DESIGN AND LEARNING HYBRID RADIAL BASIS FUNCTION NETWORKS FOR HETEROGENEOUS DATA

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2022

Nouf A. Alghanmi

Department of Computer Science

Contents

Word Count: 33065

List of Tables

List of Figures

Nomenclature

$\{A_j^{V_m}\}_{j=1}^{p_{v_m}}$	The set of attributes describe the feature m
α, β	The maximum and the minimum of target value
β_{V_m}	coefficient matrix
ε	The threshold accuracy
$\mathbf{Y}, \mathbf{H}, \mathbf{W}$	output matrix, activation matrix and weight matrix
$\mathcal{H}$	The heterogeneous activation matrix
$\dagger$	A matrix that represent outputs from regression models
$\mathfrak{D}$	A heterogeneous dataset
ν	The RBF centers
σ	The standard deviation and the kernel width of RBF node
$D(.)$	Distance measure
D_{V_m}	Distance measure for feature V_m
$F_m(.)$	Regression function for the feature m
$h(.)$	The activation function in the hidden node

Ic_i	The input cluster
K	The number of splitting clusters
M	The total number of features
N	Total number of samples in dataset
Oc_i	The output cluster
P	The total number of nodes in the hidden layer of RBF network
p_{v_m}	The total number of attributes describing feature m
T	The total number of samples in an input cluster
V_m	The Feature m
w	The connection weight between hidden layer and output layer in RBF network
$y, \hat{y}$	True and predicted target value

Glossary

GloVe Global Vectors

BERT Bidirectional Encoder Representation from Transformer

RBF Radial Basis Function

OLS Orthogonal Least-Squares

TFIDF Term Frequency–Inverse Document Frequency

NLP Natural Language Processing

RF Random Forest

DNN Deep Neural Network

SMP Social Media Prediction

PSO Particle Swarm Optimization

KNN K-Nearest Neighbour

FCM Fuzzy *C*-means

CART Classification and Regression Trees

MSE Mean Squared Error

MAE Mean Absolute Error

SR Spearman Ranking

HRBF Heterogeneous Radial Basis Function

SSE Sum of Square Error

LR Linear Regression

DT Decision Tree

SVR Support Vector Regression

FRIOC Forward Recursive Input-Output Clustering

HDM Heterogeneous Distance Measure

Abstract

DESIGN AND LEARNING HYBRID RADIAL BASIS FUNCTION NETWORKS FOR HETEROGENEOUS DATA
Nouf A. Alghanmi
A thesis submitted to The University of Manchester
for the degree of Doctor of Philosophy, 2022

A heterogeneous dataset can be defined as a dataset having diverse types of features that describe a given instance or object. Each of these features represents a piece of valuable information. Currently, regression models limit the number of features that can be processed at one time, which means that only a subset of the information is considered. Consequently, their regression analysis deals with incomplete data descriptions, which are affected by significant information loss and miss important relationships between features.

With the rapidly increasing use of datasets containing mixed data types, current learning techniques for this kind of dataset include pre-processing and learning phases. The former focuses on unifying data types by transferring them into categorical or numerical inputs or defining distance measures. The resulting data can be used in learning models suited to its types. However, this scheme approach to dealing with mixed data types can lead to a lack of compatibility. It may also suffer from tremendous

data dimensions, which may overload the computation capacity of the learning model.

This study focuses on developing a regression model that can handle heterogeneous datasets based on a radial basis function. Three main solutions are proposed. The first solution is based on defining a heterogeneous distance measurement and then using it to train a radial basis function network. The second solution is developing a regression model without the need to define a distance measure or unifying data types by the rough development of a heterogeneous radial basis function regression model that can directly learn from heterogeneous data. As each feature has its own characteristics and has been widely explored in the literature, a hybrid-regression model is proposed by combining multiple regression models. With this strategy, information can be extracted efficiently, and underlying knowledge is revealed optimally by developing a model for each data type.

These three proposed models as a solution to the regression analysis for heterogeneous dataset, were evaluated using a set of mixed numerical and categorical datasets and social media prediction data that contained numerical categorical and textual features. The results of these models were compared to well-known regression models, such as random forest, support vector regression, and linear regression. The best results were achieved from the hybrid-regression, where the learner's performance was significantly increased. This model proved effective, and its results showed that with suitable models and simple approaches, heterogeneous data learning problems can be solved quite easily.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

When it was dark, you gave me light,

when I was hopeless, you gave me hope,

when I was weak, you gave me strength,

when I was upset, you cheered me,

when I was stressed, you gave me peace.

To one who feels I am his princess,

To the one who means everything to me and fulfils every relation in the world, you have been the father, the friend, the brother, the husband, the lover...

To you, Fareed, I love you.

Acknowledgements

It is Allah who deserves all thanks and praise for guiding me through this achievement. Thank you, my Lord, for giving me the strength to overcome my hardest moments. Thank you, my God, for helping me achieve my goal.

My supervisor, Prof. Xiao-Jun Zeng, provided me with invaluable guidance, support, and encouraging remarks, which I gratefully acknowledge. His vision played a critical role in shaping my research, and I am truly thankful for this.

It is my great pleasure to extend my deepest thanks to my PhD sponsor, King Abdul-Aziz University, the Ministry of Education and the Saudi Cultural Bureau in London, for their continued help and support during my research in the United Kingdom.

A grateful thanks to my family for their support by taking care of my daughters during my studies. A special thanks to my mother, Mrs. Alyiah Alghanmi, the strongest woman I have ever seen. You sow in me the strength to fight and challenge the odds in order to succeed. Thanks for your endless support by raising my girls for two years. It is hardly possible to thank you enough. A special thanks to my father, Mr. Atihaallah Alghanmi, for his loving support—knowing you were there praying for me was more than enough. I really love you, Dad.

I would like to give a special thanks to the sweetest little ladies that I have in my life, my sisters Nada, Samaher, and Rafad; you were part of this achievement because you were able to cheer me up, splitting my responsibilities as a mother with me. Thanks to my gentlemen, my brothers, Nawaf, Ahmed, Mohammed, and Abdulrahman for their support and care. Thank you for being there for me.

Also, I would like to thank my best friends Jawaher, Ibtisam, Nada, and Sajwa for all the lovely nights we spent together. You were like my sisters in Manchester. Those times will not be forgotten.

Finally, for my soulmate, my husband, Fareed, I can never express my gratitude to you in words. You encouraged me, stood by me in my weaknesses, and gave me the strength to complete my Ph.D. You are my mentor, the shoulder I lean on when I'm weak, and my best friend. I cannot wait to get back to you again because you are my favourite person, and I cannot wait to rest in your arms. Special thanks to my lovely princesses, Ladan, Hamss, and Muhra; you are my butterflies who bring colour and joy to my life. I love you all.

COVID-19 Impact Statement

At the beginning of the pandemic, I had to evacuate my country and return to Saudi Arabia. I spent the first week in Riyadh isolated from the outside world, and then I travelled to Jeddah for 10 hours in my car. I had no residence, so my children and I had to live with my parents. My parents moved to Dammam 1200 km from Jeddah, so I had to move there with them. I lived with my mother in Dammam for one year, during which I had to help her care for my paralyzed father. On top of that, I also had to look after my girls, including special attention that was required by my the little one, who was one year old. For my elder daughters, I had to ensure that they continued to receive their education through distance learning and reduce their fears.

Being a good daughter, a responsible mother, and a caring wife caused me to be overwhelmed; the more responsibility I had, the more time I had to devote to them and the more time I had to spend on chores. In addition to managing my time and dealing with numerous distractions, I had to make efforts to manage my mental and physical health during the COVID-19 pandemic by managing stress and focusing on the things I could control.

Chapter 1

Introduction

1.1 Context and Motivation

Advancements in communication and computing technology continuously generate vast amounts of information. This has triggered an information explosion, and the situation is increasingly complex and challenging to manage as the Internet of Things spreads. In many fields, the term Big Data is now more popular even on non-technical environment [105]. Big data describes three main characteristics, commonly referred to as the three Vs: volume, velocity, and variety. These aspects of big data are beyond the capabilities of conventional systems to handle. Volume refers to the massive amount of data produced; velocity is the increasing data flow rate; and variety defines the nature of collected and managed data.

The research presented in this thesis attempts to analyse the variety aspect of big data. Variety is interpreted as the diversity of data types associated with the magnitudes and variables that describe data. With these data, board-wide domains of problems, situations, and processes are captured and described. Additionally, these data can be imprecise and incomplete, introducing another heterogeneity factor. A growing need for learning from heterogeneous information is prevalent in a wide range of domains, since these information sets include variables that may be scalar, vector, or of a more complex structure with values of different types (numerical, non-numerical, images, signals and videos) [105].

Data from Internet of Things devices or clinical records are typical examples of information comprised of heterogeneous data types. The clinical record of a patient can contain qualitative scalar variables, such as age or gender; quantitative scalar variables such as temperature and pressure; imprecise magnitudes, such as level of pain and radiographic images; signal magnitudes such as an EKG or ECG; and numerical scalar variables such as blood pressure, and documents (for example laboratory reports and blood tests) that may be interpreted as time series data. The objective of this study is to deal explicitly with the data heterogeneity (i.e. different types of features describing same objects). It mainly deals with the complexity of multiple data types describing data/information.

Although these variables describe the same object (patient), they have different types of data, and each one represents a piece of valuable and partial information. Several regression models, including linear regression, support vector regression, and neural networks, are well-defined and validated to analyse numerical data efficiently [53, 64]. These models limit the number of variables/features being processed at a time, which means that only a subset of the information is considered. Consequently, the analysis will deal with incomplete data description, meaning it will be affected by significant information loss and missing important relationships between variables. Furthermore, a complete analysis of such data will demonstrate information that otherwise would not be disclosed if only a subset of information is considered during the analysis. As the current regression models do not currently address this area of development, there is a gap to be filled.

There are two types of machine learning algorithms: supervised and unsupervised.

A key difference between supervised and unsupervised learning is the presence of target values. Supervised machine learning models can be divided into classification and regression models that classify or predict response values based on their inputs. The former predicts qualitative information while the latter predicts quantitative information [5]. In the context of big data, as heterogeneous data with different variables are becoming increasingly prevalent, learning from such data is receiving greater attention. Researchers have considered this issue and proposed some solutions to address the problem. The first solution is a pre-processing step, where the data are unified so they can be trained by machine learning models that accept that type of data. One of the most well-known unifying techniques is the encoding/embedding/transforming method, which gives categorical features a numerical representation. As this method is simple, fast, and does not lose underlying information or data, it suffers from tremendous data dimensions when the categorical cardinality is high, which overloads the computation capacity of the learning model [62, 63]. Discretisation is another well-known unifying technique, which interprets numerical variables as categorical variables by splitting the numerical value range into sub-ranges, and each numerical value is consistently associated with an interval [120]. Discretisation can be considered a data reduction technique since it reduces the indefinite domain of numerical attributes to a restricted set of categorical values [86]. This thesis addresses the challenge of learning from heterogeneous datasets, particularly in regression learning fields, by developing regression models that can train and learn efficiently from heterogeneous datasets.

1.2 Research Aim and Objectives

With regard to the heterogeneous data regression issue, the overall aims are: (i) to examine the current approaches to dealing with heterogeneous datasets and suggest simple, efficient approaches; and (ii) to design and develop regression models that learn from heterogeneous datasets. Radial Basis Function (RBF) network is used throughout this study as the basic model. Radial Basis Function (RBF) network is simpler and easier to design than neural networks due to their simple and fixed three-layer architecture. From a generalisation standpoint, RBF networks can perform well with patterns that are not trained. The robustness of RBF networks to input noise enhances the stability of the designed systems [116].

In more detail, the research objectives of this thesis are as follows:

1. Define a heterogeneous distance measurement to compute the distance between two heterogeneous samples. Then use this measurement to train a Radial Basis Function RBF model to learn from heterogeneous datasets.

2. Propose structure learning for Radial Basis Function to obtain the optimal number and locations of RBF kernels.

3. Develop a Radial Basis Function regression model that will learn directly from heterogeneous data, thus eliminating the need to unify all data types or implement heterogeneous distance measurement.

4. Develop a simple hybrid regression model for heterogeneous datasets based on the combination of multi-regression models. The hybrid regression model should consider data heterogeneity and construct a proper regression model for each distinct data type before combining their outcomes.

5. Verification and comparison of models should be performed. These proposed models should be verified with a variety of heterogeneous datasets. The strengths

and weaknesses of the proposed approaches should be discussed and compared with existing state-of-the-art approaches.

1.3 Thesis Contributions

- A heterogeneous distance measure based on an attribute-weighted scheme is proposed for measuring the distance between heterogeneous objects. It should be possible to calculate the distance between two heterogeneous samples based on this measurement.

- The RBF kernel is computed based on distances between its input samples and its kernel centres. A heterogeneous distance measure should be defined to train RBF on heterogeneous datasets. Moreover, a supervised clustering technique referred to as Forward Recursive Input-Output Clustering (FRIOC) is used to determine the optimal location and number of kernels for the RBF network.

- To learn from heterogeneous data directly, a more robust Heterogeneous Radial Basis Function (HRBF) model is proposed that does not require the definition of distance measurement based on constructing an RBF network with heterogeneous nodes at its hidden layer.

- A combination of multi-regression models, referred to as a hybrid regression model for heterogeneous datasets, has been constructed. A simple hybrid mechanism is proposed in contrast to the current trend of more complicated models, leading to the impression that complicated models are necessary for complicated heterogeneous data. A model is proposed based on a very simple combination mechanism of multi-regression models. It starts by choosing a regression model that suited for each type of data. The final prediction output is then derived from the combined results of these regression models.

- Multiple state-of-the-art approaches have been compared with the proposed approaches for heterogeneous data regression on a set of heterogeneous data with two and three types of data.

- Discuss the weakness and the strength of the proposed approaches and compare them with the most common regression models such as Support Vector Regression (SVR), Random Forest (Random Forest (RF)), Linear Regression (Linear Regression (LR)), and published results from competing models.

1.4 Thesis Structure

The rest of this thesis is organized as follows:

Chapter 2. The related work used throughout this thesis is introduced in this chapter. The definition and the challenges of the heterogeneous dataset are first introduced. A literature review of the regression approaches used to handle heterogeneous datasets is then presented. Following this, the most famous model for training heterogeneous data is described. Finally, the preparation steps for the heterogeneous datasets used throughout this thesis are described.

Chapter 3. The focus of this chapter is on developing a Radial Basis Function network (RBF) with a special initialization approach to train heterogeneous datasets. Furthermore, a distance measure for heterogeneous datasets is defined and applied to the computation of RBF kernels.

Chapter 4. This chapter develops a heterogeneous RBF network that consists of a set of heterogeneous nodes in its hidden layer. Models consist of two main phases: structure learning stage and parameter learning stage. In the first phase, the optimal number of centers for each data type is determined. A heterogeneous

RBF network is then constructed before the learning process begins. Finally, optimum connection weights are computed between the hidden and output layers by applying the Least Square method.

Chapter 5. This chapter develops a simple multi-regression model for heterogeneous datasets, namely a hybrid regression model. It combines multiple regression models to produce the final model outcome. It started by training each feature with a well-formed learning model, followed by linearly combining their results to create the final prediction.

Chapter 6. In this chapter, the conclusions of the study and suggestions for future work are presented.

Chapter 2

Background and Related Work

This chapter summarises the background and related work that is of relevance throughout this thesis. Firstly, a literature review describing the common supervised machine learning approaches that used for training mixed numerical and categorical data types is provided in Section 2.1. The following sections will describe the most famous regression models that deal with mixed data types, and these models will be used for comparison with the proposed models. These models are: Support Vector Regression in Section 2.2, Decision Tree in Section 2.3, Random Forest in Section 2.4, and Radial Basis Function in Section 2.5. Finally, this chapter is briefly summarized in Section 2.6.

2.1 Supervised Learning on Mixed Data Type

There has been a rapid rise in the incidence of datasets containing mixed data types. Current learning techniques for this kind of dataset include pre-processing and learning phases. The former focus either on unifying data types by transferring them into categorical or numerical inputs, or defining distance measures. The resulting data can be used in learning models that are suited to its types. This scheme approach to dealing with mixed data types can lead to a lack of compatibility [64]. Thus, designing a supervised machine learning model that directly learns from mixed data (heterogeneous data) is a challenging task. Recently, this problem has been tackled by [64] who proposed a classification RBF-ELM framework to manage mixed numerical and categorical data types directly. They define a distance measure for mixed data types, in the form of a weighted sum of categorical and numerical distances. This distance measure was then used to compute distances in RBF-ELM kernels. Additionally, [53] proposed a hybrid regression tree model for mixed data types (numerical and categorical). These train the categorical feature with a decision tree to predict y_tree. This value represents the contribution of categorical features in the final predicted values. After which, this value and the numerical variables can be used to train regression models and predict the final output. They examined five regression types, i.e., linear, ridge and lasso regression, k-NN regression, and SVR model, to estimate the target values. Their model will be used as the basis from which to evaluate our proposed model. Moreover, [4] developed sampling filtering techniques for a classification model based on similarity measures. The similarity space utilised Minkowski distance for numeric features, and simple matching for categorical features. However, their method has been tested on datasets comprising moderate amounts of data, but has not been used to evaluate a dataset containing a significant number of samples.

