance over all folds is then used to estimate its performance.

Then according to the different parameters, use gridsearch to find best parameters series:

kernel	C	gamma	# cv
rbf	0.1	1	3
	1	0.1	
	10	0.01	
	100	0.001	
	1000	0.0001	

Table 2.7 Gridsearch parameters

Then the best parameter series will be generated. Fro different conditions, it will generate four different best parameter series. Finally, the SVR r square score will be computed.

3. EXPERIMENTS RESULTS

3.1 Evaluation and Comparison

The support vector regressor was employed in the previous trials, and the results were positive. It was decided to employ a deep neural network this time, which resulted in improved performance and assessment. A magnitude model and a phase model were produced throughout the deep learning processing, which was done according to magnitude or phase. Aside from that, the concentration of metal ppb in water was predicted by the values of these two variables at various frequencies, and the prediction was assessed in relation to the ground truth concentration of ppb. R Square and Mean Square Error are two metrics that may be used to assess model accuracy and the performance of a regression model.

It is possible to determine how near a regression line is to a collection of points by calculating the mean squared error (MSE). Specifically, it achieves this by squaring all of the distances between points and the regression line (these distances are referred to as the "errors"). The squaring is required in order to eliminate any negative indications. Larger discrepancies are also given greater weight as a result of this. Because you're calculating the mean squared error of a collection of mistakes, the term "mean squared error" is used. It is predicted that the MSE will be lower in the future.

It is necessary to calculate MSE using the following formula:

$$MSE = \frac{\sum(y_i - \hat{y_i})^2}{n} \tag{2.2}$$

	Cu_magnitude	Cu_phase	Pb_magnitude	Pb_phase
MSE	0.0277	0.1568	0.1826	0.1326

Table 3.1 MSE of different conditions using different metal ions in magnitude and phase.

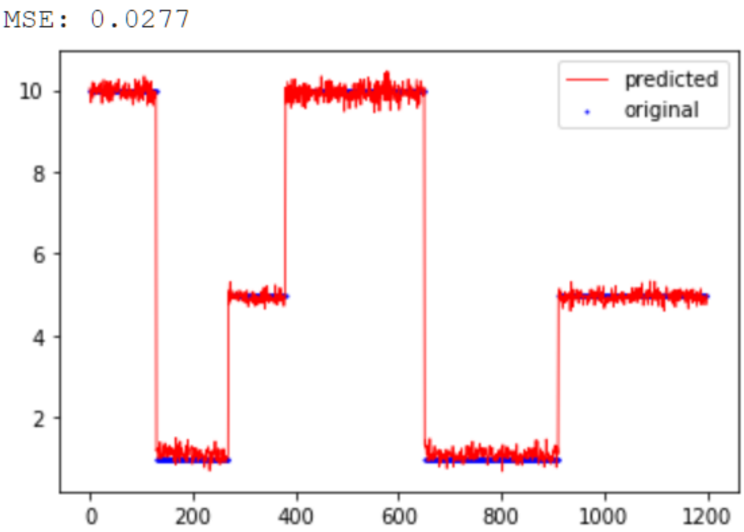


Fig. 3.1 Cu magnitude prediction

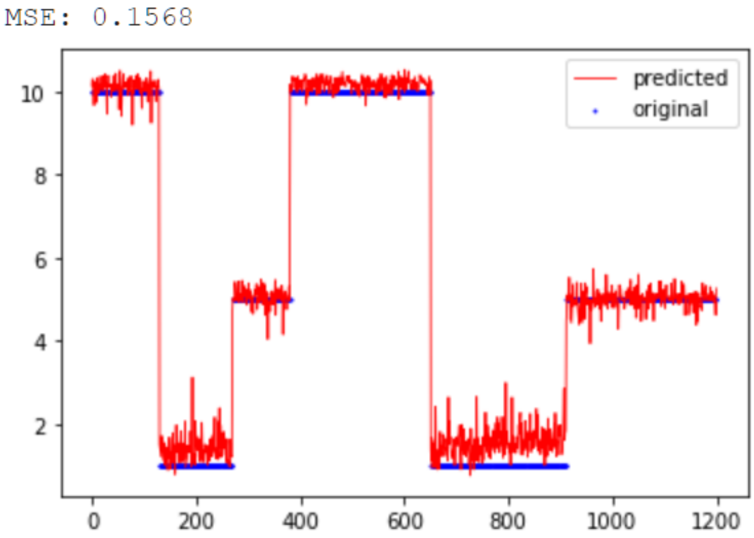
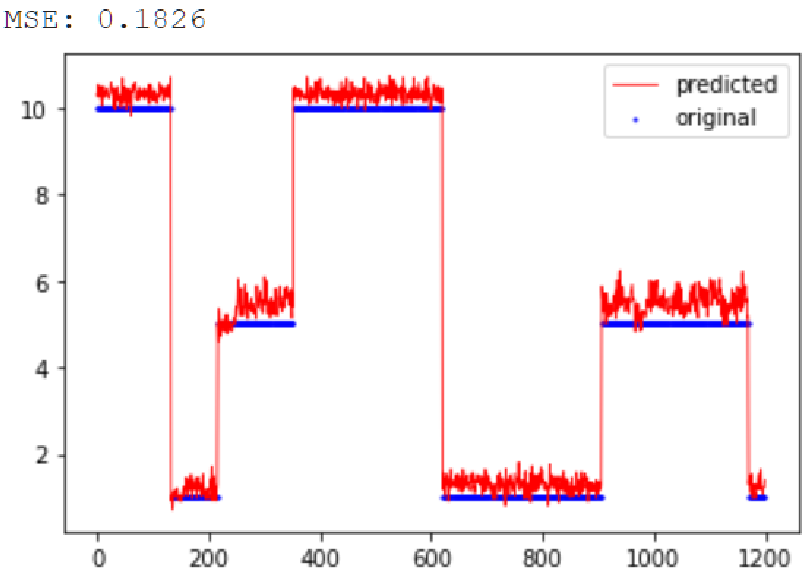
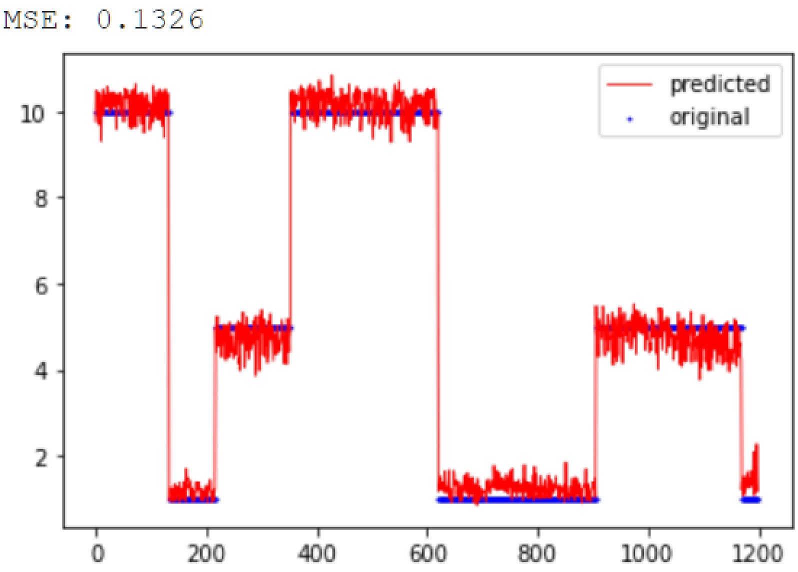