2.1.1 Dealing with Categorical Data in Supervised Learning

In the literature, the term mixed data type refers to a dataset containing both numerical and categorical data. State of the art machine learning algorithms are suited to numerical data or digital data representation. In contrast, categorical features are represented by characters or words that contain a semantic meaning within them, and as such cannot feed directly into these models [29, 45, 3]. [3] mentions two chief characteristics of categorical features: explicit semantic meaning and disabilities of inherent mathematical operations. Notably, mathematical operations cannot be directly applied to them. Researchers have considered this issue, and proposed some solutions to address the problem. There are two common approaches to giving the categorical feature a numerical representation so that it can be trained by machine learning models that accept numerical values. The first technique is to transform categorical features into numerical representations via encoding/embedding/transforming methods. The second is to define a distance metric for them. A brief description of the work done to them is detailed below:

2.1.1.1 Transforming Methods: Categorical to Numerical

One of the widely used encoding methods is one-hot encoding, which uses binary coding to represent categorical features on a zero-one matrix [12, 74]. In One-hot encoding, features are transformed to dimensional vectors, where each vector has a value of 1 corresponding to one category and the rest entries are 0s [45]. Although this method is simple, fast, and does not lose the underlying information or data, it suffers from tremendous data dimensions when the categorical cardinality is high, which overloads the computation capacity of the learning model [62, 63]. Similar to word embedding in natural language processing, [32] combined one-hot encoding with a neural network to develop an entity embedding method to capture the representation of categorical values automatically. However, it required extreme computation time to find the dimensions of the embedding layers. The bound of the dimension is between 1 and $m_i - 1$, where m_i is the number of categories in the categorical variable x_i. It requires several experiments to select the optimal value for each entity embedding dimension. Moreover, [12] proposed a similarity encoding scheme, which took the form of generalisation of one-hot encoding considering similarity across categorical values.

Recently, several efficient embedding methods have been used to define a similarity measurement and examine different combinations of categorical features values. For instance, a pairwise distance measure is defined in [119] to capture the relationship between all categorical values. While the work in [29] choose the most significant eigenvectors from a pairwise distance matrix to represent categorical features. The study in [38] proposed transferring the categorical feature by using Separability Split Value Transformation (SSVT) based on Separability Split Value (SSV). This method relied on the splitting criteria of the decision tree, then upon ordering leaf nodes to represent categorical features. Subsequently, this work was improved by [62] who applied a fine-tuning conditional probabilities method.

2.1.1.2 Discretisation Methods: Numerical to Categorical

An important aspect of discretisation is the interpretation of numerical variables as categorical ones, identifying a non-overlapping division of a continuous domain. Specifically, the numerical value range is split up into subranges, and each numerical value is consistently associated with an interval [120, 13, 86]. Furthermore, discretisation may also be thought of as a data reduction technique since it reduces an indefinite domain of numerical attributes to a restricted set of categorical values [86].

For many reasons, discretisation is an essential pre-processing mechanism used by data scientists. First, there is a demand for data discretisation since, in some cases, machine learning models can only train categorical variables, such as C4.5 and Naive Bayes [56]. Furthermore, discretisation can have a significant impact on learning speed and accuracy. In addition, decision trees utilising discretisation produce more accurate, shorter, and compact results, which can be closely examined, compared and reused [66].

In general, a typical discretisation process is composed of four steps, as illustrated in Fig. 2.1 [86].

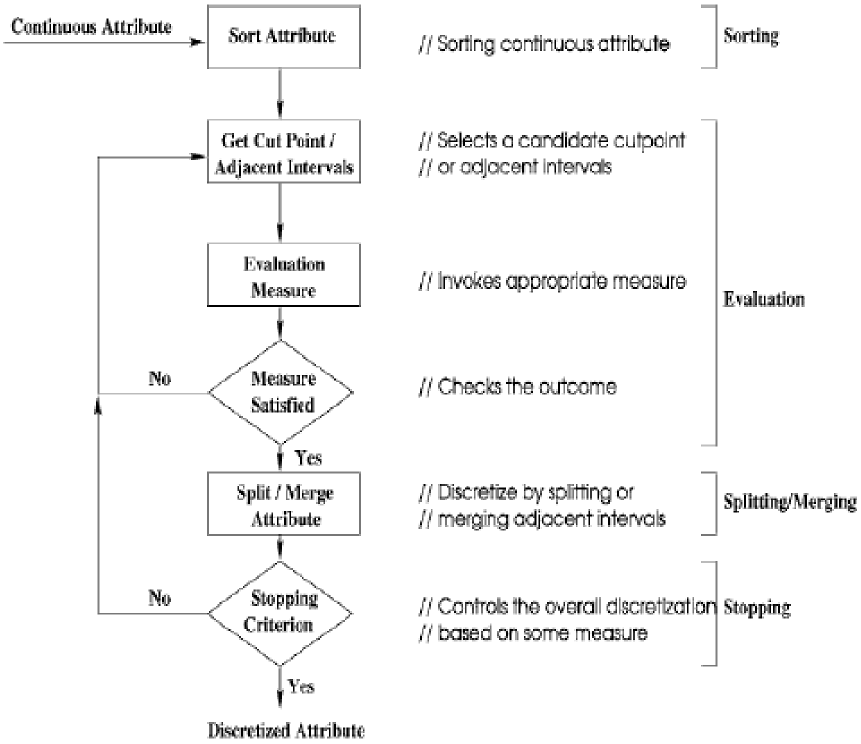

Figure 2.1: Discretisation process [66].

- Sorting. The value of the continuous attribute is sorted in ascending or descending order. Sorting is a computationally expensive process that uses an efficient sorting algorithm with a time complexity of O (NlogN). The most used sorting algorithm is ”QuickSort”. The sorting step should be performed only once to permit efficiency at the beginning of the discretisation process.

- Cut-point selection. After sorting, the next step is to find the best cut-point for the best pair of adjacent intervals to perform merging or splitting in the following steps. An evaluation measurement or function determines the best cut-point according to entropy, gain, performance-boosting, or any other benefit to the output value.

- Splitting or merging. In the top-down discretisation approach, splitting occurs, while in the bottom-down approach, interval-merging occurs. In splitting, the cut point divides the range of numerical attributes into two intervals, while in merging, it seeks to select the adjacent intervals to merge.

- Stopping Criteria. This specifies when the discretisation process should end. It should make a trade-off between accuracy and the final lower number of intervals.

Over a hundred discretisation models exist, and choosing the right one for a problem has a significant impact on performance. The goal is to obtain a nominal value for each numerical value. In the supervised discretisation, the obtained results should have a close association with the class label in the classification problem by using the information from output space to decide the discretisation intervals [112, 25]. Here, we will describe the three common discretisation approaches: Equal width, Equal frequency, and Minimum Descriptive Length (MDL).

- Equal width. This is the simple unsupervised discretisation model that computes the range of numerical attributes and divides it into intervals with equal length. This will result in unbalanced intervals, with different numbers of items within each interval, leading to certain intervals being more popular than others.

- Equal frequency. This model aims to generate intervals with a constant number of items.

- Minimum Descriptive Length (MDL). The model finds the best cut-point based on minimum information entropy.

Generally, the equal width and equal frequency approaches are simple and easy to implement; however, finding the optimal number of intervals can be difficult [16]. Some use random numbers, while others rely on training and error. In addition, these models ignore the useful information provided by the output space. Nonetheless, the classical discretisation methods failed to handle big data as they are not designed to deal with large amounts of data. The most common discretisation approaches have been modified by using distributed computation frameworks to resolve this issue. Computational costs should not exceed $O(n)$ to ensure scalability even when dealing with big data. MDL is the most common supervised discretisation method, which has been implemented by using the multi-thread process by [23], and by using Apache Spark [86, 85].

2.1.2 Distance Measure for Mixed Type

This section will focus on a study that developed distance measurement for mixed numerical and categorical data types. Distance measurement in machine learning involves a mathematical representation of distance functions representing how far away two points are. A distance metric should fulfil the following three primary axioms:

1. The identity of indiscernible elements; if the distance between two points is 0, these points are considered identical.

2. Symmetry; there is no explicit order when computing the distance between two points.

3. Triangle inequality; such that the distance sum for any two sides, has to be greater than or equal to any other side.

The most famous and straightforward measures for numerical data and categorical data are Euclidean distance and matching distance, respectively. Real datasets utilising mixed categorical and numerical data are ubiquitous in the real world. The Gower [30] similarity measure was the very first measurement introduced to compute the distance between categorical and numerical observations. Notice that similarity is another concept related to distance,$similarity = 1 - distance$.The Gower coefficient between two samples i and j is computed as in Eq.(2.1).

$$S_{ij} = \frac{\sum_{k=1}^{P} s_{ijk}\delta_{ijk}}{\sum_{k=1}^{P} \delta_{ijk}} \tag{2.1}$$

Where δ_{ijk} es a missing value indicator, which is when there are otherwise no null value or zeroes. The s_{ijk} e matching similarity for categorical features; meanwhile, for numerical features, the range normalised Manhattan distance is used. A modification to the Gower distance equation was proposed by [37] to include the variance of Gower distance, as in (2.2).

$$S_{ij} = \sqrt{\frac{\sum_{k=1}^{P} s_{ijk}\delta_{ijk}}{\sum_{k=1}^{P} \delta_{ijk}}} \tag{2.2}$$

Later on, [81] Gower distance was extended to include the ordinal variables by replacing another distance measure defined by Huang [42], combining the square Euclidean distance and matching distance for numerical and categorical data, respectively, as in (2.3).

In that equation, the d_{ij}^{N} is numerical distance, d_{ij}^{C} is the categorical distance, and γ is the user-defined weight-based on the distribution of numerical variables. Soon after, [1] defined the γ in the measure function itself as the co-occurrent distance between two categorical values. Moreover, a generalised distance measure for categorical, numerical and binary features was defined by [37]. They argue that binary variables have different probability distributions than categorical features. Consequently, their measurements were based on three distance metrics: Manhattan, Hamming and co-occurrence for numerical, binary, and categorical, respectively.

$$d_{ij} = d_{ij}^{N} + \gamma d_{ij}^{C} \tag{2.3}$$

and

$$d_{ij}^{N} = \sum_{k=1}^{P_n} \left(x_{ik} - x_{jk}\right)^2$$

$$d_{ij}^{C} = \sum_{k=1}^{P_c} \delta_C \left(x_{ik}; x_{jk}\right)$$

Furthermore, a heterogeneous distance measure that includes the overlap metric between categorical features was introduced by [111]. This distance can work only on classification tasks where the target value is set based on distinct classes. After a number of methods they conclude with a Heterogeneous Value Difference Metric (HVDM) defined as:

$$HVDM = \sqrt{\sum_{k=1}^{p} \left(d_{ij}\right)^2}$$

$$d_{ij} = \begin{cases} 1 & \text{if } x_i \text{ or } x_j \text{ is unknown, else} \\ normalised\_diff(x_i, x_j), & \text{for numerical features} \\ normalised\_vdm(x_i, x_j), & \text{for categorical features} \end{cases}$$

They found the results were achieved by defining the $normalised\_vdm$ and $normalised\_diff$ functions as:

$$normalised\_vdm(x_i, x_j) = \sqrt{\sum_{c=1}^{C} \left| \frac{N_{k,x_i,c}}{N_{k,x_i}} - \frac{N_{k,x_j,c}}{N_{k,x_j}} \right|^2}$$

$$normalised\_diff(x_i, x_j) = \frac{|x_{ik} - x_{jk}|}{4\sigma}$$

where $N_{k,x_i,c}$ is the number of samples that have a class label c, and a value x_i for attribute k.

Similarly, where $N_{k,x_j,c}$ is the number of samples that have a class label c, and a value x_j for attribute k. Meanwhile, N_{k,x_i} is the number of samples with a value x_i for attribute k. Similarly, N_{k,x_j} is the number of samples with a value x_j for attribute k.

2.2 Support Vector Regression

Support Vector Regression (SVR) is an extended model based on the Support Vector Machine (SVM) developed by Vapnik et al. [117]. The objective of SVM is to compute a hyperplane that correctly classifies input samples. A good hyperplane is found by maximising the margin between the data points nearest to the hyperplane. In regression tasks, the ε insensitive loss function is used to compute the hyperplane, allowing an ε deviation between the predicted and actual values. For the generalised bounds computation of regression, the hyperplane plus ε is used to form an ε-insensitive tube. The optimisation then aims to minimise the ε-insensitive to narrow the training samples as much as possible [117, 26].

For a dataset $\{x_i, y_i\}_{i=1}^{n}$, where x_i is the input sample and y_i is the target value The linear ε-SVR can be formed as follows.

$$y = f(x) = \langle w, x \rangle + d = w^T x + d \tag{2.4}$$

where $\langle w, x \rangle$ is the dot product between the input x and the weight vector. The objective of linear SVR to find the ε-insensitive tube should be as flat as possible, which can be performed by minimising the norm of w as follows.

$$\min_{w} \frac{1}{2} ||w||^2, \tag{2.5}$$

$$\text{subject to } \begin{cases} y_i - w^T x_i - d \leq \varepsilon \\ w^T x_i + d - y_i \leq \varepsilon \end{cases} \tag{2.6}$$

The general performance of the SVR model is directly affected by the ε parameter. ε values determine the number of support vectors used to construct the decision function and hence the generalisation complexity and capacity of the SVR. Specifically, the ε value is critical for the SVR performance because a lower ε value leads to a higher probability of creating hard margins, while a higher ε value allows for greater error tolerance in training samples.

In addition, a non-linear transformation can be used with SVR to map non-linear input space to a higher dimension space where a hyperplane can easily separate data. Several kernel types are used with SVR, including linear, polynomial, and RBF. Linear kernels are typically applied to large sparse data, polynomial kernels are commonly used with image processing, and RBF kernels are general-purpose.

Table 2.1: Kernels in SVR

Kernel function	**Mathematical expression**	**Description**
Linear	$K(x,y) = x \cdot y$	x,y : data patterns
Polynomial	$K(x,y) = (\gamma(x \cdot y) + c_0)^d$	γ: slope parameter d : polynomial degree x,y : data patterns c_0: independent term
Radial basis function	$K(x,y) = \exp(\gamma \lvert x-y \rvert^2)$	γ: RBF width c_0: independent term x,y : data patterns

2.3 Decision Trees

Several factors have led to the increasing use of decision trees in solving regression and classification challenges over the past few years. Firstly, they are quite simple to construct, the outputs can be understood easily and the decision-making process is easy to follow [70]. Secondly, even with high order interactions between predictors, they are able to detect nonlinear effects on the response variable [89]. In addition, they are also highly flexible and work well when dealing with high-dimensional data due to their nonparametric nature and low bias [8]. Finally, decision trees have flexible constructions, high robustness to noise, and are capable of handling redundant attributes and missing values [43].

Decision trees are top-down, rule-based, acyclic graphical trees used in machine learning to solve classification and regression problems. Predictions are made between discrete and continuous values in the classification tree and regression tree models respectively. The tree begins with a single node at the top containing all the data referred to as the root node. There are only outgoing edges on the root node and one incoming edge on all other nodes. The nodes in the tree that do not have successor nodes (no outgoing edges) are called leaf nodes, and constitute the decision nodes. Nodes in the tree represent variables in the feature space, and their branches contain either attributes or conditions based on their types. If the attribute is numerical, there will be two branches representing conditions, and if the attribute is categorical, the branches will represent the attribute values. Obviously, the longer and wider the decision tree is, the more splits there will be in the data.

The construction of the decision tree follows a recursive divide and conquer manner. The input samples are divided recursively from the root node using selected attributes, which are chosen based on a statistical measurement. The recursive partitioning process continues until there are no attributes to be split or the partition becomes pure. In the next step, the sub-branches that have a minor influence on the tree performance are removed through ”tree pruning”.

One of the reasons for the popularity of decision trees is their ability to handle missing values. Missing values are treated as a data category when applying split rules or a surrogate rule. A surrogate rule is an alternative way to handle missing values, and is applied when the missing values prevent data splitting. The evaluation of the split rules is based on a statistically significant test (chi-square or F-test) or based on a reduction in entropy, variance, or Gini impurity. The statistical significance test is used to determine the combination of the values based on their relationship with the target value. Ideally, the values should be combined if the relationship to the target is strong; if it is weak, the values should not be combined.

The literature proposes numerous variants of the decision tree algorithms. The most common ones are classification and regression trees (Classification and Regression Trees (CART)s) developed by Breiman in 1984. The Iterative Dichotomizer 3 (ID3) was developed by Quinlan in 1986, and C4.5. The most popular decision tree algorithms are classification and regression trees C4.5 and ID3.

2.3.1 Classification and Regression Trees CARTs

Classification and Regression Trees (CART) is a binary decision tree that solves both regression and classification tasks. In the regression tree, the predicted value is interval, while in the classification tree it is categorical. The tree starts with a large dataset recursively divided into binary sub-groups. The predictors’ values are used to perform a binary split based on its impurity at every step. The measurement of node impurity is computed based on the sum of squared deviations. This measurement is calculated for all the possible splits. The node with the least squared deviations is chosen for splitting. The node impurity is computed as follows [43].

Where $\overline{y}_i$ is the average of target values belonging to partition, a partition is selected based on the attribute, leading to the smallest squared deviation sum. The partition is divided again until the impurity measure is below a threshold value or the number of samples in the partition is relatively small. In addition, CART uses the surrogate rule to handle missing values. A missing value attribute can be split by finding an attribute that is highly correlated with the original attribute and replacing it with the original one.

In the classification trees, the Gini index is used to measure node impurity.

$$GI(D) = 1 - \sum_{i=1}^{n} P_i^2, \tag{2.7}$$

where $P_i = \frac{|S_i|}{S}$ represents the ratio of the number of samples present in a dataset with respect to a particular class relative to the total number of samples within the dataset. As CART uses binary splitting, (2.7) can be written with respect to attribute t as follows.

$$GI_t(D) = \sum_{i=1}^{2} \frac{|D_i|}{D} GI(D_i) \tag{2.8}$$

where GI_i is the Gini index and the impurity reduction of attribute t can be computed as:

$$GI_{red} = GI(D) - GI_t(D) \tag{2.9}$$

The impurity reduction is computed for every feasible split, and the subset with the lowest reduction is used for splitting.

2.3.2 Iterative Dichotomizer(ID3)

As ID3 was among the first versions of decision trees and form the basis of C4.5 as will discussed in the next sub section, in this section we will give an overview of this algorithm and its selection criteria.

In its original form, ID3 is intended for non-binary trees, but can easily be changed to work in binary mode [90]. It utilises the entropy mechanism to measure node impurity. It measures the uncertainty of a class in a subset of examples. The probability Pi of any sample belonging to a particular class can be measured. The entropy can be computed for n classes in a dataset as follows:

$$En(D) = -\sum_{i=1}^{n} P_i \log_2(P_i) \tag{2.10}$$

Where $P_i = \frac{|S_i|}{|S|}$, in which S represent the total number of samples in the dataset D and S_i is the number of samples with class C_i. Moreover, for attributes t with s different values $t_1, t_2, \cdots, t_s$ the entropy is :

$$En_t(D) = \sum_{i=1}^{s} \frac{|D_i|}{|D|} \times En(D_i). \tag{2.11}$$

where D is partitioned into $D_1, D_2, \cdots, D_s$, and $En(D_i)$ is the entropy for D_i having attribute value t_i. Then the information gain is calculated as:

$$G(t) = En(D) - En_t(D). \tag{2.12}$$

The attribute that yields the highest information gain was chosen as the splitting attribute. ID3 was initially designed to handle nominal features only, but later evolved to handle continuous data. Some methods have used the midpoint procedure to address the split point, while others use the discretisation mechanism to turn continuous variables into discrete ones.

2.3.3 C4.5

This was proposed by Quinlan to overcome a limitation of ID3. The problem with ID3 is that it gives preference to the attributes with larger or missing values. In C4.5, the gain ratio rather than the entropy is employed for attribute splitting selection, and it uses the following splitting information:

$$SI_t(D) = \sum_{i=1}^{s} \frac{|D_i|}{|D|} \times \log_2(\frac{|D_i|}{|D|}) \tag{2.13}$$

where the Gain ration is computed as

$$GR(t) = \frac{G(t)}{SI_t(D)} \tag{2.14}$$

The attribute with the highest *GR* is selected as the splitting attribute.

2.4 Random Forest

Random forest (RF) [10] is an ensemble learning model that builds a collection of de-correlated trees (forest) according to the CART algorithm. Random forest uses bagging to construct the forest. Bagging generates bootstrap samples from the training data by sampling with replacements, i.e. generating samples that are the same size as training data. After the sampled data has been gathered, a CART algorithm is used to build a decision tree based on each bootstrap sample. Since each tree is derived from a different portion of the original data, ensemble de-correlated trees are formed [87].

Moreover, random forests add additional constraints while building the individual trees. The tree growth procedure is restricted to a subset of randomly selected features at each splitting point. Therefore, the Random Forests model involves randomisation with respect to the features (feature bagging) and samples (bootstrapping).

Consequently, randomisation reduces the correlation between the trees and, therefore, reduces the prediction variance [55, 51].

The Random Forest's performance is highly dependent on tuning two main parameters: the number of randomly selected features to consider at each split, and the number of trees in the forest. The latter is typically set to be a sufficiently large value, while the former is depends on the problem to be solved, and a range of values should usually be considered (starting from 5 (regression) and 3 (classification).

Once a random forest has been produced, an ensemble of trees can be used for prediction. In regression tasks, the prediction of each regression tree $\mathfrak{T}_b$ is recorded and then averaged over all the B trees to report the final prediction value as follows [51]:

$$\widehat{f_B}(x) = \frac{1}{B}\sum_{b=1}^{B} \mathfrak{T}_b(x) \tag{2.15}$$

For the classification task, the predicted class $\widehat{C_b}(x)$ from each tree is recorded, and then the majority voting is chosen[51]:

$$\widehat{C_B}(x) = majorityvote\left\{\widehat{C_b}(x)\right\}_1^B. \tag{2.16}$$

2.5 Radial Basis Function

Designing Radial Basis Function RBF involves two stages: the structure learning and parameter learning. The total number of hidden neurons, and their central location have to be identified in the initialisation phase. The value of the connection weights between the hidden layer and the output layer is estimated in the learning phase.

2.5.1 RBF architecture

RBF is a feed-forward neural network that provides non-linear transforming characteristics with a simple network structure and an efficient learning mechanism. Similar to a multi-layer network, RBF consists of three main layers: input, hidden, and output, as shown in Fig. 2.2. The input layer receives the input and directs them to the hidden layer. The number of nodes in this layer is typically equal to the number of features in the input space. The hidden layer is where the non-linear transformation is performed through the kernel activation function. Thus the input layer is directly connected to the hidden layer. The hidden layer is then connected to the output layer, whose weight is adjustable through the Orthogonal Least-Squares (Orthogonal Least-Squares (OLS)) method. The final network output provides a linear combination of the hidden layer and adjustable weights. In our research, we consider the problem with only single output, which is represented as in (2.17).,

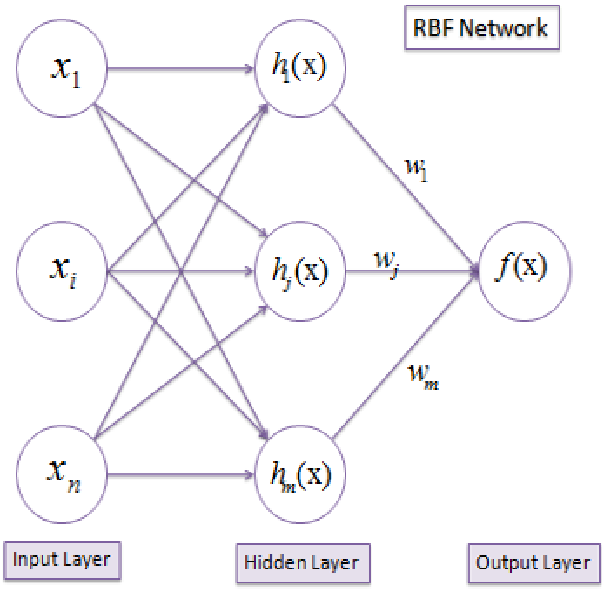

Figure 2.2: RBF network architecture.

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^{L} w_j h_j(\mathbf{x}) \tag{2.17}$$

where L represent the total number of neurons in the hidden layer, j is the index of L subnodes,w_j is the connection weight between jth neuron in the hidden layer to the output layer, and h_j is the output from jth hidden neuron. In this study, h_j is computed according to the resulting clusters from the Forward Recursive Input out clustering techniques, and with the Gaussian kernel.

$$h_j(\mathbf{x}) = \exp\left(\sum_{i=1}^{n} \frac{-D(x_i - v_{ji})}{\sigma_{ji}}\right)^2 \tag{2.18}$$

where $\mathbf{x} = (x_1, x_2, \cdot, x_n)$, v_{ji} and σ_{ji} are the neuron centre and corresponding width. Additionally, more basic functions can be conducted according to the researched object; such as linear, cubic, multi-quadric, and thin spline.

2.5.2 Center initialisation

The performance of RBF networks is sustained by the location of centres, regardless of the kernel functions being used in the hidden nodes [102]. [84, 100, 95] stated that, finding the centres of the hidden nodes is the most important step when constructing an RBF network structure. [100] outlines that this importance is due to the dependency between field distribution, underlying data structure, hidden node activation, and the node width estimation with centres location.

In a primitive study of the RBF network [11, 82] allocated all the training samples as network centres. This strategy for the allocation of centres may lead to prohibitive time consuming and overfitting when samples are huge, and noise data exists. Later, [67] proposed a solution in which centres are randomly selected from the training space. However, this technique does not consider data distribution, so these centres are unrepresentative. With the expansion in research, many centres allocation algorithms have been developed based on sequential strategy, such Orthogonal Least Squares (OLS) [18], Resource Allocating Networks (RAN) [80], the fast orthogonal estimation algorithm [122], and fast incremental supervised learning for centre selection [76]. The RBF network constructed by these approaches has a poor generalisation capability due to the redundant hidden units.

To overcome these issues, many scholars have adopted a pruning strategy to optimise the network structure; e.g. the growing and pruning algorithm proposed in [41]. In contrast, the others integrate evolution algorithms to RBF construction, such as in [110, 17, 34, 103, 104, 114]. Rather than gradually building an RBF network, other sophisticated methods have been developed based on clustering algorithms that are applied to the training samples. The centres resulting from these clustering methods are used as neurons in the RBF's hidden layers. Many scholars have proposed different algorithms based on these techniques, such k-means [96, 19, 72], Fuzzy C-means (Fuzzy C-means (FCM)) [97, 31, 96, 121], K-Nearest Neighbour (K-Nearest Neighbour (KNN) [98].

Moreover, besides considering the application of clustering methods to train samples, some studies have examined the effects of corporate output samples on the RBF clustering initialising process. The basic idea of input-output clustering is carried out by concatenating the input space with a weighted output vector, and then projecting it into the training space. Thus, the clustering method here is influenced by variance within both the input and output space [14, 92]. However, the resulting centres are representative of the underlying structure of the input space. This negatively affects the performance of the RBF network. The issues raised here have been reviewed in several studies; for example, [77] applied the conditional fuzzy mean based on the weighted information obtained from clustering the output space, and [102] demonstrated the impact when integrating information gained from the output space to minimise the upper limit of the mean squared error. Moreover, [96] employed a linear regression model to capture the relationship between the input and output spaces. A clustering algorithm was performed on the input samples, and then a linear regression performed in each cluster using both training and target values. This method has generated high performance, but requires a high number of parameter initialisations. The scholars in [100] combined the fuzzy c-mean with Particle Swarm Optimization (Particle Swarm Optimization (PSO)) to generate the optimal cluster centres. Besides, clustering algorithms and other methods such as decision tree have been studied as ways to initialise centres in RBF networks [57] and genetic algorithms [59, 101], as well as Differential Evolution [75, 115].

2.5.3 Computation of RBF output weight

The output weight from RBF network can be found by applying the ordinary least squares to solving the linear equations. Based on the training set, the hidden layer output can be defined as a matrix of activation functions H as follow:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} h_1(x_1) & h_2(x_1) & \cdots & h_P(x_1) \\ h_1(x_2) & h_2(x_2) & \cdots & h_P(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(x_n) & h_2(x_n) & \cdots & h_P(x_n) \end{bmatrix} W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_P \end{bmatrix} \tag{2.19}$$

Moreover, based on (2.17) we can defined the output matrix as follow

$$\mathbf{Y} = \mathbf{HW} \tag{2.20}$$

from the above equation the values of **W** can be calculated by using least square method follows:

$$\mathbf{W} = \left[\mathbf{H^T H}\right]^{-1} \mathbf{H^T Y} \tag{2.21}$$

Another approach to finding the optimal output weight is the gradient-based technique, which minimizes the mean square error (MSE) between the desired output and the networks' output [73]. In addition, many researchers has investigate other methods for optimising the output weigth such as genetic algorithms [47, 36], and practical swarm optimisation [36, 35].It is beyond the scope of this thesis to review these methods.

2.6 Summary

In this chapter, we review the most common machine learning models, SVR, Random Forest, Decision Tree, Linear Regression and RBF, and they are limited to one data type at a time. Then, we provide an overview of the most common techniques to deal with mixed types datasets, including the transforming methods, discretisation methods and distance measures.

Upon reviewing the literature, we discovered that most commonly used regression models only deal with one type of data at a time and that little work has been done on regression learning with heterogeneous data. Data heterogeneity is a major challenge facing data scientists, and this study attempts to fill this gap by developing learning models that can cope with heterogeneity.

The next chapter will define heterogeneous data and outline its main challenges. Also, it will provide a detailed description of the datasets that will be used in this research.

Chapter 3

Overview of Heterogeneous Data and Datasets Description

A description of heterogeneous data, its challenge, and formalisation will be presented in Section 3.1 of this chapter. Section 3.2 will describe the dataset used in this thesis as well as how it was prepared. A brief summary will be provided in Section 3.3.

3.1 Heterogeneous Data

According to [94], the significance of big data is increasing with regard to its multiple characteristics. Notably, data heterogeneity is an innate feature of big data, which increases the complexity associated with processing, managing and analysing big data. Moreover, [48] clearly indicated that mining heterogeneous data plays a vital role in knowledge discovery and pattern recognition when handling massive amounts of data.

3.1.1 Heterogeneous Data Challenge

The definition of the categorisation and description of heterogeneous data in the literature primarily reflects scholars' perspectives. For example, some have used the term heterogeneous to refer to semantic heterogeneity [54], or structural heterogeneity [28]. This term was further explained and highlighted by [65], who stated that the former refers to differentials in terms of data content and meaning. In contrast, the latter refers to differentials present in the data structure it is store in, i.e. relational vs spreadsheet. In contrast, [99] describes semantic heterogeneity to include different data types, distinct specifications or alternative interpolations of the data structure.

Moreover, some researchers used the term heterogeneous data to refer to the different data environments or data generation sources [46]. In the context of the Internet of Things (IoT) field, and extensive data analysis, heterogeneity can be divided into four levels: syntactic, conceptual, terminology, and semiotic heterogeneity. Syntactic heterogeneity represents differences in the languages used when representing two data sources; conceptual heterogeneity variations in domain modelling. Meanwhile, terminology heterogeneity is the variation in name when describing the same entity, and different interpolations for the same entity is known as semiotic heterogeneity. Building on the previous definition, [105] asserted that the essential feature of heterogeneous data concerns the *nature* of the information being stored. This has been used to refer to situations in which diverse data types are utilised to describe the information. Moreover, [123] summarised the specification of complex data heterogeneity according to three points:

- Different types of data associated with an object recorded in the dataset; Data can be stored as numerical, categorical, text, images and videos.
- Diversity in the data source; data are coming from different sources e.g. a patient's medical file contains information from different sources. such as surveys, laboratory records, x-ray images, etc.
- Data can evolve or be re-described at different times in various places. For instance, a patient may have different records associated with those doctors who

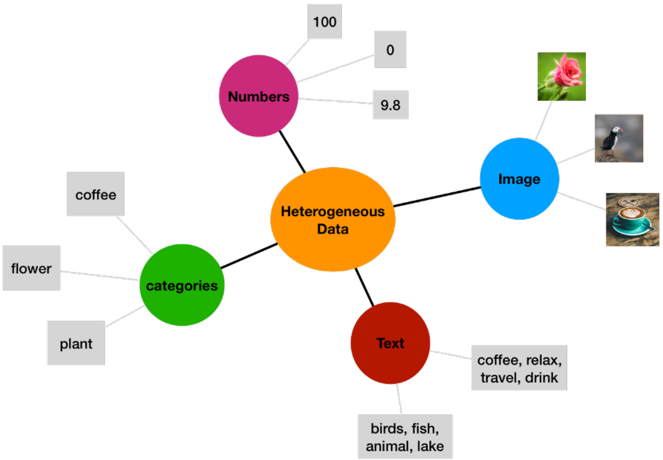

Figure 3.1: Example of a heterogeneous data types.

have examined patients at different times.

The previous research shows that data heterogeneity is a complex task, and that homogeneous learning methods need to be extended to address data heterogeneity. In this research, we focus on defining heterogeneous data according to the nature of the processed data. The following section formally describes our definition of heterogeneous data.

3.1.2 Heterogeneous Data Definition

In this research the heterogeneous data set refers to the diverse data types that describe a given instance. Each instance (record) can be described by a set of unique features. Each feature may contain multiple attributes (variables). For example, a post in a simplified example is provided in Fig. 3.1. The first column represents qualitative information in the form of categorical features, while the second column details the numerical information. In the third and fourth columns, more complex information is provided, such as images, and text. This data could include more complex types, such as graphs, signals and time series data. Thus every row represents an object or instance consistent with multiple variables.

Patient records or Internet of Things data are typical examples of heterogeneous data types [105]. For example, a patient record may contain different features; e.g. categorical features like blood type, gender, and numerical features like temperature, blood pressure, and textual information such as family history and image information like X-ray images. Although the information all describes the same object (patient), it comprises different data types that are complementary.

Data features that may be composed in the form of heterogeneous objects are as follow:

- **Structured data**

  Structured data referring to a well-defined and organised data set that can be easily processed and analysed [93]. The majority of structured data is integrated into a relational database or a well-structured file format. Simply put, the information is highly dependent on the data model, which specifies the process of data generation, integration, storage and access [24].

- **Unstructured data**

  Unlike structured data, unstructured data is usually unorganised and does not obey a common scheme. It may be textual or non-textual, and may be generated by either a human or a machine. Some researchers consider, images, video and recording files as unstructured data, while others categorise these as a multimedia data. [58] reported that approximately 90% of big data information takes an unstructured format.

- **Multimedia data**

  The multimedia data type can be classified into video, image, and audio data. According to [58], multimedia data can be classified as two types: dynamic media and static media. Dynamic media is represented by Audio and Video, and static media contains images.

3.1.3 Formalising Heterogeneous Data

In this study, the formal definition of heterogeneous data is a set of objects (records or samples), $X = \{x_i, y_i\}_{i=1}^{N}$ where X is a dataset containing N number of objects (samples or records) and x_i and y_i represent the input sample and it associated output, respectively. Each object is described by a set of features or variables $X = \{X_j\}_{j=1}^{M}$, in which M represents the total number of features (variables). The feature X_j may contain multiple descriptive attributes as described below. This research commences by considering a number of data types:

- **Numerical** Heterogeneous data generally contains numerical features X_{num}. feature can be described by a set of attributes in the data set, such that $X_{num} = \{X_{num}^{p}\}_{p=1}^{P}$, where P is the total number of numerical attributes in the dataset.

- **Categorical** (cat) A heterogeneous dataset may also contain categorical features X_{cat}. can be also described by a set of attributes, such that $X_{cat} = \{X_{cat}^{c}\}_{c=1}^{C}$ where C is the total number of categorical features present in the dataset. In addition, the attribute $\{X_{cat}^{c}\}$ contains distinct nominal values.

- **Textual** (txt)A heterogeneous element may be described using a text element. $X_{text} = \{X_{text}^{t}\}_{t=1}^{T}$, where T is the total number of text attributes present in the dataset.

Upon reviewing the literature, we discovered that most commonly used regression models limiting the number of data type being processed at a time. For the best of our knowledge, little work has been done on regression learning with heterogeneous data, as seen in the previous chapter. For that, we aim in this research to design and develop a regression model that efficiently learn from heterogeneous dataset.

Table 3.1: Description of mixed numerical and categorical datasets. Max cardinality-the total number of features after applying one-hot encoding.