Fig. 3.2 Cu phase prediction



```
MSE: 0.1826
```

Fig. 3.3 Pb magnitude prediction



```
MSE: 0.1326
```

Fig. 3.4 Pb phase prediction

All data points have dependent variables that are just 1,5,10, according to the dependent variables in the chart. Because of this, only three PPB grades were evaluated in this trial. There are 1200 data points in each figure, which are considered data sets. The red line indicates the anticipated value, and the blue line represents the PPB concentration that the data should have had if the predictions were correct. The model makes accurate predictions according to different ppb levels successfully. On the graph, it can be observed that there is very little discrepancy between the anticipated and actual values. In particular, it is projected that the concentration of PPB will approach or surpass 2 in the Cu phase condition when Cu ions are present in low quantities. However, it has little impact on the experimental findings since 2 PPB may be treated as 1 PPB without having any effect under the big PPB difference between 1,5 and 10 and so has little impact on the results. In a similar vein, as more PPBS are tested, the model produces more precise predictions.

When used in a regression model, the R-squared (R2) statistic reflects the percentage of the variation for a dependent variable that can be explained by an independent variable or by a combination of independent variables. When compared to correlation, which indicates how strong a link between an independent and dependent variable is, R-squared tells how much the variance of one variable explains how much the variation of the other variable. For example, if the R2 of a model is 0.50, then the inputs to the model can explain nearly half of the variance in the observed variation.

According to the R-Squared statistic, how much variance in a dependent variable can be explained by the independent variables in a regression model is shown by the R-

Squared statistic.

Here is the formula for R-squared:

$$R^2 = 1 - \frac{SS_{regression}}{SS_{total}}$$

(2.3)

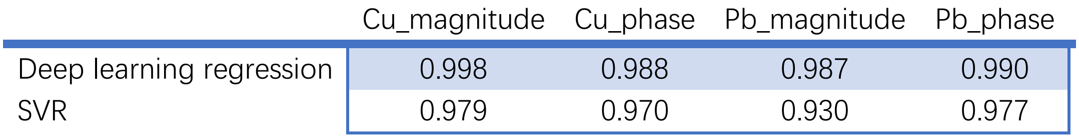
Where:

$SS_{regression}$ is the sum of squares due to regression (explained sum of squares)

SS_{total} is the total sum of squares

This statistic reflects how well the regression model describes the data that was used for modeling by computing the sum of squares owing to regression. The total sum of squares (TSS) is a statistic that quantifies the variation in observed data (data used in regression modeling).

Here are the different R square values of different conditions and the comparison between deep learning regression and SVR:



	Cu_magnitude	Cu_phase	Pb_magnitude	Pb_phase
Deep learning regression	0.998	0.988	0.987	0.990
SVR	0.979	0.970	0.930	0.977

Table 3.2 Difference R square values of different conditions and the comparison between deep learning regression and SVR

It can be observed that the test involves 9000 data points in each example, which is a significant number. It can be observed from the interaction between various metal ions, their magnitude, and their phase that there is not much difference between forecast and ground truth in the assessment process. Using a deep learning neural network in comparison to an SVR model, it can be shown that the R Square of the deep learning network is more than 0.98, while the prediction level of the SVR model is greater than 0.93. This demonstrates that the concentration of metal ions in water can be predicted more correctly with the aid of deep learning, resulting in a more stable model as a result. Here are the deep learning regression comparison between ground truth and predictions in different conditions:

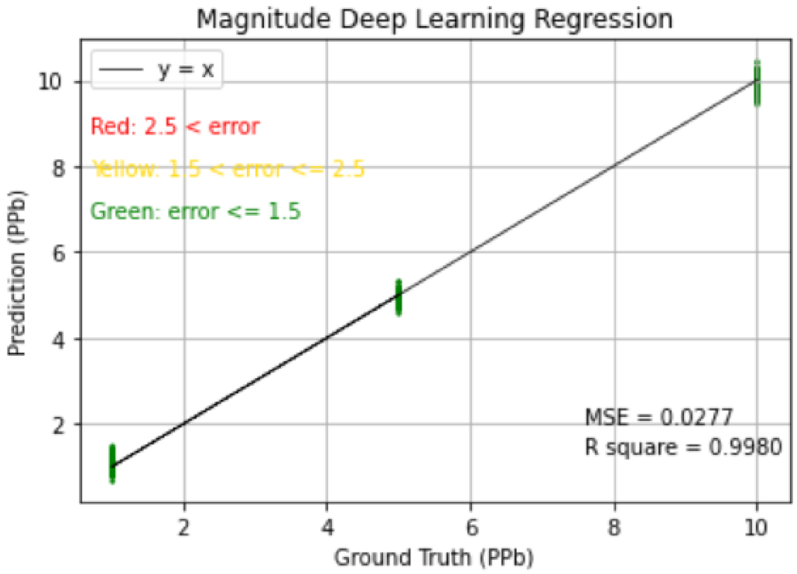


Fig. 3.5 Cu magnitude prediction and comparison

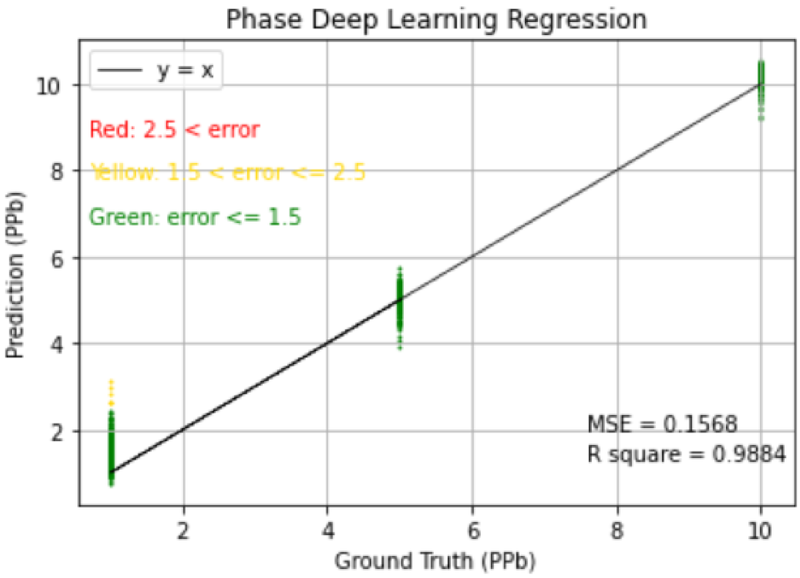


Fig. 3.6 Cu phase prediction and comparison

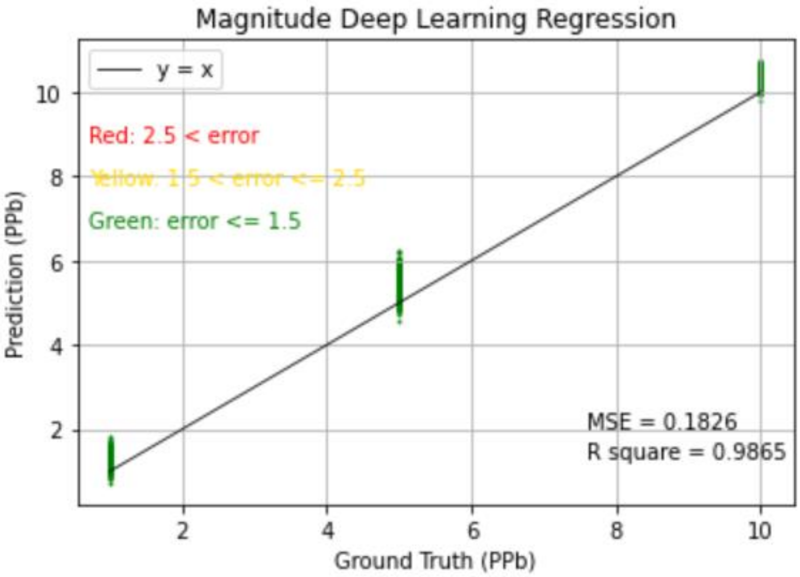


Fig. 3.7 Pb magnitude prediction and comparison

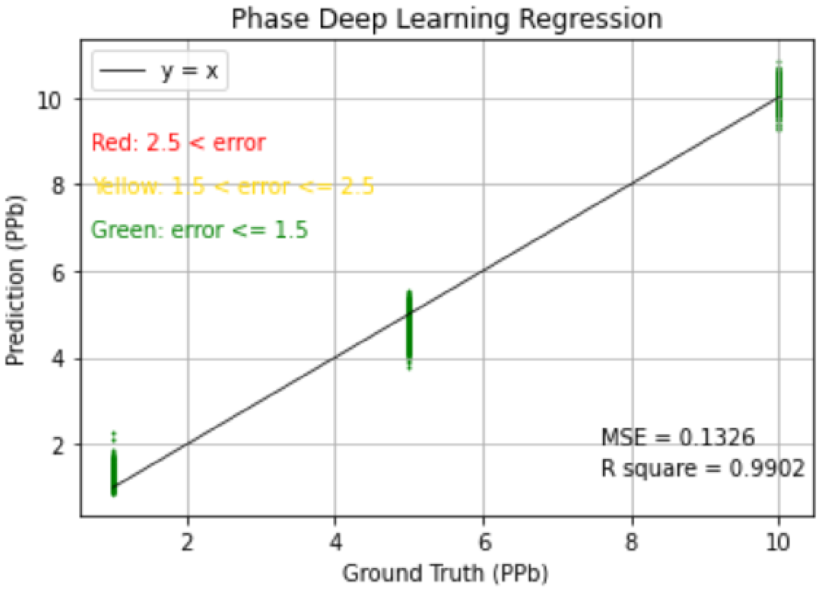


Fig. 3.8 Pb phase prediction and comparison

As can be seen in the picture, the value of the dependent variable Y, which is the ppb level, is represented by both the X-axis and the Y-axis. The actual value is shown by the X-axis, while the anticipated value is represented by the Y-axis.

The error scale is shown in the top left-hand corner of the illustration. Successive predictions are those in which the difference between their anticipated value and their actual value is less than or equal to 1.5. As long as the error stays between 1.5 and 2.5, the forecast will out to be confusing and not very accurate. Unless the gap between the anticipated value and the actual value is more than 2.5, the prediction will be found to be incorrect. For example, if the real value is 5ppb and the error is 3, the projected value is 2, or 8. This will be regarded as 1ppb or 10ppb, depending on the circumstances. This would be a symptom of a mistake, and it would demonstrate that the model is not sufficiently precise.

In these experiments, the fact that all of these predictions are accurate in the presence of metal concentrations may be observed. Most notably, in the prediction of 1ppB for Cu metal ions under phase conditions, some predicted values reach 3, implying that the results of several data points will be treated as 5ppb and that there will be some ambiguities. However, it is a minuscule figure that will not affect the soundness of the model.

3.2 Contribution of Cu to Network in Different Frequencies (Magnitude)

By using explainable AI algorithms, it can be seen that different attribution in different ppbs. In the 1200 testing data, there are 400 1ppbs, 400 5ppbs, and 400 10ppbs. According to each different frequency, a relevance score can be generated after going through the previous deep learning neural network model. Here are the results of the experiments:

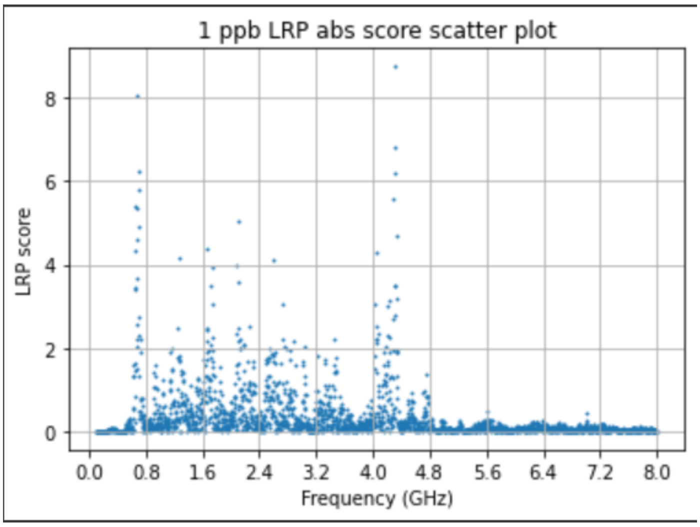


Fig. 3.9 LRP scatter plot of concentration 1 ppb in Cu_magnitude conditions

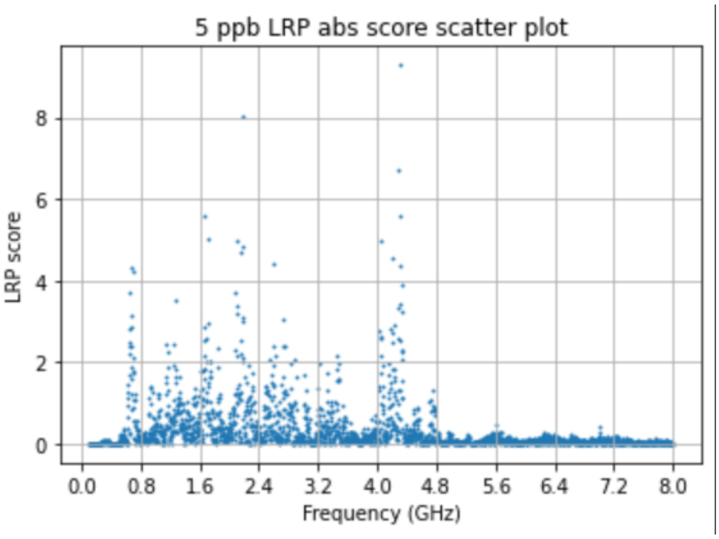


Fig. 3.10 LRP scatter plot of concentration 5 ppb in Cu_magnitude conditions

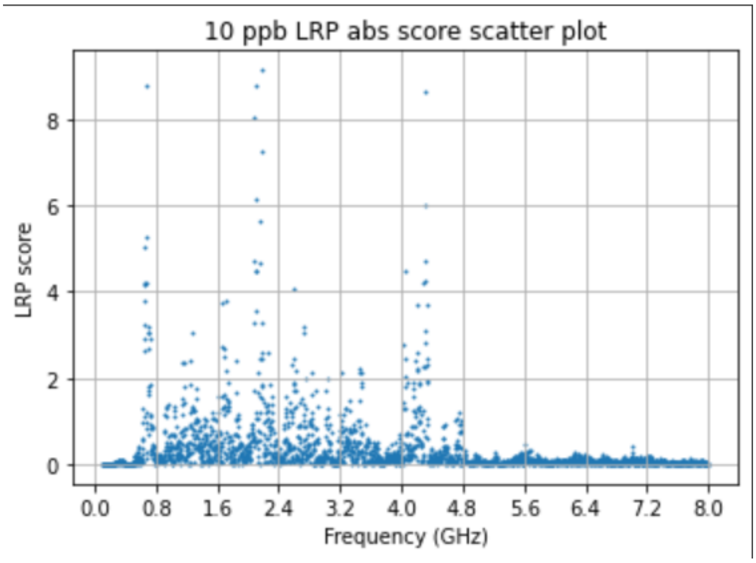


Fig. 3.11 LRP scatter plot of concentration 10 ppb in Cu_magnitude conditions

From the plots, it is easy to see that there are some high LRP scores at 0.7GHz, 2.3GHz, and 4.3GHz. After 6GHz, the data 6GHz didn’t contribute too much.

3.3 Contribution of Cu to Network in Different Frequencies (Phase)

Here are the plots in phase for Cu metal ions:

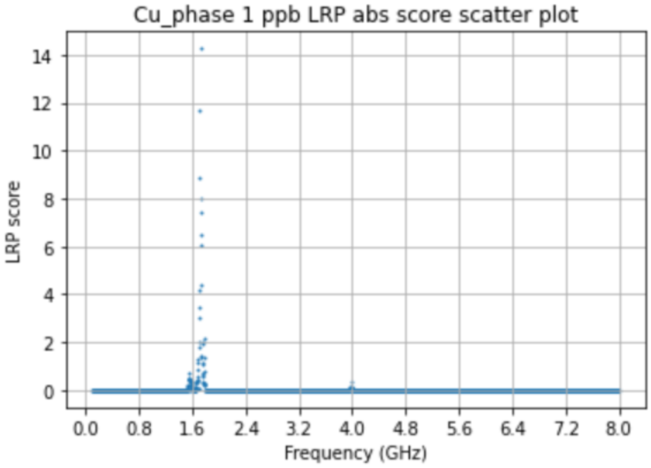


Fig. 3.12 LRP scatter plot of concentration 1 ppb in Cu_phase conditions

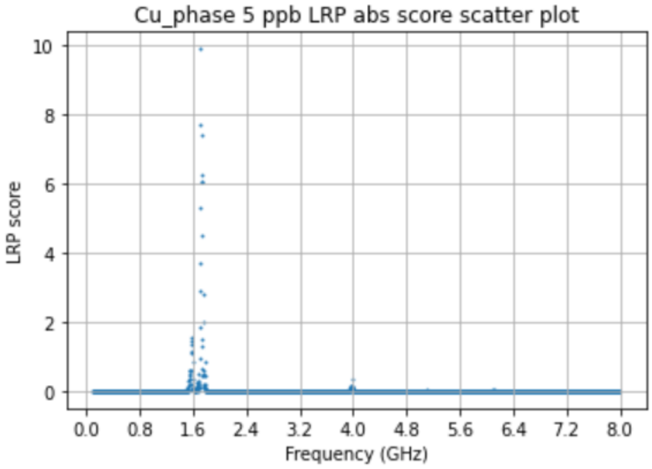


Fig. 3.13 LRP scatter plot of concentration 5 ppb in Cu_phase conditions

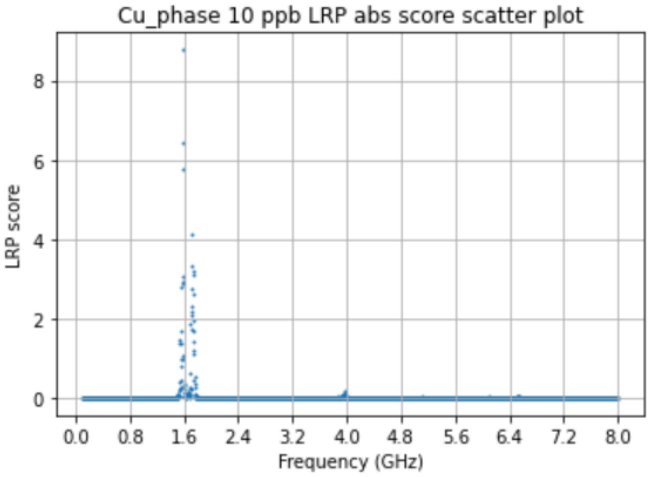


Fig. 3.14 LRP scatter plot of concentration 10 ppb in Cu_phase conditions

It is clear that there is just one high peak at 1.6GHz, with the maximum score coming in at 14, and that the remaining frequencies maintain extremely low scores, indicating that they contribute only a few or nothing.

3.4 Contribution of Pb to Network in Different Frequencies (Magnitude)

Here are the plots in magnitude for Pb metal ions:

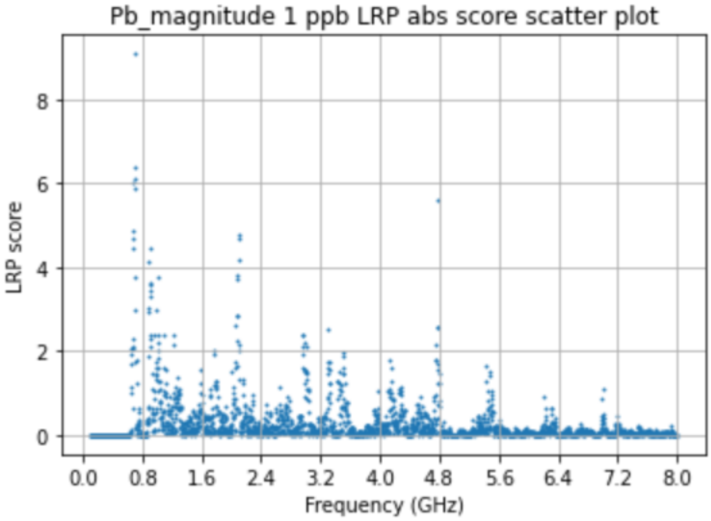


Fig. 3.15 LRP scatter plot of concentration 1 ppb in Pb_magnitude conditions

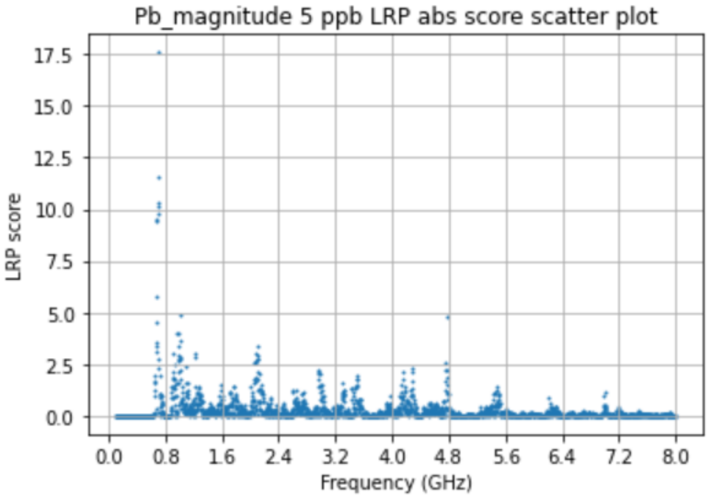


Fig. 3.16 LRP scatter plot of concentration 5 ppb in Pb_magnitude conditions

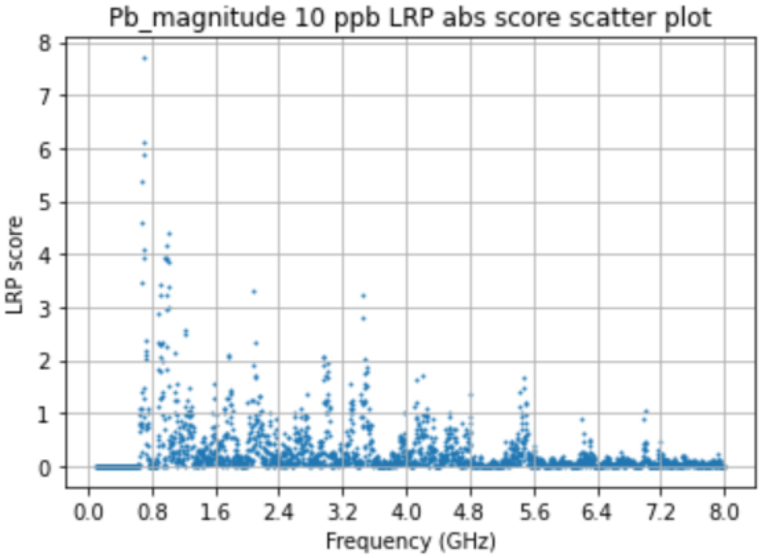


Fig. 3.17 LRP scatter plot of concentration 10 ppb in Pb_magnitude conditions

Generally, the LRP score distribution of Pb metal ions under the condition of magnitude is very ordinary, however, there are several peak values around 0.8GHz and 1.8GHz that are noteworthy. Overall, this scatter chart is not regular, and it is difficult to discover a vast variety of peaks or valleys in a very short period of time.

3.5 Contribution of Pb to Network in Different Frequencies (Phase)

Here are the plots in phase for Pb metal ions:

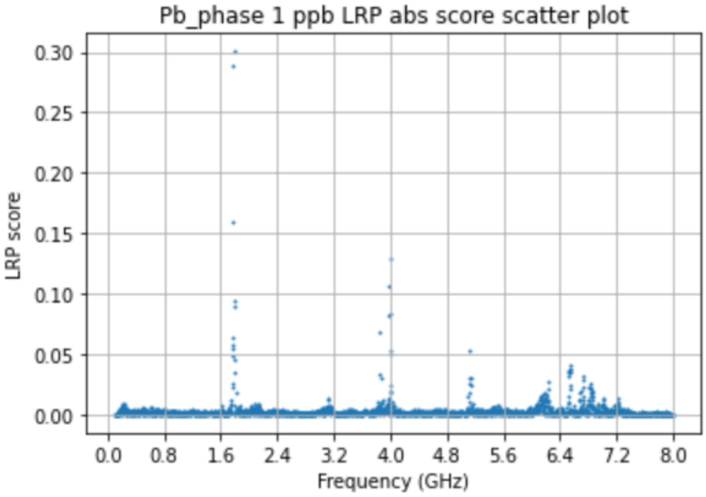


Fig. 3.18　LRP scatter plot of concentration 1 ppb in Pb_phase conditions

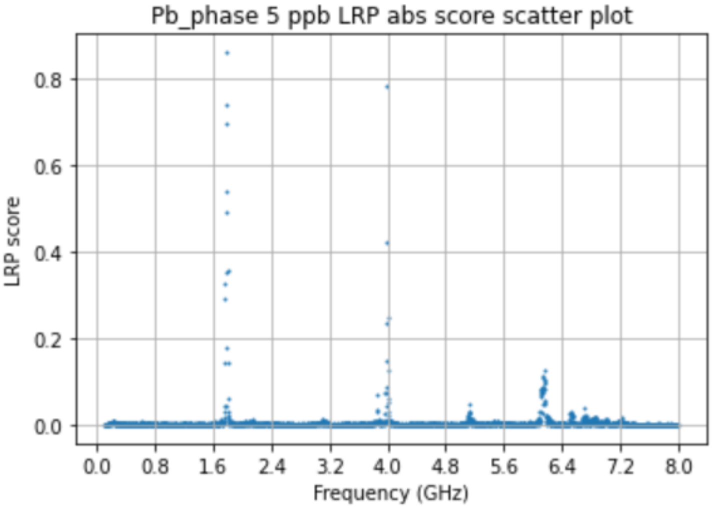


Fig. 3.19　LRP scatter plot of concentration 5 ppb in Pb_phase conditions

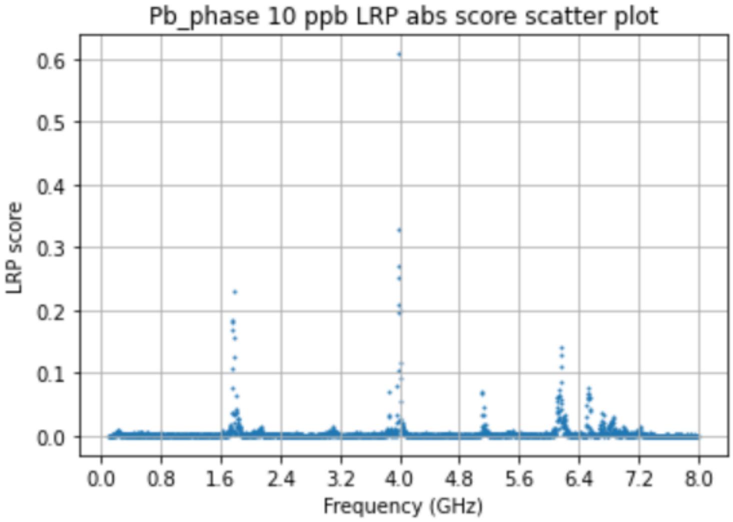


Fig. 3.20 LRP scatter plot of concentration 10 ppb in Pb_phase conditions

It can be seen that there are several peaks at around 1.7GHz and 4.0 GHz.