Dataset	# of instances	# of numerical attributes	# of categorical attributes	Max cardinalit
Nashville	11956	9	9	74
Autos	64066	4	7	286
House	1440	15	29	176
Horse	73596	8	7	54
Bike	10886	7	4	14
KDD	3175	310	19	1340
Sales	398	7	361	993

3.2 Data Description and Preparation

The proposed model's chief objective is to handle heterogeneous data types effectively when performing regression tasks. Therefore, we first tested models across a data set comprising two different types, i.e. numerical and categorical. We then deployed our model across a more heterogeneous data set containing three different types. Section 3.2.1 provides a description of the mixed numerical and categorical dataset. For comparison purposes, the datasets selected have been the same as those in [53]. The Social Media Prediction dataset with a high level of heterogeneity is described in Section 3.2.2.

3.2.1 Mixed Numerical and Categorical datasets

3.2.1.1 Data description

We tested our models applying seven mixed categorical and numerical data sets for regression tasks. The data sets were selected from Kaggle [1], an online analytical platform upon which companies and researches release their data to enable data analysts compete to build the best models. A description of the pre-processed data is provided in Table 3.1. The total number of variables in the final data sets after dummy coding is shown as the maximum cardinality.

- Nashville Housing Data (*Nashville*)[2]:

  This data set was collected from 2013 to 2016 from the Nashville market. The ID, sale date and house image link were deleted. Also, samples with missing values were dropped. The target value was set to be the house sale price. This set consists of 9 numerical and 9 categorical variables.

- Used cars database(*Autos*)[3]:

  This data set provide information about used cars taken from Ebay Kleinanzeigen. All those records with missing values were removed. Then the variables with a single value were eliminated. Additional variables, such as car name, date of advertisement, and postcode were also dropped. The final data set consists of 4 numerical and 7 categorical variables.

- House Prices (*House*)[4]:

  The main objective of this data set is to predict the sale price of a home. Those

[1] http://www.kaggle.com

[2] https://www.kaggle.com/tmthyjames/nashville-housing-data

[3] https://data.world/data-society/used-cars-data

[4] https://www.kaggle.com/c/house-prices-advanced-regression-techniques

variables containing more than 50% missing values were eliminated, and variables with one value were also dropped. The final data set consists of 15 numerical and 29 categorical variables.

- Horse Racing in Hong Kong (*Horse*)[5]:

  This data set provides information about horse racing events held in Hong Kong. It contains two files; one describes the race, and the other the horse. The two files have been merged, and the finishing time is set as the target value. Only the information related to the horse is kept, and during the race the other information is dropped. The resulting set comprises 8 numerical variables and 7 categorical variables.

- Bike Sharing Demand (*Bike*)[6]:

  The main objective of this data set is to predict bike rental demand for the Capital Bikeshare program in Washington, D.C. They provide weather data and historical usage records to predict total number of rentals. The DateTime variable was dropped, and consequently the final data set consists of 7 numerical and 4 categorical variables.

- KDD Cup 1998 Data (*KDD*)[7]:

  The data aims to predict direct mail returns to maximise profit donation. This data includes two separate files: training and validation. The only file considered in this experiment was the training file. All the columns with 80% missing values were eliminated. Variables with one value were also removed as were variables with no description in the dictionary files. More than 90% of the remaining records have a target value of 0, so only the non-zero records are used

[5] https://www.kaggle.com/gdaley/hkracing

[6] https://www.kaggle.com/c/bike-sharing-demand

[7] https://archive.ics.uci.edu/ml/datasets/KDD+Cup+1998+Data

in the experiment. The final data consists of 310 numerical and 19 categorical variables.

- Online Product Sales (*Sales*) [8]:

  The aim of this data set is to predict online sales for a product. The Sales data sets provides the monthly online sales for a period of 12 months after the product is launched. The average online sales for the 12 months is set as the target value. The features with more than 70% missing values are deleted. After this samples with missing values are eliminated. The result is a data set with 398 categorical features and 7 numerical variables.

3.2.2 Social Media Prediction Dataset

Recently, popularity predictions for social media have attracted much attention from the academic domain. To predict popularity a score is given to images posted by users on social media. Existing research focuses on three main aspects of these posts: image, text, and user information. Some scholars [27, 88] have been able to expose the effectiveness of visual information based on the predicted scores. Others have demonstrated that textual information [61, 21, 15, 109, 60] plays a significant role in post trending, either by applying classical word embedding models such as Word2Vec, and TFIDF, or a deep neural pre-trained model, such as BERT and Glove. Additionally, the majority of these studies highlight the correlation between users and their post trending scores, so additional user information can be crawled to enrich the model [60, 40, 50]. Additional to these main features, other researchers have utilised temporal-spatial information to enhance accuracy scores [60, 21]. Another aspect of the studies on social media prediction is the regression models applied to the tasks. The fast and simple

[8] https://www.kaggle.com/c/online-sales

basic model employed to solve this problem is Linear Regression, which linearly combines all the features [68]. A tree-based Ensemble model, random forest (RF), which relies on building different trees and reporting the averaging prediction of a single tree for regression tasks is also applied in [40]. A more advanced gradient boosting decision tree has been deployed to solve this task, such as LightGBM [61, 88], XGboost [15], and CatBoost [50, 109, 60]. Among them, CatBoost achieved the best accuracy and time compliant performance.

Furthermore, more sophisticated models, including deep neural networks, have also been examined, as addressed in [21]. [61] examined the effectiveness of text feature extraction based on the Doc2Vec model for social media headline prediction. In [40] user and post-meta-data was used to train Random Forest (RF) without extracting any information from text or images. [27] combined enriched user and post features with statistical features and image object detection features from Instagram posts. In In [22] a high level pre-trained Deep Neural Network (DNN) is used to extract textual and visual features from text and images to train a DNN model. [88] focused on visual semantic features such as concept, objects and scenes using computer vision techniques to interpolate their relationship to prediction value. Both [15, 50] used a gradient boosting regression tree to predict popularity without considering temporal spatial features. [15] applied XGboost with text and visual features, and [50] applied CatBoost to users and post information only.

3.2.2.1 Data Analysis

The aim of this section is to discover and explore the main features that affect social media popularity scores. The data used in this section was obtained from ACM Multimedia Challenge SMP [113].The task set was to predict the popularity of a post prior to publication. The dataset explains the Flickr post with a set of heterogeneous features (e.g. text, temporal-spatial, and use profile). The text features, including tags and titles, play a significant role in setting the popularity prediction scores. For instance, widely used tags significantly increase the opportunity to expose the post beyond followers [21]. NLP tools are essential for text feature representation, such as GLoVe [79], and BERT [20]. User profile has a significant impact on realising the prediction target. Many researches have shown that image popularity is highly correlated to users [21, 52, 40]. For instance, number of followers has a positive impact on the popularity of a post. Users with a large number of followers are more likely to have their post viewed. Important information is missing from SMP, so we have to crawl for it using the alias provided in dataset. Moreover, the time and the location of the post may also have a great impact on its popularity. The earlier the post is uploaded the more viewers it is likely to gain. All these features will be explained in detail in the next section.

3.2.2.2 Feature Representation

Since a heterogeneous data set combines multiple data types, feature extraction is a mandatory step. High-level feature representation can be extracted from a text by adopting a Natural Language Processing NLP tool. Some of the features used can be obtained directly, while the others have to be calculated in some way.

- **User profile.** Three main features: user's ID, the total number of photos, and whether the user is a professional or not, are first determined. Then, we identified further user characteristics by crawling additional user information from their pages. These features are the number of followers, the number following, total favourites, total groups, total geotags, total tags, and total views. In addition, we construct more averaging information, such as mean favourites, mean tags and mean view.

- **Temporal-spatial features.** Geographical information regarding the posts, such as granularity levels of location and longitude, latitude information are first extracted. After which, some additional temporal information is obtained, including hour, day and week. Then, we compute hour in the week, day in the week, and week in the year.

- **Categorical** There are three main categories featured in social media posts: Category with 11 classes, Subcategory with 77 classes and concept with 668 classes.

- **Deep text feature extraction.** The text features associated with each post are represented by *Title* and *Tags* variables. Instead of using a classical machine learning text representation, such as Word2Vec or TFIDF, we adopted GloVe (Global vectors) [79], and BERT (Bidirectional Encoder Representations from Transformer) [20] to obtain word embedding vectors. GloVe is based on global corpus statistics, which are employed to represent word vectors, while BERT is a more sophisticated deep learning model that achieves a state-of-the-art performance in a wide range of text processing tasks. In addition, more text representative features are constructed when calculating total number of words and total number of characters (length) in tags and titles.

3.3 Summary

We defined heterogeneous data in this chapter and highlighted its main challenges. There is also a description of different datasets with different levels of heterogeneity. The first set includes mixed numerical and categorical features; the second comprises numerical, categorical and textual features obtained from the SMP challenge. Detailed preparation of these data has been provided.

In order to fulfil the first objective of this thesis, the next chapter will define a heterogeneous distance measurement that will be used to develop a radial basis function network for regression by adopting the forward recursive input-output approach as a structure learning for the RBF model.

Chapter 4

An Input-Output Clustering Approach to Structure Learning of Radial Basis Function Networks with Heterogeneous Data

One of the learning models that are widely used is the Radial Basis Function (RBF) network [107, 69]. Due to their fast learning, representation capabilities and straightforward structure [91]. In addition, (RBF) network is considered as the research fruits in the learning methods [108]. For that, our proposed solutions to regression heterogeneous data are to explore (RBF) network with a particular initialisation stage. The construction of (RBF) network consists of two primary stages of learning procedures: structure learning stage and parameter learning stage. The number of hidden neurons (basis functions), and the appropriate centers and width for these hidden neurons, are determined in the structure learning stage. At the same time, the determination of final-layer weights is performed in the learning stage, which easily can be computed by solving a linear regression problem.

Hence, the structure learning of RBF is a crucial process. The final system performance in terms of accuracy and fewer neurons is highly correlated to the structure learning of the RBF network [108]. In addition, The performance of RBF networks is sustained by the location of centres regardless of the kernel functions being used in hidden nodes [102]. Moreover, [84, 100] stated that finding the hidden nodes centres is critical in initialising the RBF network structure. [100] outlines this importance because of the dependency between field distribution, underlying data structure, node activation, and node width estimation with centroids location.

Much research has been conducted for RBF structure learning as described in Section 2.5.2. The intuitive structuring method randomly selected the centres from the input space [67, 106], or randomly generated centres from training space [49]. The drawback of this random selection is not guaranteed well training data representation [69]. The classical approach for (RBF) network centres allocation is to apply clustering techniques such as K-means clustering [19, 72], Fuzzy c-means (FCM) [97, 31, 96], and K-Nearest Neighbor (KNN) [97, 98]. The RBF structure learning methods based on clustering models can be divided into supervised and unsupervised learning. The former considers both input and output data in the clustering methods, and the latter considers only input samples.

There are two main weaknesses in the classical approaches (k-means, KNN) for system (RBF) network identification. First, they ignored the system information represented by output samples, although this information is major importance for network fitting and approximation [96, 78, 100].

Secondly, they partition input space into a predefined number of clusters and ignore finding the optimal number of clusters [107]. Consequently, the valuable information and knowledge provided by the output data for system identification are not used [84]. For that, input-output clustering, referred to as supervised clustering, is used for the

RBF structure learning stage. That is used for determining the neuron centroid locations and their associated width. The input-output clustering takes into account the information provided by the output space. Thus, the locations of RBF centres are affected by the input space and the deviation of the output space. Additionally, information from the output space is used to guide the clustering method [100]. The idea of using the supervised clustering approaches for RBF system identification has been developed in many different ways, as discussed in Section 2.5.2.

The remainder of the chapter is organised as follows. Section 4.1 outline the problem need to be addressed by this research. Section 4.2 presents the Heterogeneous Distance Measure (Heterogeneous Distance Measure (HDM)). Section 4.3 explain the concept of Forward Recursive Input-Output Clustering (FRIOC) approach and Section 4.4 explain the structure learning of Radial Basis Function (RBF) network. Section 4.5 and Section 4.6 present the experiments on mixed numerical and categorical datasets and on the Social Media Prediction SMP dataset,respectively. Finally, this chapter is summarized in Section 4.7.

4.1 Problem Statement

The heterogeneous dataset with M distinct features/variables and N number of input samples can defined as follows.

$$\mathfrak{D} = \{x_i^{V_m}, y_i\}_{i=1,m=1}^{N,M}$$

where a feature V_m can be described by a number of attributes $\{A_j^{V_m}\}_{j=1}^{p_{V_m}}$ such that:

$$A = \left\{A_j^{V_1}\right\}_{j=1}^{p_{V_1}} \cup \left\{A_j^{V_2}\right\}_{j=1}^{p_{V_2}} \cup \cdots \cup \left\{A_j^{V_m}\right\}_{j=1}^{p_{V_m}} \tag{4.1}$$

form total set of attributes describing the dataset $\mathfrak{D}$. So, the object x_i can be represented as a vector as follows.

$$\left[x_i^{V_1,1},x_i^{V_1,2}\cdots,x_i^{V_1,p_{V_1}},x_i^{V_2,1},x_i^{V_2,2}\cdots,x_i^{V_2,p_{V_2}},x_i^{V_m,1},x_i^{V_m,2}\cdots,x_i^{V_m,p_m}\right] \tag{4.2}$$

Based on this definition of heterogeneous datasets , there are two problems to be solved by FRIOC-RBF to minimise the Sum of Squares Error (SSE) as in (4.3). Firstly, distance calculation between heterogeneous samples, and the second problem to be resolved involves structure learning of RBF networks, in which we determine where and how many RBF kernels there are in the network, as well as their widths.

$$SSE = \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{4.3}$$

4.2 Heterogeneous Distance Measurement for Heterogeneous Dataset

The most common distance measure is well-defined and can efficiently calculate the distance for a distinct data type. These well-known distances, such as Euclidean and Overlap distances, cannot directly deal with mixed and heterogeneous data. Consequently, there is not a unified measurement for heterogeneous and mixed data. One of the proposed solutions for this issue is to apply a weighted measurement. This involves using different distance measures for different features, such as Euclidean distance and Hamming distance for numerical and categorical features. Accordingly, each distance is given a weight, which is determined by the number of attributes it possesses. The sum of these weighted distances determines the final distance between objects.

Similarly, distances between objects in a heterogeneous dataset can be calculated. This research identified data heterogeneity by identifying the major data types represented in a dataset. Next, a distance measure is identified for each feature, and its weights are computed based on the attributes associated with each feature. A final heterogeneous distance measurement HDM is derived by combining the results from the previous calculations. In the case of a heterogeneous dataset X that contains M distinctive features, the Heterogeneous Distance Measure (HDM) between two objects can be expressed as follows

$$D(x_i, x_j) = \sum_{m=1}^{M} \frac{p_{V_m}}{P} D_{V_m}(x_i^{V_m}, x_j^{V_m}) \tag{4.4}$$

The p_{V_m} and P represent the number of attributes in type m and the total number of attributes in the dataset X, respectively. Further, the D_{V_m} represents the distance measurement identified for a type m attribute.

4.3 Basic Idea of Forward Recursive Input-Output Clustering (FRIOC) Approach

The FRIOC approach consists of two main phases: In the first phase: a coarser cluster is performed to partition input space by applying a clustering method such as k-means. The second phase is the recursive clustering phase, where refined sub-clustering is performed when needed for each cluster by computing their corresponding output variation. If a cluster output variation meets the accuracy requirement, that cluster is considered a final cluster, representing a neuron in the RBF network, and no further sub clustering is performed. Nevertheless, if the output variation exceeds the accuracy level, a further sub clustering is necessary, and the cluster is considered inadequate to represent a node in the RBF system. Until every cluster can produce output variations within acceptable levels, the sub clustering procedure is repeated.

Moreover, during the experiments, it was noticed that the recursive process produced clusters with few samples and a high output variation. When this is the case, the recursive process will continue until producing a cluster with one or two samples. As a result, the recursive process will be endless and time-consuming. In order to solve these issues and reduce execution time, we limit the number of samples in each cluster and the number of recursive iterations. These issues have been addressed with two new constraints. First, the refine sub-clustering process may result in clusters with a few samples, and as their variation does not meet the required accuracy level, these clusters are considered inefficient. Secondly, we limit the number of iterations in the recursive process to reduce time complexity.

For a given dataset, $X = \{X_1, X_2, \cdots, X_n\}$ and $Y = \{y_1, y_2, \cdots, y_n\}$, $n = \{1, 2, \cdots, N\}$, where X is the input data and Y is the corresponding output values and N is the total number of samples, and α and β are the maximum and the minimum of target value respectively as in (4.5).

$$\alpha = \max_{n=1,2\cdots,N} \{y_n\} \quad \text{and} \quad \beta = \min_{n=1,2\cdots,N} \{y_n\} \tag{4.5}$$

Consider the threshold accuracy as $\varepsilon\,(\varepsilon > 0)$, which is a small real value specified by the designer based on the problem and requirements considered, and K is the number of clusters used during the initial clustering phase and is defined as the smallest integer such that:

$$\frac{\alpha - \beta}{\varepsilon} < K \tag{4.6}$$

In other words, K is the smallest integer that divides output space $[\beta, \alpha]$ into K even output intervals where the variation of each output interval is smaller than ε then we can compute K as follows:

$$K = \frac{\alpha - \beta}{\varepsilon} \tag{4.7}$$

The next section will elaborate on the FRIOC approach following the notation used here.