3.6 Comparison after Optimized Frequency Features

LRP is an interpretable artificial intelligence algorithm, according to the company. On the basis of the scatter diagram above, it can be determined that the majority of the peak is located below 6.0GHz. As a result, deep neural networks may be used to study the optimized frequency characteristics again, this time from 0.1ghz to 6GHz, in order to evaluate their performance. Maintaining a high-frequency range of high correlation scores while deleting the unneeded range may make the learning process more bearable while increasing the running duration and efficiency at the same time.

Following the selection of features, a number of advantages have shown themselves.

Overfitting is reduced since there is less duplicated data, which means there is less potential to make conclusions based on noise.

Increased Accuracy: Because there is less misleading data, the accuracy of the modeling is improved.

Reduces Training Time: Because there is less data, algorithms may be trained more quickly.

After shortening the frequency feature, the accuracy of the model remains good. The R square of each deep learning model is as follows:

Under 6GHz R square	Cu_magnitude	Cu_phase	Pb_magnitude	Pb_phase
Deep learning regression	0.996	0.992	0.516	0.939

Table 3.3 Optimized neural network R square

Here is the LRP under 6GHz frequency in Cu:

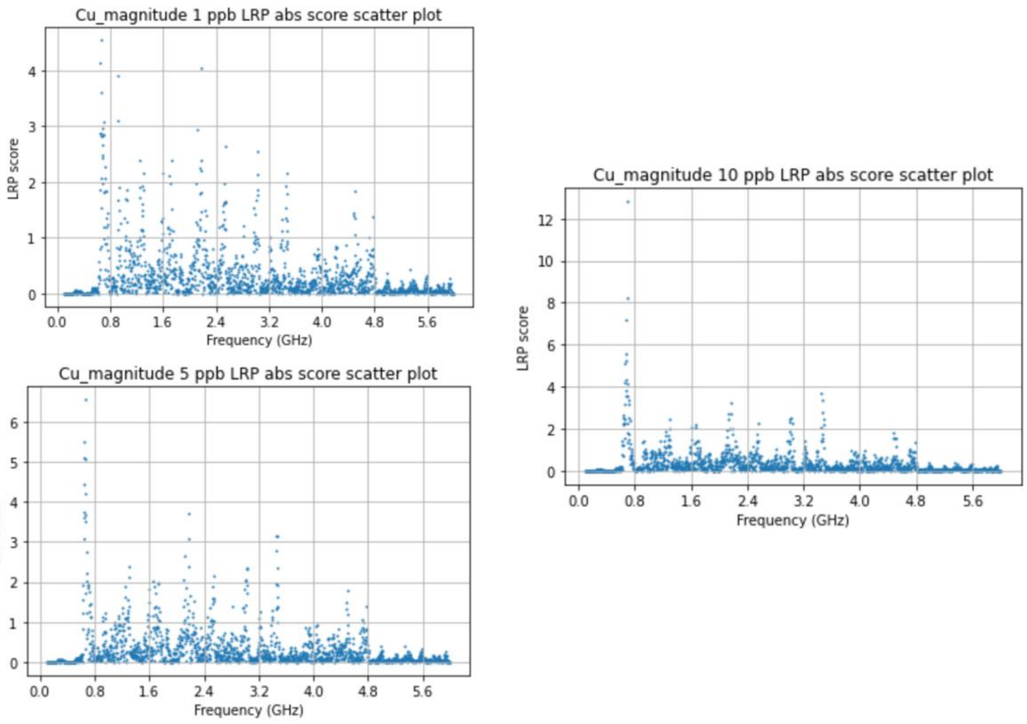


Fig. 3.21 Optimized LRP scatter plot for Cu in magnitude

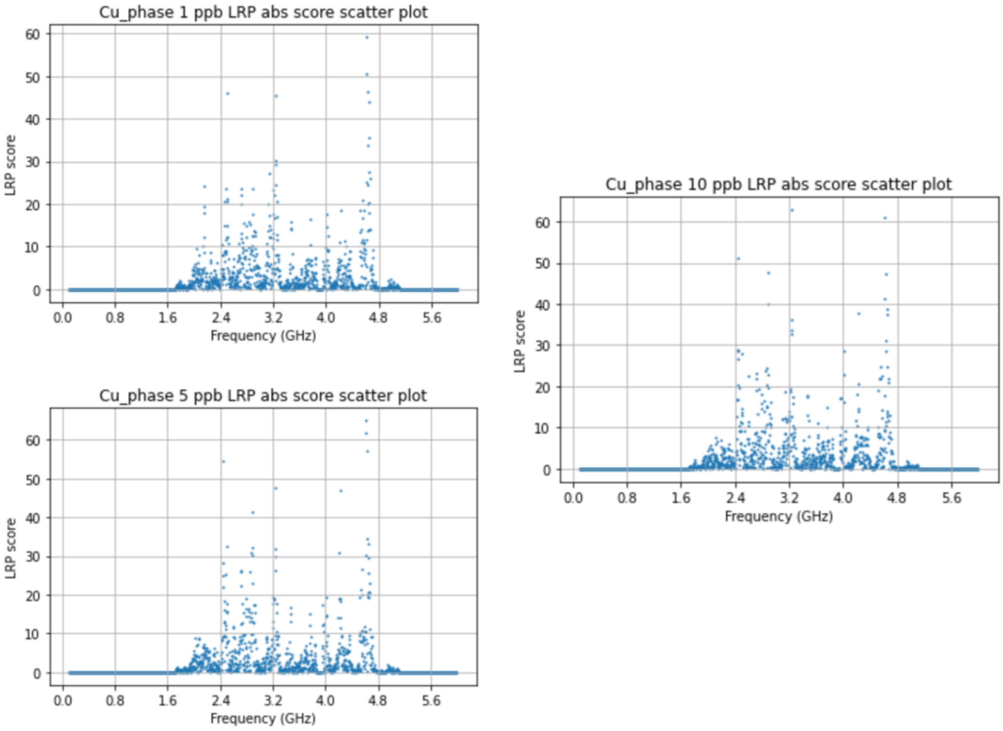
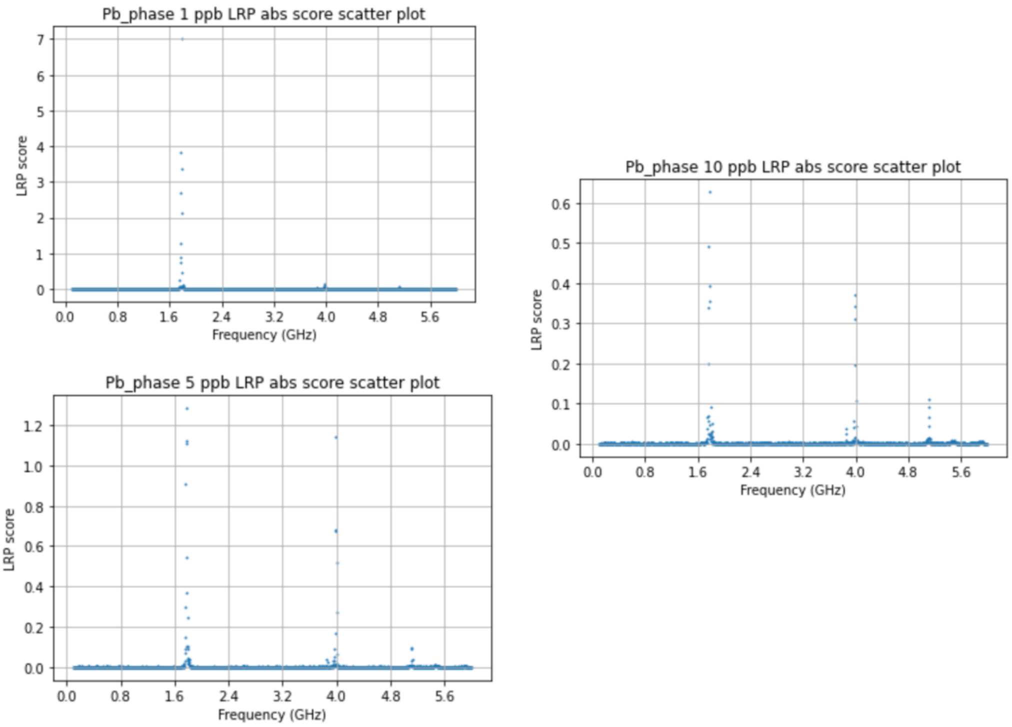