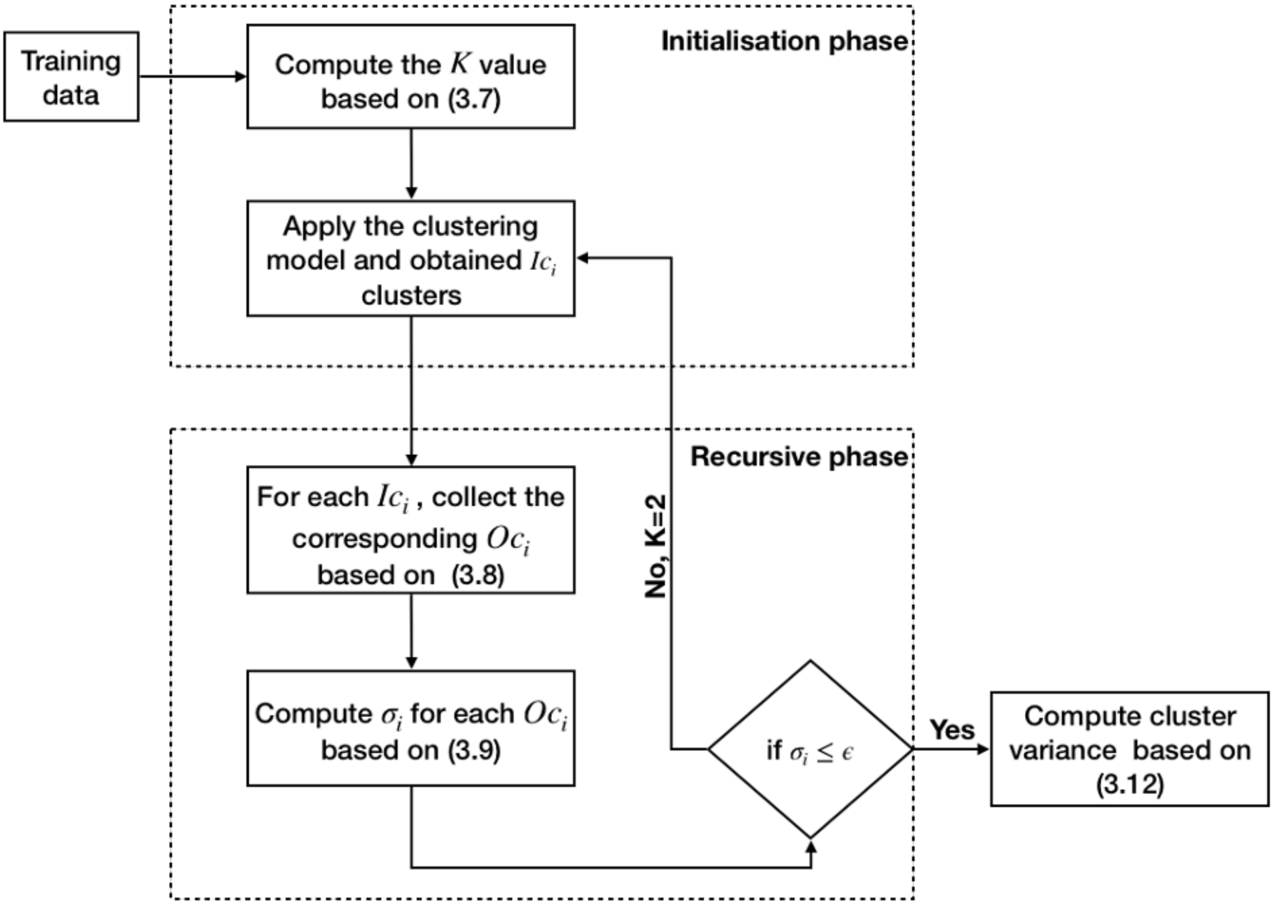
Training data
Initialisation phase
Compute the K value based on (3.7)
Apply the clustering model and obtained Ic_i clusters
Recursive phase
For each Ic_i , collect the corresponding Oc_i based on (3.8)
Compute σ_i for each Oc_i based on (3.9)
No, K=2
if $\sigma_i \leq \epsilon$
Yes
Compute cluster variance based on (3.12)

Figure 4.1: Structure simplification procedure for FRIOC.

4.3.1 FRIOC Approach

This section will describe the concept of the FRIOC approach for identifying the number of RBF centres and their location. The proposed cluster method as shown in Fig. 4.1 begins by applying a cluster model to the input samples, and the number of the clusters K is calculated using (4.7), depending on a threshold value of ε.

In most cases, the initially identified clusters are ineffective candidates to represent neurons in the RBF system because of their high variability. In this way, it is important to evaluate the output variation of each initial cluster to determine if it satisfies the output accuracy criteria in the second stage of the clustering process. Sub-clustering is conducted if a cluster's output does not meet the accuracy threshold values. The process is repeated until the required variance in cluster output is achieved, as described below.

The second phase consists in collecting all the output data Oc_i that are associated with each input cluster $Ic_i(i = 1, 2, \cdots, K)$ resulting from applying the clustering method in the initial state, as shown below:

$$Oc_i = \{y_n | x_n \in Ic_i\} \tag{4.8}$$

Following this, the standard deviation (σ_i) for each Oc_i is calculated as follows:

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{T_i} (y_i - \mu_i)^2}{T_i}} \tag{4.9}$$

Where μ_i is the mean value of the ith cluster Oc_i, and T_i is the total number of samples in cluster Ic_i. As long as the output variance ($\sigma_i \leq \varepsilon$) is less than or equal to the accuracy threshold, then a sub-clustering step for Ic_i is not required due to its representation as a neuron in an RBF network. Whereas if the output variation ($\sigma_i \geq \varepsilon$) exceeds the defined accuracy threshold, then a further sub-clustering step is carried out for Ic_i, since it is deemed inadequate as a tool to represent the system.

With $k = 2$, a clustering technique is applied to the clusters with an output variation greater than the threshold. An evaluation step is performed for each resulting cluster to determine whether it is qualified to represent the system or whether a second clustering step is required. Each cluster is then recursively evaluated until the output variation falls within an acceptable range or meets one of the previous constraints.

In summary,This approach will be conducted to learn the RBF's structure and find the optimal number and location of RBF kernels. Three main characteristics of the FRIOC approach can be derived from the above-detailed description:

- The approach is qualified for complex systems with different output variations as previously described: finer clustering for a variable region and coarse clustering for a smoother or linear region.

- It can be classified as supervised clustering as it examines both input and output information during the clustering process.

- This approach efficiently can obtain the optimal number of clusters avoiding the train-error process in the existing methods.

4.4 Radial Basis Function Network Structure Learning

After the process of FRIOC is ended, we then apply the resulting clusters for RBF network structure learning. The number of neurons in the hidden layer is equal to the number of clusters obtained from FRIOC approach, based on the idea that each cluster can be represented as a neuron.

The basic structure of RBF network consist of three main layers: one of Input layer, hidden layer, and output layer. The hidden layer contains the neurons of radial basis

Algorithm 1 Forward Recursive Input-Output Clustering (FRIOC) Algorithm

```
Input $X, Y, \varepsilon$
Initialisation
$Clust_{Final}, Cent_{Final}$ =[ ], [ ] //This is to store the final clusters and their centers
$Clust_{temp}, Cent_{temp}$ =[ ], [ ]
$Clust_{check}, Cent_{check}$ =[ ], [ ]
Calculate the initial number of clusters, $K$, based on (4.7)
$Clust_{check}, Cent_{check} = K_c luster(X, K)$
for $i = 1$ to $K$ do
    For $Ic_i \in Clust_{check} andCent_i \in Cent_{check}$, collect all the corresponding output data $Oc_i$
    Calculate the min and the max values $\beta_i$ and $\alpha_i$ for $Oc_i$
    if $\alpha_i - \beta_i \leq \varepsilon$ then
        Add $Ic_i$ to $Clust_{Final}$
        Add $Cent_i$ to $Cent_{Final}$
    else
        Add $Ic_i$ to $Clust_{temp}$
        Add $Cent_i$ to $Cent_{temp}$
    end if
end for

$Clust_{check} = Clust_{temp}$
$Cent_{check} = Cent_{temp}$
```

Algorithm 2 Forward Recursive Input-Output Clustering (FRIOC) Algorithm - Part 2

```
Recursive iteration
while $Clust_{check} \neq empty$ do
    $L =$ length of $Clust_{check}$
    $Clust_{temp}, Cent_{temp}$ =[ ], [ ]
    for $i = 1$ to $L$ do
        For $Ic_i \in Clust_{check} and Cent_i \in Cent_{check}$, collect all the corresponding output data $Oc_i$
        Calculate the min and the max values $\beta_i$ and $\alpha_i$ for $Oc_i$
        Compute $\sigma_{Oc_i}$ standard deviation based on (4.9)

        if $\sigma_{Oc_i} \leq \varepsilon$ then
            Add $Ic_i$ to $Clust_{Final}$
            Add $Cent_i$ to $Cent_{Final}$
        else
            $clusters, centers = K_c luster(Ic_i, 2)$
            Add $clusters$ to $Cent_{temp}$
            Add $centers$ to $Cent_{temp}$
        end if
    end for
    $Clust_{check} = Clust_{temp}$
    $Cent_{check} = Cent_{temp}$
end while
Outcome
$L =$ length of $Clust_{Final} = C_1, C_2, \cdots, C_L$
for $i = 1$ to $L$ do
    $\sigma_i = \sqrt{\frac{\sum_{j=1}^{T_i} (y_i - \mu_i)^2}{T_i}}$
    for $C_i$compute the cluster's width based on (4.12)
end for
The final outcomes $c = [c_1, c_2, \cdots, c_l]$, $\sigma = [\sigma_1, \sigma_2, \cdots, \sigma_l]$ represent the kernel centers and widths in RBF network
```

function and the output layer formalizes the estimated network output as in (4.10)

$$\hat{y} = \sum_{j=1}^{P} w_j h_j(\mathbf{x}) \tag{4.10}$$

where P is the total number of radial basis function neurons obtained from the FRIOC clustering process, and w_j is the connection weight between the hidden neuron and the output node. h_j is hidden neuron activation function results, and it is computed based on the Gaussian kernel as follow.

$$h_j(X) = \exp\left(\frac{-D(x_i, \nu_j)^2}{\sigma_j^2}\right) \tag{4.11}$$

where ν_j represents centres of hidden neuron and σ it representative width. x_i is the i-th input of the training samples.

In the next section, we will discuss the formula for computing the hidden layers based on distance calculations from neurons centres and width calculations.

4.4.1 RBF Parameter Learning

Kernel width is determined by measuring the distance between heterogeneous samples. Based on the distance used in the previous section, the kernel width is calculated as follows:

$$\sigma_j = \frac{\sum_{i=1}^{n_j} D(x_i, \nu_j)}{n_j} \tag{4.12}$$

The kernel centre is defined as ν_j, a distance measure is calculated as (4.4), and n_j is the number of the total samples of the cluster j.

The output weight from RBF network can be found by applying the ordinary least squares to solving the linear equations. Based on the training set, the hidden layer output can be defined as a matrix of activation functions H as follow:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} h_1(x_1) & h_2(x_1) & \cdots & h_P(x_1) \\ h_1(x_2) & h_2(x_2) & \cdots & h_P(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(x_n) & h_2(x_n) & \cdots & h_P(x_n) \end{bmatrix} W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_P \end{bmatrix} \tag{4.13}$$

Moreover, based on (4.10) we can defined the output matrix as follow

$$\mathbf{Y} = \mathbf{HW} \tag{4.14}$$

from the above equation the values of $\mathbf{W}$ can be calculated by using least square method follows:

$$\mathbf{W} = \left[\mathbf{H^T H}\right]^{-1} \mathbf{H^T Y} \tag{4.15}$$

4.4.2 Procedure of FRIOC-RBF Model

The learning steps of the proposed RBF with the FRIOC approach is presented in this section in order to help understand the combination of these two methods to train heterogeneous dataset.

step 1 Determining the M diverse features that describe the heterogeneous dataset $\mathfrak{D}$.

step 2 For each distinct feature m in the dataset, define the distance measurement D_{V_m}.

step 3 Apply FRIOC approach with k-medoids and HDM as defined in (4.4) to find RBF centers.

step 4 For each resulting clusters compute their corresponding width based on (4.12).

step 5 Start the RBF learning procedure and the matrix **H**.

step 6 Solve (4.15) to determine the RBF connection weight.

step 7 Use (4.10) to compute the final prediction output.

4.5 Mixed Dataset Experiment

In this section, we will evaluate the proposed model across several mixed numerical and categorical datasets. We will also measure the results of the proposed model against baselines and a competing mixed dataset model.

4.5.1 Datasets and Evaluation Metrics

Several benchmark datasets obtain from UCI [6] and Kaggle [1]. A detail description of these data were given in Section 3.2.1. Each dataset is split into training and testing samples with 80%-20% sizes, respectively. As a performance metric, the Mean Squared Error (MSE) was used to assess the proposed model as in (4.16).

$$MSE = \frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{N} \tag{4.16}$$

Where N is the total number of testing samples, y_i is the actual output and $\hat{y}_i$ is the RBF predicted value.

4.5.2 Model Framework and Selection

The framework of this model is presented in Fig. 4.2. Since the datasets contain two distinct features, we must first construct distance measures for each feature. Distance between numerical features is represented by Euclidean distance, while the distance

[1] http://www.kaggle.com,accessed 07 August 2022

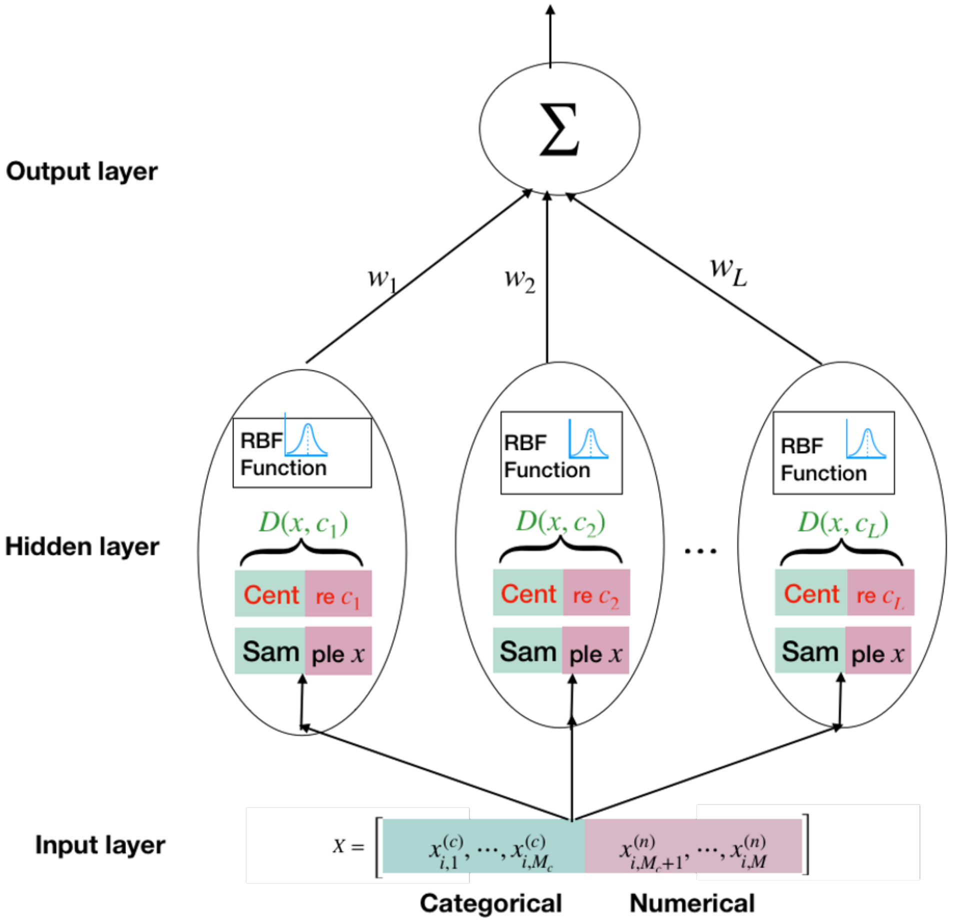

Figure 4.2: Structure of mixed numerical and categorical RBF network.

between categorical features is represented by matching distance. Based on the HDM defined in Section 4.2 and the distance measure from (4.4), we can estimate the distance between two objects in a mixed dataset as follow:

$$D(x_i, x_k) = \frac{M_c}{M} D_c(x_i^c, x_k^c) + \frac{M_n}{M} D_n(x_i^n, x_k^n) \tag{4.17}$$

The number of attributes is M, the number of categorical and numerical features are M_c and M_n. The distance measures are D_c and D_n, in this case representing the Euclidean and matching distance.

Following the definition of the Heterogeneous Distance Measure (HDM), the FRIOC algorithm is applied to determine the optimal number and location of RBF centres. Then, the RBF networks can be trained using the mixed numerical and categorical data.

4.5.3 Evaluation

Two different evaluation studies have been conducted to evaluate the performance of the proposed model in the following subsection.

Firstly, in Section 4.5.3.1 various baseline regression models are developed for comparison purposes. These models are: Linear Regression (LR), Decision Tree(CART), Random Forest RF, and Support Vector Regression (SVR). As these models can not directly deal with mixed datasets, we had to convert the categorical features to numerical using one-hot encoding.

Secondly, in Section 4.5.3.2 the performance of the proposed model is compared with a competing method. The hybrid regression tree model proposed in [53] is chosen for comparison purposes as they used the same mixed numerical and categorical datasets.

4.5.3.1 Baseline Models

From Table 4.1 we can find that the proposed algorithm outperforms the basic regression models (Linear regression, SVR, and Decision tree) for the majority of the dataset. The results showed that defining a distance measure for each data type and using FRIOC to initialise RBF centres and widths can improve learning performance in some dataset. The proposed model outperformed other models in the Autos, and Bike datasets.

The linear regression model achieved the highest MSE score for all datasets except Horse and Bike, indicating that linear regression is inefficient to train mixed datasets. In contrast, the SVR and decision tree showed a equivalent results in the Nashville, Autos, House, and KDD data. The Random Forest achieved the best results in Nashville, KDD and Sales dataset.

Meanwhile, the proposed model outperformed these models on datasets Autos, and Bike, whereas it produced a comparable result as the best model on the remaining datasets. On the Sales dataset, for instance, the Random Forest obtained a result of 7.55E+05, while the proposed model achieved a result of 9.26E+05.