Fig. 3.22 Optimized LRP scatter plot for Cu in phase

The improved neural network for the Cu concentration learning model in water achieves a better increase in running time and accuracy owing to the decrease in the number of features while maintaining a high degree of capacity to predict outcomes.

Here is the LRP under 6GHz frequency in Pb:



Pb_phase 1 ppb LRP abs score scatter plot

Pb_phase 10 ppb LRP abs score scatter plot

Pb_phase 5 ppb LRP abs score scatter plot

Fig. 3.23 Optimized LRP scatter plot for Pb in phase

When subjected to the magnitude condition, the improved Pb learning model performs poorly. As can be observed in the above table, R Square is just 0.51; this indicates that the model does not perform effectively. The following are the primary reasons behind the speculative nature of the situation that is there are some little peaks from 6-8 GHz in the plot of Pb_magnitude LRP, which indicates these peaks matter and it seems non-reasonable to cut them off.

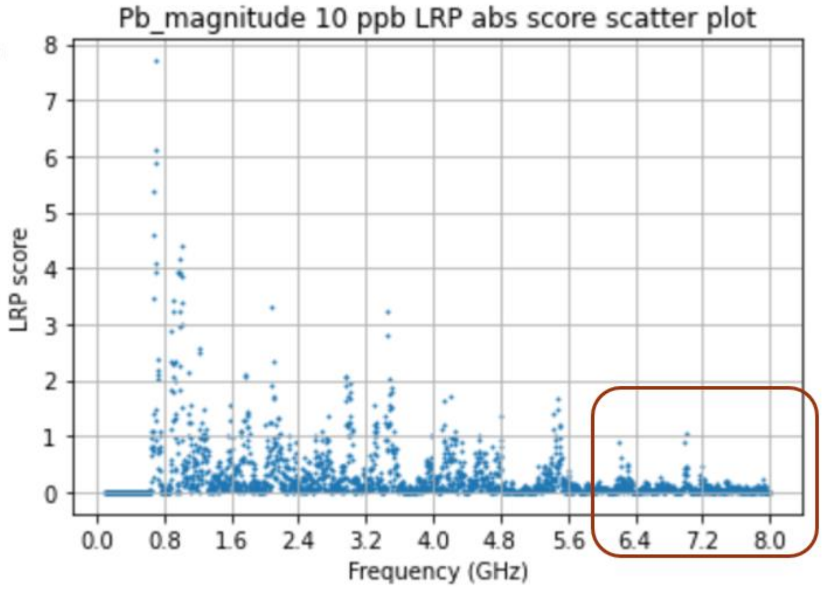


Fig. 3.24 Abnormal frequency range for Pb_magnitude

The Pb_phase R square under 6 GHz comes to 0.939, which is not so high as others because there are also some little peaks after 6GHz, but it does not matter to the model, the model still can perform well.

On the other hand, the Pb_magnitude shows an abnormal frequency range between 6-8GHz, it is important to train the model through these frequencies. There are some nonnegligible peaks during the last part of whole frequencies and that is the main reason why could not get a good model at the end.

3.7 Introduction of Advantages of Feature Selection

Some traits may be beneficial for a learning assignment while others may not be useful depending on the collection of attributes. The characteristic, in this case, is referred to as a "feature." A relevant feature is a characteristic that is relevant to the present learning activity at hand. An unimportant attribute is also a significant characteristic. When picking appropriate feature subsets from a given feature set, the procedure known as "feature selection" is used [28].

The following benefits may be gained from the selection of frequency characteristics in this experiment:

1. Simplify the model to make it more understandable. It will be easier to learn new things if you remove the elements that aren't important. Furthermore, the explainable performance provides greater certainty about the stability of the model effect.

2. improved performance: reduces the amount of storage and computation overhead

3. Increase generality while decreasing over-fitting danger: minimize the probability of dimension catastrophe. The addition of features will dramatically expand the search area available to models, and the number of training samples needed by most models will rise significantly as the number of features is added. Although the inclusion of features may help to improve the fit of the training data, it can also raise the variance of the results.

For the hardware area, VNAs (vector network analyzers) have long been associated with the RF/microwave business, since they are the sole instrument utilized in the high-frequency industry, and only by those with very large test-and-measurement budgets. However, in recent years, more engineers have discovered the utility of VNA measurements, and more consumers have discovered VNAs that are both inexpensive and perform well [29] [30].

Due to the substantial electrical and mechanical engineering necessary to develop and manufacture a dependable instrument with consistent performance, specifying a microwave VNA is used to inevitably guarantee a six-figure price tag. A VNA is an all-in-one measurement system that includes test signal sources, receivers, high-speed switches, and low-loss transmission lines, all controlled by a high-end embedded computer. To better understand the impacts of a DUT's impedance on the rest of a system, signals are switched between the ports of a DUT to measure direct and reflected power levels in both forward and reverse directions.

Even when comparing various benchtop equipment, a reduced price tag does not always imply a significant reduction in performance. When lowering VNA costs, the frequency range is the most apparent tradeoff. In this experiment, if 0.1 - 8GHz was used will almost cost double the price of 0.1 - 6GHz. Most benchtop VNA manufacturers sell their instruments as a set, with a single low-frequency limit and an increasing number of higher frequency limits as the price goes up. Therefore, reducing the frequency range would greatly reduce the cost and improve the efficiency of the experiment.

4. CONCLUSION

4.1 Summary

In this experiment, using an interpretable artificial intelligence algorithm, we learn how the weight of each feature varies depending on the different features used in the process of machine learning, allowing machine learning to produce the desired prediction results while also explaining the black box theory. An LRP algorithm is used to determine the contribution degree of various metals at different frequencies and in different conditions at the same time. This is done in order to accomplish the objective of cost reduction and efficiency improvement. It was possible to overcome the obstacles of this experiment by using standardization, MSE, R Square, and other data pretreatment techniques, as well as knowledge points utilized in model assessment, and to meet all of the goals.

4.2 Future Work

Since the data structure contains the magnitude and phase, future work can combine the magnitude and phase together to conduct a two-dimensional convolutional neural network. For each different metal ion, the dimensions of the two-dimensional data set will increase to (9000,2500,2). This has the advantage of consolidating data, avoiding too much fragmentation, and increasing the stability of the data set.

In addition, water quality from different dates can be added to merge and retrain the model, making the dataset more stable. After certain theoretical progress, we can try to sample water in other cities and use the same model to solve the detection and prediction of metal ion concentration in drinking water in different cities.

5. REFERENCE

[1] Oh S, Hossen I, Luglio J, et al. On-Site/In Situ Continuous Detecting ppb-Level Metal Ions in Drinking Water Using Block Loop-Gap Resonators and Machine Learning[J]. IEEE Transactions on Instrumentation and Measurement, 2021, 70: 1-9.

[2] Q.-Z. Hu and C.-H. Jang, "Liquid crystal-based sensors for the detection of heavy metals using surface-immobilized urease," Colloids Surf. B, Biointerfaces, vol. 88, pp. 622–626, Dec. 2011.

[3] K. M. Dean, Y. Qin, and A. E. Palmer, "Visualizing metal ions in cells: An overview of analytical techniques, approaches, and probes," Biochim. Biophys. Acta (BBA)-Mol. Cell Res., vol. 1823, no. 9, pp. 1406–1415, Sep. 2012.

[4] C.-S. Wu, M. K. K. Oo, and X. Fan, "Highly sensitive multiplexed heavy metal detection using quantum-dot-labeled DNAzymes," ACS Nano, vol. 4, no. 10, pp. 5897–5904, Oct. 2010.

[5] M. Mehdizadeh, T. K. Ishii, J. S. Hyde, and W. Froncisz, "Loop-gap resonator: A lumped mode microwave resonant structure," IEEE Trans. Microw. Theory Techn., vol. MTT-31, no. 12, pp. 1059–1064, Dec. 1983.

[6] G. A. Rinard and G. R. Eaton, "Loop-gap resonators," in Biological Magnetic Resonance Biomedical EPR Part B: Methodology Instrumentation and Dynamics, vol. 24. New York, NY, USA: Springer, 2005.

[7] W. Froncisz and J. S. Hyde, "The loop-gap resonator: A new microwave lumped circuit ESR sample structure," J. Magn. Reson., vol. 47, no. 3, pp. 515–521, May 1982.

[8] D. D. Thomas, C. H. Wendt, W. Francisz, and J. S. Hyde, "Saturation transfer EPR spectroscopy on spin-labeled muscle fibers using a loopgap resonator," Biophys. J., vol. 43, no. 1, pp. 131–135, Jul. 1983.

[9] R. L. Wood, W. Froncisz, and J. S. Hyde, "The loop-gap resonator. II. Controlled return flux three-loop, two-gap microwave resonators for ENDOR and ESR spectroscopy," J. Magn. Reson., vol. 58, no. 2, pp. 243–253, Jun. 1984.

[10] M. N. Gaboriaud, M. Desaintfuscien, and F. G. Major, "Absolute measurement of the total number of ions stored in an RF quadrupole trap," Int. J. Mass Spectrometry Ion Phys., vol. 41, nos. 1–2, pp. 109–123, Dec. 1981.

[11] A. Rahafrooz and S. Pourkamali, “Detection of sub-ppm traces of aqueous heavy-metal ions using micro-electro-mechanical beam resonators,” J. Micromech. Microeng., vol. 19, no. 11, Nov. 2009, Art. no. 115003.

[12] C. Ma and A. Wang, “Optical fiber tip acoustic resonator for hydrogen sensing,” Opt. Lett., vol. 35, no. 12, pp. 2043–2045, 2010.

[13] G. Gramse, M. Kasper, L. Fumagalli, G. Gomila, P. Hinterdorfer, and F. Kienberger, “Calibrated complex impedance and permittivity measurements with scanning microwave microscopy,” Nanotechnology, vol. 25, no. 14, Apr. 2014, Art. no. 145703.

[14] Kraus, Mathias & Feuerriegel, Stefan & Oztekin, Asil. (2019). Deep learning in business analytics and operations research: Models, applications and managerial implications. European Journal of Operational Research. 281. 10.1016/j.ejor.2019.09.018.

[15] Rinnan Å, Nørgaard L, van den Berg F, et al. Data pre-processing[J]. Infrared spectroscopy for food quality analysis and control, 2009: 29-50.

[16] Bucci O M, D'agostino F, Gennarelli C, et al. Data reduction in the NF-FF transformation technique with spherical scanning[J]. Journal of Electromagnetic Waves and Applications, 2001, 15(6): 755-775.

[17] Ba J L, Kiros J R, Hinton G E. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016.

[18] Wu Y, He K. Group normalization[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[19] Albawi S, Mohammed T A, Al-Zawi S. Understanding of a convolutional neural network[C]//2017 international conference on engineering and technology (ICET). Ieee, 2017: 1-6.

[20] O'Shea K, Nash R. An introduction to convolutional neural networks[J]. arXiv preprint arXiv:1511.08458, 2015.

[21] Vinayakumar R, Soman K P, Poornachandran P. Applying convolutional neural network for network intrusion detection[C]//2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2017: 1222-1228.

[22] Hasan M, Ullah S, Khan M J, et al. Comparative analysis of SVM, ANN and

CNN for classifying vegetation species using hyperspectral thermal infrared data[J]. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2019, 42: 1861-1868.

[23] Bach S, Binder A, Montavon G, et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation[J]. PloS one, 2015, 10(7): e0130140.

[24] Binder A, Bach S, Montavon G, et al. Layer-wise relevance propagation for deep neural network architectures[M]//Information science and applications (ICISA) 2016. Springer, Singapore, 2016: 913-922.

[25] Binder A, Montavon G, Lapuschkin S, et al. Layer-wise relevance propagation for neural networks with local renormalization layers[C]//International Conference on Artificial Neural Networks. Springer, Cham, 2016: 63-71.

[26] Böhle M, Eitel F, Weygandt M, et al. Layer-wise relevance propagation for explaining deep neural network decisions in MRI-based Alzheimer's disease classification[J]. Frontiers in aging neuroscience, 2019: 194.

[27] Alber M, Lapuschkin S, Seegerer P, et al. iNNvestigate neural networks![J]. J. Mach. Learn. Res., 2019, 20(93): 1-8.

[28] Kira K, Rendell L A. A practical approach to feature selection[M]//Machine learning proceedings 1992. Morgan Kaufmann, 1992: 249-256.

[29] Rumiantsev A, Ridler N. VNA calibration[J]. IEEE Microwave magazine, 2008, 9(3): 86-99.

[30] Wollensack M, Hoffmann J, Ruefenacht J, et al. VNA Tools II: S-parameter uncertainty calculation[C]//79th ARFTG Microwave Measurement Conference. IEEE, 2012: 1-5.