4.5.3.2 Competing Methods

In comparison to the results published on [53], Table 4.2 shows a better performance in the most of the datasets. Several datasets have shown improvements in accuracy, including those on Autos, Bikes, KDDs, and Horses, while scores on Sales and Houses declined slightly. However, the improvements observed in Nashville were notable.

Table 4.2 shows the MSE test errors for both the hybrid model and the proposed model. The proposed MD_RBF outperformed the competing model in the majority of the datasets, Nashville, Autos, Horse, Bike and KDD datasets. This reduction in MSE for the Nashville datasets was significant. It decreased from $2.71E+10$to

Table 4.1: Testing MSE error in mixed numerical and categorical datasets as a comparison of the MD_RBF against the baseline models (Linear Regression, Support Vector Regression (SVR), Decision Tree, and Random Forest). Highlighted values indicate the best performance model.

Dataset	MD_RBF	Linear regression	SVR	Decision Tree	Random Forest
Nashville	2.08E+09	2.32e+32	2.78E+09	2.78E+09	**1.6827E+09**
Autos	**1.80E+06**	1.37e+22	1.89E+06	1.93E+06	1.8025E+06
House	3.93E+09	6.67e+28	7.61E+09	**2.54E+09**	2.5218E+09
Horse	1.11E+00	**1.08e+00**	3.35E+02	2.13E+00	2.6858E+01
Bike	**6.80E-03**	0.00e+00	9.58E+02	2.76E+01	3.4913E+03
KDD	3.98E+01	6.45e+21	8.36E+01	6.06E+01	**2.8061E+01**
Sales	9.26E+05	7.24e+06	1.23E+06	8.21E+05	**7.5500E+05**

$2.65E+09$ with the proposed model. The Autos datasets improved significantly to reach $1.80E$+06, while the improvement in the remaining dataset were marginal.

While the MD_RBF did not obtain better results in the case of House and Sales datasets, its results are similar to the Hybrid_model. MD_RBF as an example return was $3.93E$+09, while Hybrid_model returned $1.08E$+09, resulting in a difference of $2.85E$+09.

Table 4.2: Testing MSE error in mixed numerical and categorical datasets. Highlighted values indicate the best performance model.

Dataset	MD_RBF	Hybrid_model
Nashville	**2.08E+09**	2.71E+10
Autos	**1.80E+06**	4.97E+06
House	3.93E+09	**1.08E+09**
Horse	**1.11E+00**	1.13E+00
Bike	**6.80E-03**	31.13E-03
KDD	**3.98E+01**	8.00E+01
Sales	9.26E+05	**1.81E+05**

4.6 Heterogeneous dataset: SMP

The second stage of our experiment was to evaluate our model on a heterogeneous dataset with different features with numerical, categorical, and textual variables. The Social Media Prediction (SMP) dataset was chosen to test the proposed model. A full description of this dataset is provided in Section 3.2.2. The dataset was divided into 80%-20% train and test samples, respectively. Detailed information about the experiment is provided in this section.

4.6.1 Evaluation Metrics

For the evaluation metrics, the Spearman ranking correlation (Spearman's Rho)(4.19), and the Mean Absolute Error (MAE) (4.18) was adopted to evaluate the proposed model performance. Spearman's Rho (SR)is a ranking correlation metric ranging from 0 to 1, and the highest value indicates a better performance and can be computed as follows:

$$MAE = \frac{\sum_{i=1}^{N} |\hat{y}_i - y_i|}{N} \tag{4.18}$$

$$SR = \frac{1}{N-1} \sum_{k=1}^{N} \left(\frac{y_k - \bar{y}}{\sigma_y} \right) \left(\frac{\hat{y}_k - \bar{\hat{y}}_k}{\sigma_{\hat{y}_k}} \right) \tag{4.19}$$

Where N is the total number of training or testing samples, y_i is the actual output and $\hat{y}_i$ is the predicted value. The $\bar{\hat{y}}_k$,$\sigma_{\hat{y}_k}$ are the mean and the variance of the predicted value, and $\bar{y_k}$,σ_{y_k} are the mean and the variance of the actual output.

4.6.2 Model Framework and Selection

This section will illustrate the main aspect of the proposed model for the SMP dataset. In the first step, four main features are extracted from the Flickr dataset, and several of these features were built by crawling user profiles. See Section 3.2.2 for more information. Besides the two text features, Tags and Title, the final data also includes several numerical and categorical elements. To extract the word embedding vectors from textual features, a standard NLP model is adopted. GloVe is a Global Vector representation of text that compare the local text with a global vectors. Word embedding vectors were obtained in 50 dimensions from GloVe model to represent the text features, Tag and Titles.

The experiment begins with identifying a distance measure for each feature in the heterogeneous dataset and then compute the HDM to compute the final distance between two heterogeneous objects. The Euclidean distance is used as a distance measure for numerical and textual variables, while the matching distance is used for categorical features. The final distance equation is as in (4.20)

$$D(x_i, x_k) = \frac{M_c D_c(x_i^c, x_k^c) + M_n D_n(x_i^n, x_k^n) + M_{TI} D_{TI}(x_i^{TI}, x_k^{TI}) + M_T D_T(x_i^T, x_k^T)}{M} \tag{4.20}$$

Then the FRIOC approach with *k*-medoids is applied with the suitable distance measure in (4.20) and resulting centres are used to initialise RBF centres and widths. After finding the centres, the RBF is trained with train samples to compute the final layer weights.

4.6.3 Evaluation

Three different evaluation studies have been conducted to evaluate the performance of the proposed model in the following subsection.

Firstly, in Section 4.6.3.1 various baseline regression models are developed for comparison purposes. These models are: Linear Regression (LR), Random Forest (RF), Support Vector Regression (SVR), and Decision Tree (Decision Tree (DT))

Secondly, in Section 4.6.3.2 the performance of the proposed model is compared with the results released via the SMP competition website[2]

Finally, in Section 4.6.3.3 an ablation study has been performed by performing an overall features combination to investigate further each feature's contribution to the proposed model's performance.

[2] https://smp-challenge.com/2020/leaderboard.html,accessed 07 August 2022

4.6.3.1 Baseline Models

Four main regression models were developed for training the SMP dataset, and then compared with the proposed model. These models include Linear Regression, Random Forest, Decision tree and Support Vector Regression (SVR). Fig. 4.3 displays the results of the respective models.

The Decision Tree and Random Forrest models produced the best MAE error with 1.41 and 1.496, respectively, while the proposed HDM achieved 1.841, similar to the SVR result. As for SR results, both the DT and the RF achieved the highest score around 0.6, whereas the RBF with the proposed HDM achieved 0.3 higher than SVR.

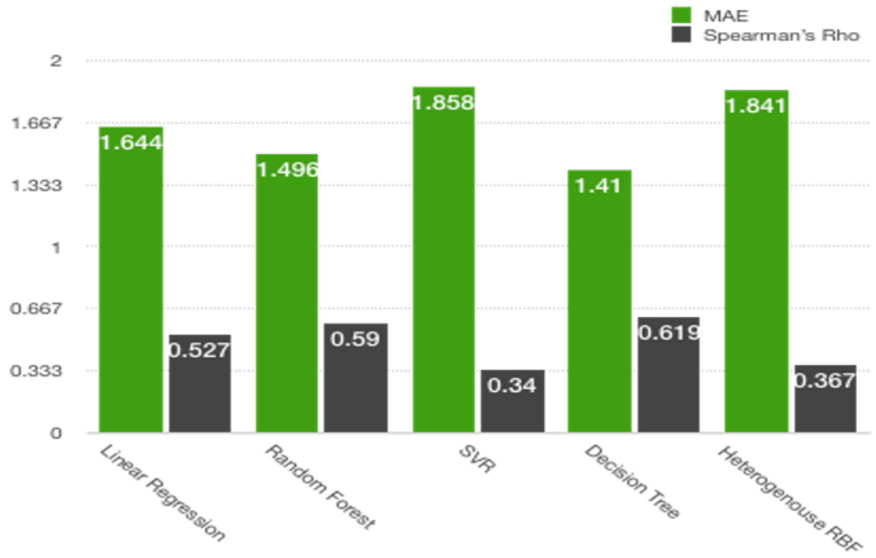

Figure 4.3: Evaluation metrics result from the testing samples of SMP dataset. MAE - Mean Absolute.

4.6.3.2 Competing Model

The proposed model outperformed the released scores, despite the poor performance of categorical features in the HRBF. By comparison, the first place provided a MAE 1.3707 and SR 0.7040, while our model produced a MAE 1.2274 and SR 0.7306. Due to the increasing dimensionality of the data, the calculation of the distance between heterogeneous samples becomes more complex. In addition, the attribute weight scheme assigns high weight to the features with the largest number of attributes.

4.6.3.3 Ablation Study

Through an ablation study, the model is evaluated by testing all the possible combinations of features in the model to examine the impact of different types of features. The results are displayed in Table 4.3.

In combining two features at once, the numerical features produced the best MAE score, around 1.95, while including textual features together (Tags and Titles) produced the highest score, 5.1760, which indicated the lowest performance.

Furthermore, the best results were obtained when the numerical and textual features (Tag and Title) were combined, 1.8572 for MAE and 0.3 for SR. However, the worst performance was achieved when considering the numerical, categorical and Title features with an MAE error of 3.0589. This is much better than considering the textual features alone.

SR results indicated a negative correlation between the output value and the categorical and textual features, but this improved when considering the numerical features to achieve 0.3670.

4.7 Summary

This chapter introduced an attribute-weighted distance measurement for the heterogeneous dataset, which can compute the distance between two heterogeneous samples based on the attribute-weighted distance scheme. This measurement is then applied to train an RBF regression model. Moreover, the FRIOC approach is followed to learn the proper structure of the RBF network by determining the optimal number and location

Table 4.3: MD_RBF performance for different combinations of features is shown as MAE and SR testing results. MAE-Mean Absolute Error, SR-Spearman's Rho The X indicates which feature or features were used. The highlighted values indicates the best performance.

numerical	X	X	X	X				X	X	
categorical	X	X			X	X		X	X	X
Tags	X		X		X		X	X		X
Title	X			X		X	X		X	X
MAE	**1.8411**	1.9539	1.9403	1.9339	2.6238	3.0205	5.1760	-0.1541	3.0589	2.6440
SR	**0.3670**	0.2136	0.3205	0.2946	0.1966	-0.0877	-0.1541	0.0781	0.3419	0.0055

of RBF centres.

Firstly, the model was evaluated on several mixed numerical and categorical datasets. In some cases, the results obtained from the experiment were promising, leading to better results than those from baselines and competing models. Secondly, the model was tested on the SMP datasets, where the heterogeneity level of the dataset has increased. In the second experiment, the results were not as promising as expected due to two reasons; first, the weighted attribute scheme emphasized features with the highest number of attributes rather than those most relevant to the data. As dimensionality increases, distance measurement becomes more complex and computationally expensive.

Towards addressing the second objective of this thesis and overcoming the limitation of this model, the following chapter introduces a two-phase Heterogeneous Radial Basis Function HRBF approach to learning from heterogeneous datasets. There is no longer a need to define and apply a heterogeneous distance measurement and to adopt a transformation/encoding steps in order to learn effectively from heterogeneous datasets.

Chapter 5

Learning Heterogeneous Data Based on Heterogeneous Radial Basis Function Network

Having seen how the distance function can be defined for heterogeneous data in the previous chapter, this chapter addresses the second objective of this thesis. That is, to construct a model that can learn directly from mixed or heterogeneous data, without requiring data types to be unified or a heterogeneous distance measures to be implemented. Thus, we present a heterogeneous RBF that can learn directly from heterogeneous data.

RBF can efficiently learn from numerical data by projecting the input space into the hidden layer, but is unable to handle direct mixed or heterogeneous data. In addition, the number of centres in the hidden layer must adequately represent the data to guarantee performance stability.

Here, we extend the RBF's ability to process mixed and heterogeneous data by introducing a Heterogeneous Radial Basis Function (HRBF) model, to be applied to the mixed and heterogeneous data. The main objective here is to learn directly from a

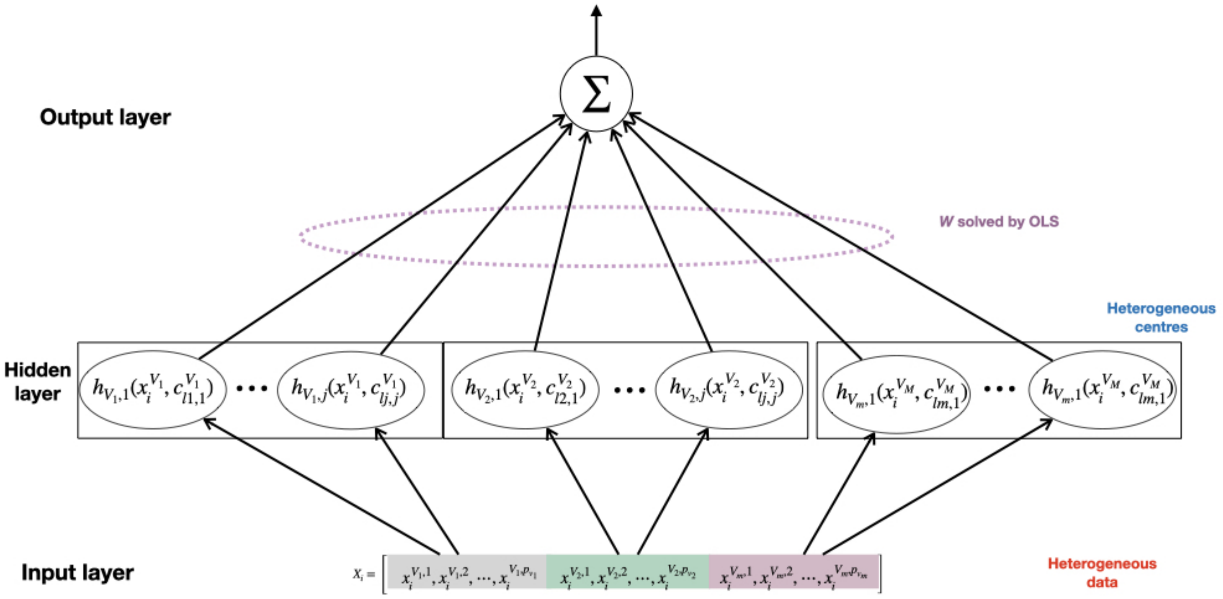
Output layer
Σ
W solved by OLS
Heterogeneous centres
Hidden layer
$h_{V_1,1}(x_i^{V_1}, c_{l1,1}^{V_1})$
$h_{V_1,j}(x_i^{V_1}, c_{lj,j}^{V_1})$
$h_{V_2,1}(x_i^{V_2}, c_{l2,1}^{V_2})$
$h_{V_2,j}(x_i^{V_2}, c_{lj,j}^{V_2})$
$h_{V_m,1}(x_i^{V_M}, c_{lm,1}^{V_M})$
$h_{V_m,1}(x_i^{V_M}, c_{lm,1}^{V_M})$
Input layer
$X_i = [x_i^{V_1,1}, x_i^{V_1,2}, \ldots, x_i^{V_1,p_{v_1}}, x_i^{V_2,1}, x_i^{V_2,2}, \ldots, x_i^{V_2,p_{v_2}}, x_i^{V_m,1}, x_i^{V_m,2}, \ldots, x_i^{V_m,p_{v_m}}]$
Heterogeneous data

Figure 5.1: Structure of HRBF network.

heterogeneous dataset without requiring a weighted distance function or unification of the data type. The model was evaluated using the same datasets as those in Chapter 4.

The remainder of the chapter is organised as follows. Section 5.1 presents the proposed Heterogeneous Radial Basis Function (HRBF) model. Section 5.2 and Section 5.3 present the experiments on mixed numerical and categorical datasets and on the Social Media Prediction SMP dataset,respectively. Finally, this chapter is summarized in Section 5.4.

5.1 Heterogeneous Radial Basis Function Model for Heterogeneous Dataset

This model introduces the Heterogeneous Radial Basis Function (HRBF), which can learn directly from heterogeneous data. The general structure of the HRBF with M different types of data, and P hidden neurons and a single output unit are given in Fig. 5.1. The HRBF input layer accepts the V_m variables, which are described by a different number of attributes. The input layer distributes each set of attributes belonging to a single data type to their represented neurons in the hidden layer. Kernel computations occur in the hidden layer, and their results are directly fed into the output layer so that the target value can be computed. The sections below describe the proposed HRBF model in more detail.

5.1.1 Structure learning of Heterogeneous Radial Basis Function model

The three-layer structure of HRBF with M different data types and P hidden neurons and one output unit is shown in Fig. 5.1. The main components of the model are as follows:

1. *Input layer.* This layer receives the heterogeneous data types as input, with M features/variables, and distributes the feature attributes' to their represented neurons in the hidden layer.

2. *Hidden layer.* This layer consists of heterogeneous units based on the heterogeneity level in the input space. Each type is represented by a set of L_{V_m} neurons in the RBF network. The FRIOC algorithms are applied separately for each data type, so as to determine how many and where the neurons are located. Then each

hidden node computes its outcome ($h_l^{V_m}$) based on the Gaussian kernel function as follow:

$$h_l^{V_m}(X^{V_m}) = \exp\left(\frac{-D_{V_m}(X^{V_m} - \nu_l^{V_m})}{\sigma_l^{V_m}}\right)^2 \tag{5.1}$$

where D_{V_m} function defines the distance calculation for type V_m. The X^{V_m} samples are the descriptions of the attributes related to the type m. The $\nu_l^{V_m}$ and $\sigma_l^{V_m}$ represent the center and variance for the lth neuron of the feature m, respectively. The output from this layer will be transmitted to the final layer for computing the final prediction values.

3. *Output layer.* The final output is calculated based on information received from the hidden layer, as the summation weights for the hidden layers are outputted as follows:

$$\hat{y} = \sum_{p=1}^{P} w_p h_p(X) \tag{5.2}$$

Where $P = \sum_{m=1}^{M} L_{V_m}$ represents the total number of neurons in the network, and w_p is the connection weights between the hidden layer and the final layer. By using the matrix notation (5.2) can be represents as follows.

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathcal{H} = \begin{bmatrix} h_1(X_1) & h_2(X_1) & \cdots & h_P(X_1) \\ h_1(X_2) & h_2(X_2) & \cdots & h_P(X_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(X_n) & h_2(X_n) & \cdots & h_P(X_n) \end{bmatrix} \quad \vec{W} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} \tag{5.3}$$

where

$$\hat{Y} = \mathcal{H}\vec{W} \tag{5.4}$$

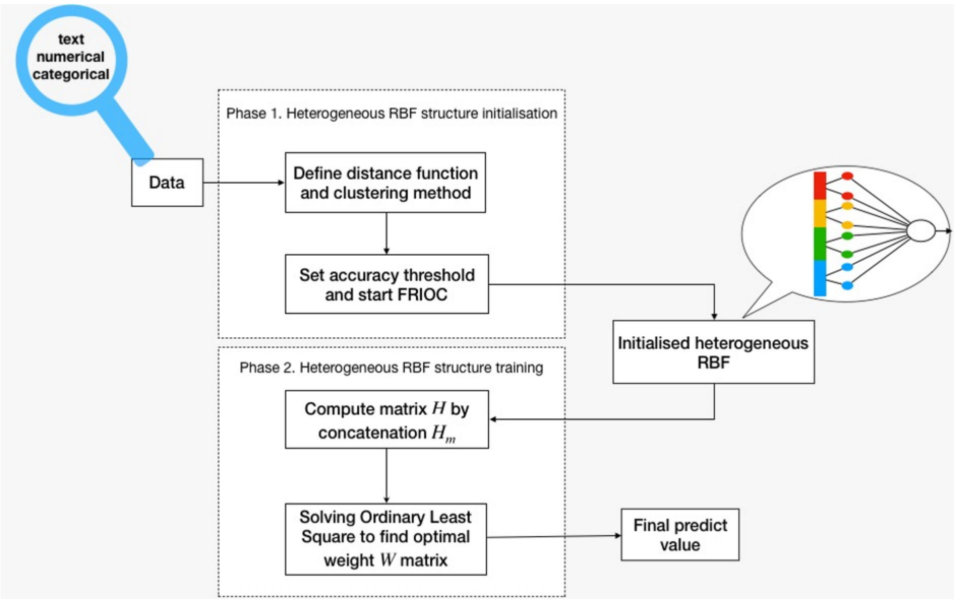
text
numerical
categorical
Data
Phase 1. Heterogeneous RBF structure initialisation
Define distance function and clustering method
Set accuracy threshold and start FRIOC
Initialised heterogeneous RBF
Phase 2. Heterogeneous RBF structure training
Compute matrix H by concatenation H_m
Solving Ordinary Least Square to find optimal weight W matrix
Final predict value

Figure 5.2: Identification process of HRBF.

which form a least square problem where the coefficient vector $\vec{W}$ can be estimated by solving the following equation.

$$\vec{W} = \left(\mathcal{H}^T\mathcal{H}\right)^{-1}\mathcal{H}^T\hat{Y} \tag{5.5}$$

5.1.2 Learning Procedure

HRBF's training phase is carried out across two phases as illustrated in Fig. 5.2. The first is HRBF centre initialisation, utilising the FRIOC approach, and the second HRBF training. These two stages are described in greater detail below.

Several steps were performed separately for each data type during the initial stage of training the HRBF. First, it is necessary to develop a distance function to be applied to both the clustering and kernel calculation processes. A clustering method is then developed using the distance measure defined previously, and the accuracy threshold set. The FRIOC will begin next, using the constructed clustering method combined with the determined accuracy threshold. Once FRIOC has been terminated, the cluster variances and cluster centres are calculated for each resulting cluster, and then used to represent the data type in the hidden layer of the HRBF network.

These steps are performed for all the data types in the dataset. The results when completing this phase will be represented as the number of kernels and their corresponding widths for every data type in the dataset, as well as the total number of neurons present in the HRBF network.

HRBF network learning is conducted across the following stage using the information gathered from the first phase. The input layer receives heterogeneous features, and then routes them to the hidden layer representing them. Each set of neurons denotes a specific type of data in the hidden layer, and the Gaussian kernel is computed using (5.1). The hidden layer's output is then concatenated to form the final matrix H as follows:

$$\mathcal{H} = [\mathbf{H_1}, \mathbf{H_2}, \cdots, \mathbf{H_M}] \tag{5.6}$$

where $m = (1, 2, \cdots, M)$ represent the level of heterogeneity in the datasets or the number of different data types. Once H has been determined, the Ordinary Least Square (5.5) can be used to calculate the optimal weight matrix.

A simple description of the training and testing function of HRBF is provided in algorithms 3,4.

In contrast to the FRIOC-RBF for heterogeneous data proposed in Chapter 3 and the HRBF model presented in this chapter, the former computes the distance between two sets of heterogeneous data by defining a heterogeneous distance measure. At the same time, the latter utilises a well-defined distance measure designed explicitly for each variable. In addition, the nodes in the hidden layer in the former model represent all the features in the heterogeneous dataset, while in the latter, each set of nodes in

Algorithm 3 Heterogeneous RBF training function

```
Input:
    $X$: is the training samples for the heterogeneous data,
    $y$: is the training outputs
Output:
    $Centers$, $\sigma$,$\vec{W}$: This function should return the centers and the width for each distinct features and the weight vector
for $i = 1$ to $M$ do
    get the train samples described by the set attributes $\{A_j^{V_i}\}_{j=1}^{pV_i}$ of the feature $i$ as $X^{V_i}$
    $Centers^{V_i}$, $\sigma^{V_i}$ = FRIOC($X^{V_i}$,$y$)
    $H^{V_i}$ = RBF ($X^{V_i}$,$Centers^{V_i}$, $\sigma^{V_i}$)    ▷ compute RBF kernels based on (5.1)
end for
$\mathcal{H} = \left[\mathcal{H}^{V_1}, \mathcal{H}^{V_2}, \cdots, \mathcal{H}^{V_m}\right]$    ▷ concatenate the $\mathcal{H}^{V_i}$ matrices to represent the heterogeneous matrix $\mathcal{H}$
$\vec{W} = \left(\mathcal{H}^T\mathcal{H}\right)^{-1}\mathcal{H}^T Y$    ▷ compute weight vector as in (5.5)
Return $Centers$, $\sigma^{V_i}$ , $\vec{W}$
    ▷ The Final outcome represents all the centers and the width parameter as well as the weight vector of Heterogeneous RBF
```

Algorithm 4 Heterogeneous RBF testing function

```
Input:
    $X$:  is the testing samples for the heterogeneous data
    $Centers$, $\sigma^{V_i}$: Heterogeneous RBF parameters
    $\vec{W}$: Heterogeneous RBF weight vector
Output:
    $\hat{Y}$: The prediction output
for $i = 1$ to $M$ do
    get the test samples described by the set attributes $\{A_j^{V_i}\}_{j=1}^{pV_i}$ of the feature $i$ as $X^{V_i}$
    $H^{V_i}$ = RBF ($X^{V_i}$,$Centers^{V_i}$, $\sigma^{V_i}$)    ▷ compute RBF kernels based on (5.1)
end for
$\mathcal{H} = \left[\mathcal{H}^{V_1}, \mathcal{H}^{V_2}, \cdots, \mathcal{H}^{V_m}\right]$    ▷ concatenate the $\mathcal{H}^{V_i}$ matrices to represent the heterogeneous matrix $\mathcal{H}$
$\hat{Y} = \mathcal{H}\vec{W}$    ▷ compute prediction output as in (5.4)
return $\hat{Y}$
```

the HRBF hidden layer represents different features. Consequently, the network in the latter model is not fully connected as the input layer maps each set of attributes representing a feature to their represented neurons in the hidden layer, as illustrated in Fig. 5.1.

Compared with the previous model proposed in Chapter 4, the main advantages of the proposed can be summarised below:

- The use of well-defined distance measurements eliminates the need to determine a unified distance measure for heterogeneous samples.

- This chapter develops a HRBF model in which each feature in the heterogeneous dataset is represented by a set of neurons with the same type in the hidden layer of the proposed model. Thus, reducing the computational cost associated with learning the structure of the HRBF network.

- Weighted distances are the most common method for defining heterogeneous measurements. Each feature in heterogeneous datasets is represented by a diverse set of attributes; defining the optimal weight for each feature adds an extra level of complexity.

5.2 Mixed Dataset Experiment

In this section, we will evaluate the HRBF model across several mixed numerical and categorical datasets. We will also measure the results of the proposed model against baselines and a competing mixed dataset model.

5.2.1 Datasets and Evaluation Metrics

This experiment can be set up by testing our model on mixed data with only numerical and categorical variables, and then comparing it to recent models using this type of

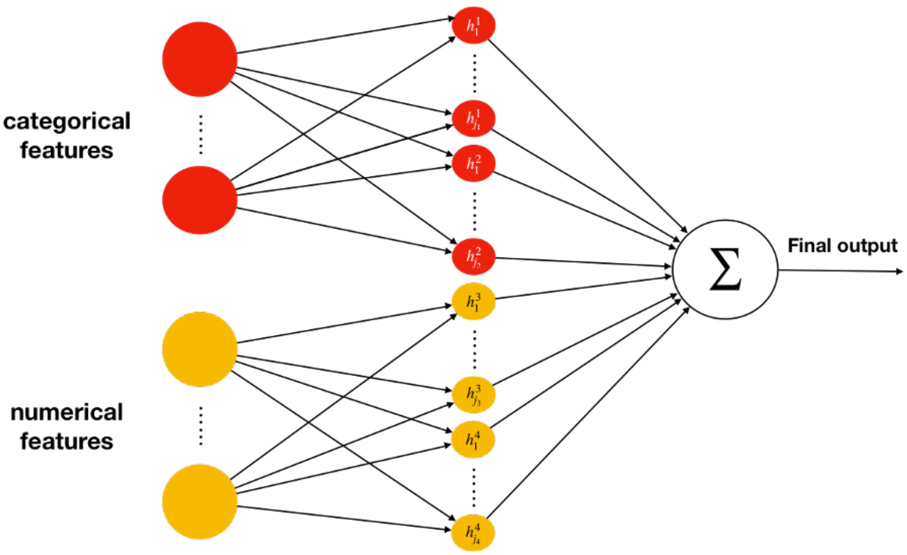

Figure 5.3: mixed numerical and categorical RBF Framework.

data. This section provides additional information about the experiment and its results. Several benchmark datasets were obtained from UCI [6] and Kaggle [1]. Section 3.2.1 provides a detailed description of this data. Datasets are split into training and testing samples, and have respective sizes of 80%-20%. As a performance metric, the Mean Squared Error (MSE) was used to assess the proposed model as in (5.7).

$$MSE = \frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{N} \tag{5.7}$$

5.2.2 Model Framework and Selection

The framework for this model is illustrated in Fig. 5.3. The datasets are represented by numerical and categorical features. The HRBF structure and parameters need to be learned in the first phase. For that, we will use a FRIOC approach using *k*-means for numerical variables and *k*-medoids for categorical variables. Numerical centres'

[1] http://www.kaggle.com, accessed 07 August 2022

widths are determined by their variations, while categorical centres' widths are decided consistently by (5.8).

After the structure identification stage, the learning process is started by computing the kernel functions in the hidden layer, the Euclidean distance is used in numerical nodes, and the overlap distance is obtained in categorical ones. In the numerical neurons, variations of each cluster are obtained to represent their associated kernel width. In the categorical neurons, the kernel width is consistent in all the categorical neurons computed based on (5.8).

$$\sigma_{ct} = \frac{d_{max}}{\sqrt{2L_m}} \tag{5.8}$$

where d_{max} is the maximum distances between any pairs of categorical centroids, and l_m is the number of categorical centroids.

5.2.3 Evaluation

Three different evaluation studies have been conducted to evaluate the performance of the proposed model in the following subsection.

Firstly, in Section 5.2.3.1 various baseline regression models are developed for comparison purposes. These models are: Linear Regression (LR), Decision Tree(CART), Random Forest RF, and Support Vector Regression (SVR). As these models can not directly deal with mixed datasets, we had to convert the categorical features to numerical using one-hot encoding. As Chapter 2 describes The Random Forest, we have only looked at this model in Section 5.3 of our evaluation experiment.

Secondly, in Section 5.2.3.2 the performance of the proposed model is compared with a competing method. The hybrid regression tree model proposed in [53] is chosen for comparison purposes as they used the same mixed numerical and categorical datasets.

Table 5.1: Testing MSE error in mixed numerical and categorical datasets as a comparison of the HRBF against the baseline models (Random Forest, Linear Regression, Support Vector Regression (SVR), and Decision Tree). Highlighted values indicate the best performance model. MSE-Mean Squared Error.

Dataset	Random Forest	Linear Regression	Decision Tree	SVR	Proposed model (HRBF)
Nashville	**1.6827E+09**	2.3185E+32	2.7844E+09	2.7844E+09	2.6519E+09
Autos	1.8025E+06	1.3674E+22	1.9331E+06	1.8952E+06	**1.7178E+06**
House	**2.5218E+09**	6.6658E+28	2.5426E+09	7.6114E+09	3.2698E+09
Horse	2.6858E+01	**1.0843E+00**	2.1289E+00	3.3516E+02	1.4797E+00
Bike	3.4913E+03	**0.00**	2.7602E+01	9.5786E+02	4.1673E+00
KDD	**2.8061E+01**	6.4496E+21	6.0597E+01	8.3561E+01	6.9699E+01
Sale	**7.5500E+05**	7.2436E+06	8.2075E+05	1.2274E+06	9.4907E+05

Finally, in Section 5.2.3.3 an ablation study has been performed by performing an overall features combination to investigate further each feature's contribution to the proposed model's performance.

5.2.3.1 Baseline models

Several well-performed baseline regression models were used to evaluate the proposed HRBF model. Table 5.1 details the corresponding outcomes. These models include Random Forest, Linear Regression, Decision Tree, and Support Vector Machine (SVR). Of all these models, Linear Regression failed to learn from all the mixed datasets. This is apparent from its high value of MSE error, which indicates poor performance.

Despite being the highest MSE score for all datasets, the linear regression model was the best for the Horse and Bike datasets. There were no significant differences between HRBF results in these datasets. According to the Horse dataset, the MSE score with linear regression is 1.08 and 1.47 with HRBF.

Although the SVR and decision tree models performed similarly, the SVR model performed significantly worse than decision tree models for the Sales dataset. The HRBF model produces results comparable to SVR and decision trees; on the Nashville, Autos and House datasets, it even outperforms them. According to the KDD, the difference between the best model and HRBF is 9.1.

For the majority of the datasets, the Random Forest was the best. The proposed model, however, performed better in the Autos dataset and produced similar results in Nashville, House, and KDD.

5.2.3.2 Competing Method

In comparison to the results published in [53], Table 5.2 shows the MSE test errors for both the hybrid model and the proposed HRBF model. The proposed HRBF outperformed the competing results in three datasets, Nashville, Autos and KDD datasets. This reduction in MSE for the Nashville datasets was significant. It decreased from $2.71E$+10 to $2.65E+09$ with the proposed model. The Autos datasets improved by 65% to reach $1.72E$+06, while the improvements in the KDD dataset were marginal. In the case of the other datasets, the proposed model performed differently. In the Horse dataset, the MSE scores differed by 0.35, while for the Sales and House datasets, the difference illustrated is considerable.

Table 5.2: Testing MSE error in the mixed numerical and categorical datasets for the Hybrid model and the HRBF model. Highlighted values indicate the best performance model.

Dataset	Hybrid model	HRBF model
Nashville	2.71E+10	**2.6519E+09**
Autos	4.97E+06	**1.7178E+06**
House	**1.08E+09**	3.2698E+09
Horse	**1.13E+00**	1.4797E+00
Bike	**31.13E-03**	4.1673E+00
KDD	8.00E+01	**6.9699E+01**
Sale	**1.81E+05**	9.4907E+05

5.2.3.3 Ablation Study

A further experiment is then conducted to examine the performance of each features alongside the proposed HRBF model. Table 5.3 shows the MSE testing error when training the numerical features with FRIOC-RBF with K-means and the categorical features with FRIOC-RBF with K-medoids.

In general, numerical features provide better results than the categorical features besides the KDD dataset. Additionally, when using both features, the results improved slightly. The MSE on the Nashville dataset increased from 2.9259E+09 for the numerical features and 9.1506E+09 for the categorical features when both were trained as 2.6519E+09.

Table 5.3: MSE for HRBF model in mixed datasets with different features combinations. MSE- Mean Squared Error

Dataset	Categorical_RBF	Numerical_RBF	HRBF
Nashville	9.1506E+09	2.9259E+09	**2.6519E+09**
Autos	1.4661E+07	1.8182E+06	**1.7178E+06**
House	9.2898E+09	3.5817E+09	**3.2698E+09**
Horse	2.6813E+03	1.6945E+00	**1.4797E+00**
Bike	3.0259E+04	3.0259E+04	**4.1673E+00**
KDD	7.0705E+01	1.0514E+02	**6.9699E+01**
Sale	1.0889E+06	9.2947E+05	**8.7790E+05**

5.3 Heterogeneous Dataset Experiment: SMP

The experiment evaluates the proposed model against a well-known regression model and a computing model on heterogeneous data with numerical, categorical, and textual variables. Detailed information about the experiment is provided in this section.

5.3.1 Datasets and Evaluation Metrics

The Social Media Prediction (SMP) dataset was collected from Flickr, which is one of the largest media sharing websites. The dataset comprises about 300K samples, describing a single social post. The primary aim of the data is to predict popularity

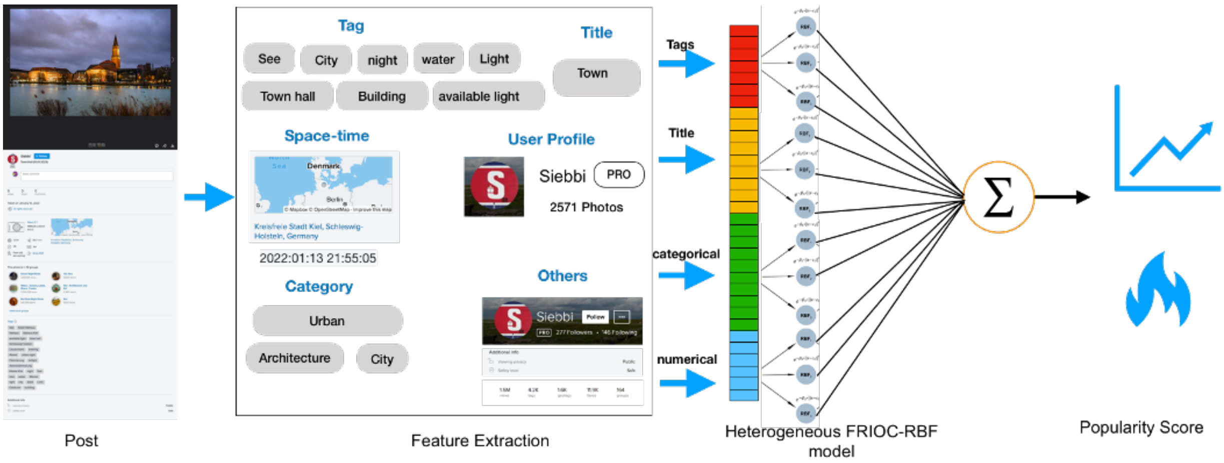

Figure 5.4: Heterogeneous Radial Basis Function (HRBF) framework.

scores for unseen posts. For evaluation purposes, we divided the data into 80% − 20% training and testing samples, respectively. For the evaluation metrics, the Spearman ranking correlation (Spearman's Rho) (5.10), and Mean Absolute Error (MAE) (5.9) were adopted to evaluate the proposed model's performance. Spearman's Rho is a ranking correlation metric ranging from 0 to 1, with the highest value indicating a better performance

$$MAE = \frac{\sum_{i=1}^{N} |\hat{y}_i - y_i|}{N} \tag{5.9}$$

$$SR = \frac{1}{N-1} \sum_{k=1}^{N} \left(\frac{y_k - \bar{y}}{\sigma_y} \right) \left(\frac{\hat{y}_k - \bar{\hat{y}}_k}{\sigma_{\hat{y}_k}} \right) \tag{5.10}$$

5.3.2 Model Framework and Selection

This section will illustrate the main aspect of the model proposed for the SMP dataset. Fig. 5.4 details the main components of the model, which are feature extraction, and the heterogeneous regression model. In the first step, four main features were extracted from the Flickr dataset; several of which were built by crawling user profiles.

See Section 3.2.2 for more information. Additional to the two text features, Tags and Title, the final data also includes several numerical and categorical elements. To extract the word embedding vectors from the textual features, an effective NLP model is adopted. For diverse areas in NLP, BERT [20] achieved a significant state of the art performance, including for the purpose of text classification, questions and answers and translations. Word embedding vectors were obtained in 768 dimensions from the pretrained BERT model as a way to represent the text features, Tag and Titles. SMP's dataset has four unique features: two numerical categorical features and two textual features. Defining distance functions, accuracy thresholds, and clustering methods for each data type is essential before starting the structure learning process of HRBF. Textual and numerical features can be clustered using K-means with Euclidean distance, whereas categorical features use K-medoids with a matching distance. When considering the distance functions and clustering methods defined in the FRIOC approach, kernels must first be defined in order to begin training. The cluster variation represents the kernels widths for the numerical and textual nodes, while the categorical neurons have a consistent kernel width, which is computed using (5.8).

5.3.3 Evaluation

Three different evaluation studies have been conducted to evaluate the performance of the proposed model in the following subsection.

Firstly, in Section 5.3.3.1 various baseline regression models are developed for comparison purposes. These models are: Linear Regression (LR), Random Forest (RF), Support Vector Regression (SVR), and Decision Tree (DT).

Secondly, in Section 5.3.3.2 the performance of the proposed model is compared with the results released via the SMP competition website[2]

[2] https://smp-challenge.com/2020/leaderboard.html,accessed 07 August 2022

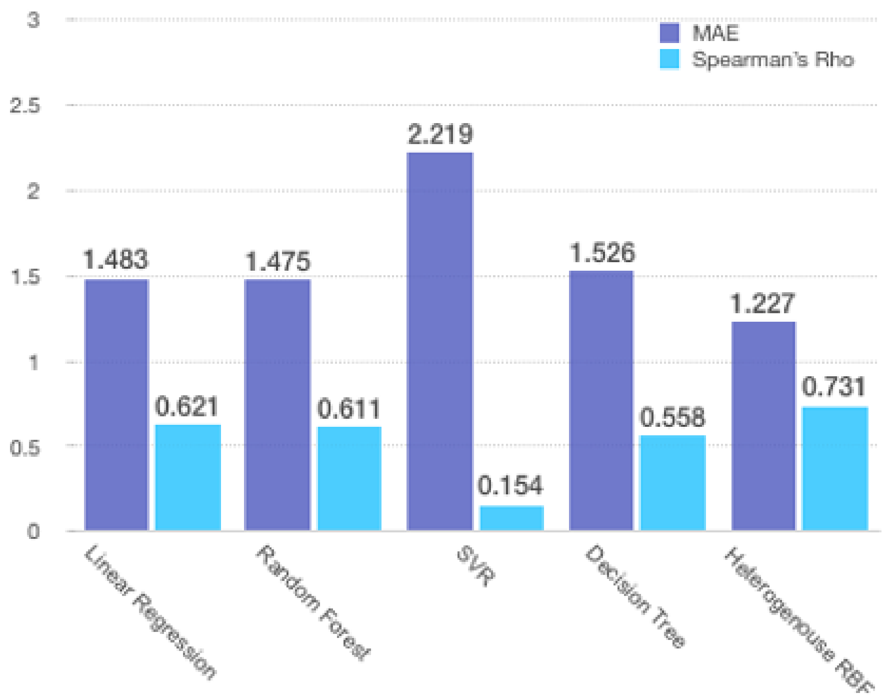

Figure 5.5: Testing error for the baseline models and HRBF. MAE-Mean Absolute Error.

Finally, in Section 5.3.3.3 an ablation study has been performed by performing an overall features combination to investigate further each feature's contribution to the proposed model's performance.

5.3.3.1 Baseline Models

Four main regression models were developed for the purpose of training the SMP dataset, and these were then compared with the proposed model. The models included Linear Regression, Random Forest, Decision tree and Support Vector Regression (SVR). Fig. 5.5 displays the results for the respective models. The SVR model with an MAE of 2.2185 produced the worst results, with the Random Forest and Linear Regression model producing similar results, of around 1.48 for MAE. Concerning the SR results, the lowest rank was achieved by SVR with the Linear Regression and

Random Forest achieving similar results at 0.61. However, the proposed model outperformed relative to the baseline scores with an MAE score of 1.227 and 0.7306 for SR rank.

5.3.3.2 Competing Results

We compare the performance of our model to the results on the leader board page for the competition website. Table 5.5 shows that the top three MAE and SR scores have been achieved. The proposed model outperformed the released scores, despite the poor performance of categorical features in the HRBF. By comparison, the first place provided a MAE 1.3707 and SR 0.7040, while our model produced a MAE 1.2274 and SR 0.7306. Based on these findings, the model is deemed generally significant, but in need of slight improvements to improve its performance.

5.3.3.3 Ablation Study

An ablation study was conducted to evaluate the model by testing all the possible combinations of features present, in an effort to examine the impact of the different feature types. The results are displayed in Table 5.4. Accordingly, the categorical features performed significantly worse in cases where each individual data type was tested separately. Their MAE score was 6.23, but the SR rank was negatively correlated. Additionally, the Title feature performed poorly, with an MAE score of 2.88 and a SR 0.06. However, the numerical and tag features performed well, achieving MAE scores of 1.56 and 1.68, respectively.

Combining the two features generates improved performance across the entire case, with numerical and tag features doing best, with an MAE of 1.30 and SR of 0.70, and categorical and title features doing worst with an MAE of 2.83 and SR of 0.09.

The performance improves further when three features are combined at a time. By combining the numerical and textual features, an MAE score of 1.24 and an SR score

Table 5.4: Proposed HRBF performance for different combinations of features is shown as MAE and SR testing results. MAE-Mean Absolute Error, SR-Spearman’s Rho. The X indicates which feature or features were used. The highlighted values indicates the best performance.

numerical	X				X	X	X				X	X	X		X
categorical		X			X			X	X		X	X		X	X
Tags			X			X		X		X	X		X	X	X
Title				X			X		X	X		X	X	X	X
MAE	1.5571	6.2372	1.6752	2.8807	1.5244	1.3084	1.4544	1.6413	2.8285	1.5663	1.2867	1.4299	1.2435	1.5435	**1.2274**
SR	0.5627	-0.0881	0.536	0.0649	0.5741	0.6999	0.6187	0.5474	0.0865	0.5778	0.7066	0.6269	0.7257	0.585	**0.7306**

Table 5.5: Proposed HRBF performance compared with top three result announced in SMP leaderboard for different combinations of features is shown as MAE and SR testing results. MAE-Mean Absolute Error, SR-Spearman's Rho.

Team	SR	MAE
ecnu_aida	0.7040	1.3707
USTC CrossModal Robot	0.6744	1.3586
UESTC IntelliGame Lab	0.6506	1.3935
HRBF model	**0.7306**	**1.2274**

of 0.73 were reached; both of which are significantly higher than for individual scores. However, the model accuracy increased by 1.5% once all the features are combined to achieve a 1.227 MAE score and 0.731 SR. Despite this, the model performance gradually increased with additional features considered, although the combination including the categorical features was the worst. The best model performance is achieved by combining all four features. To conclude, the proposed model can learn significantly from the numerical and textual features and thereby produce significant results; however, it is less effective at handling categorical features.

5.4 Summary

This chapter proposed a two-phase heterogeneous RBF to learn from heterogeneous datasets. In the first phase, a special clustering approach was implemented for each data type to indicate the numbers and locations of the REF centres. Each feature's attributes were then sent to their representative neurons in the second phase, in which the Gaussian kernel performs non-linear transformation.

We evaluated the proposed model by implementing two kinds of datasets, and compared its results to a baseline regression models and a competing model. Numerical and categorical variables are included in the first experiment type, while numerical, categorical and textual variables are included in the second type. Finally, the results were evaluated with different metrics and compared to the baseline regression models and competing models. The results show that while the model learned well from the numerical and textual data, it has not trained the categorical features sufficiently.

The next chapter fulfils the final objective of the thesis by proposing a hybrid regression model that can learn from heterogeneous data without defining a distance function or applying an encoding transformation. This model aims to train each feature/variable with a regression model and then linearly combine the results to produce the final prediction values.

Chapter 6

Hybrid-Regression model Prediction Based on Heterogeneous data

We presented two machine learning approaches for learning from heterogeneous datasets in the previous two chapters. One model defines a unified distance measurement for heterogeneous data, and the other is based on developing heterogeneous RBF networks. Our goal in this chapter is to address the final objective stated in this thesis by proposing a combined regression model capable of learning from heterogeneous datasets.

Bates and Granger first introduced combination models theory in 1969 [7], in the field of time forecasting prediction. Combining forecasts involves capturing patterns in datasets comprised of different features based on the unique features of different models. The literature related to this topic has been expanded upon significantly since then, and it has been revealed that combining multiple models greatly improves forecasting accuracy [118, 2, 39]. Moreover, many machine learning techniques developed based on the combination idea have improved predictive performance and reduce the bias. These concepts include bootstrapping, bagging, stacking, and boosting. Additionally, combining multiple models reduces the risk of using an inappropriate model for predicting the output, reduces bias, and makes the model more stable and less noisy [33]. We propose a hybrid combination regression model based on this idea as a tool to train heterogeneous data.

The key motivation behind this work emerges from these perspectives. First, the difficulty of defining a single model that can extract complete knowledge from heterogeneous data is primarily a consequence of the fact that most regression models are developed to train a single type of data, and as such may not be directly applicable to other kinds of data. Moreover, since each feature is unique; unified or simple models cannot learn or reveal pieces of information derived from this feature efficiently. In conclusion, each type of data has been extensively studied separately, with many algorithms developed to learn from single types of data. Moreover, the combination model has proven to be efficient. Here we will utilize these to establish a hybrid regression model that combines multiple models.

Although each data type has unique characteristics and applies a specific format when describing data, building a heterogeneous model and learning from each feature is proposed here. As each data type has been widely studied in isolation, and many algorithms have been proposed to learn from these single data types, we plan to take advantage of this by selecting a suitable model for each data type and combining their results to form final prediction values. Thus, heterogeneous data has been derived to learning models. Moreover, the following section will describe the models used for each data type as the base model, and their combination as high-level models.

The remainder of the chapter is organised as follows. Section 5.1 presents the proposed Hybrid-Regression model. Section 6.3 and Section 6.4 present the experiments on mixed numerical and categorical datasets and on the Social Media Prediction SMP dataset,respectively. Finally, this chapter is summarized in Section 6.5.

6.1 Problem Statement

As the problem statement stated previously in Section 4.1 in Chapter 4, The problem to be solved by the hybrid regression model is to minimise the Sum of Square Error (Sum of Square Error (SSE)) as in (6.1) by learning from heterogeneous datasets without using distance measures or encoding transformation systems that unify data types.

$$SSE = \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{6.1}$$

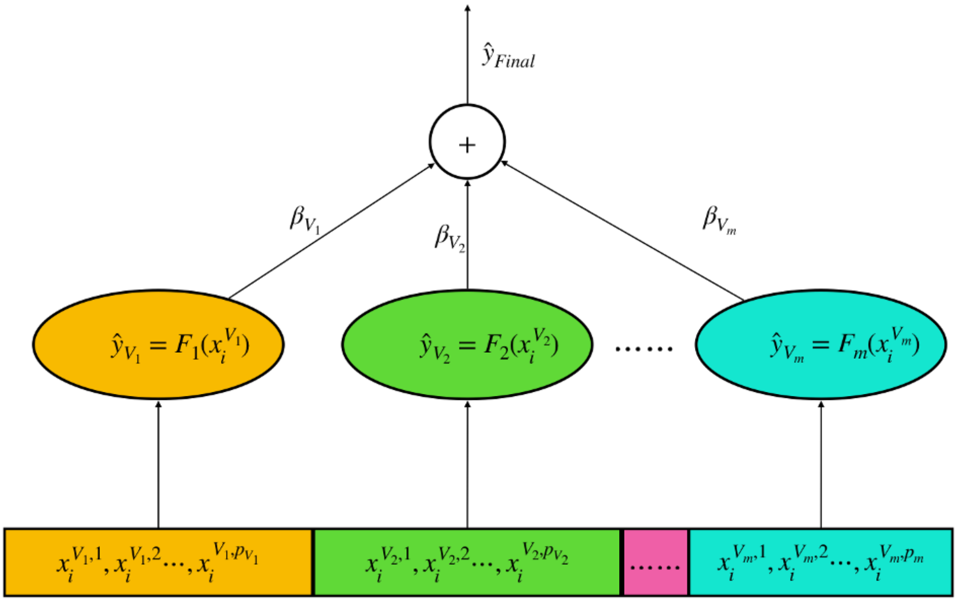
$\hat{y}_{Final}$
+
β_{V_1}
β_{V_2}
β_{V_m}
$\hat{y}_{V_1} = F_1(x_i^{V_1})$
$\hat{y}_{V_2} = F_2(x_i^{V_2})$
......
$\hat{y}_{V_m} = F_m(x_i^{V_m})$
$x_i^{V_1,1}, x_i^{V_1,2} \dots, x_i^{V_1,p_{V_1}}$
$x_i^{V_2,1}, x_i^{V_2,2} \dots, x_i^{V_2,p_{V_2}}$
......
$x_i^{V_m,1}, x_i^{V_m,2} \dots, x_i^{V_m,p_m}$

Figure 6.1: Hybrid-Regression model hierarchical structure framework. Each colour refer to different feature and each feature assigned a regession model.

6.2 Hybrid-Regression model Prediction Based on Heterogeneous data

In this section, we introduce the proposed method framework to manage heterogeneous datasets; namely, the Hybrid-Regression model Prediction Based on Heterogeneous data, designed to estimate target value. The principle behind the proposed framework is that, for each data type/variable, a regression model constructed specifically for that data type is learned, then the obtained individual models are combined to formed the final prediction model. Additionally, Unlike the current trend to use more complicated approaches to model building, which leads to the impression that complicated models are required to analyze and model complicated heterogeneous data, our approach provides a simple yet effective alternative that should be adopted. The following content of this section is organised as follows: Section 6.2.1 introduce the proposed Hybrid-Regression Model framework; and the model learning process are presented in Section 6.2.2 and Section 6.2.3; Section 6.3.

6.2.1 Framework of Hybrid-Regression Model

Fig. 6.1 illustrated the hierarchical structure of the proposed framework. The model first identified the range of diverse data types/variables included in the heterogeneous dataset. Then for each type/variable V_m on the dataset, a regression model $F_m(x_i^{V_m})$ is designed and developed. These models are then trained with their corresponding features and obtained intermediate prediction values. Finally, after all regression models are trained, their outputs are linearly combined to produce a final prediction value. A more detailed description of the framework is given in the following subsections.

6.2.2 Base Models Learning

This thesis uses three types of data: numerical, categorical, and textual data. Therefore, each of their base models will be explained in this section, which will be used during the experiment.

6.2.2.1 Learning From Categorical Features

Categorical features have a uniquely finite set of values defined according to *categories*. One of the most widely used techniques to represent categorical features is one-hot encoding. This schema uses a binary vector to represent categories where one component is set to one, and the remainder are set to zero [83, 71]. This mapping is applicable and useful when the number of categories remains small (less than ten) [71]. Using a coding system to assist data transformation increases the dimensions of the data. Thus, in this case, the categorical features have to be trained differently to enhance model efficiency. Tree-based machine learning algorithms can effectively handle high-cardinality categorical features without engaging in any external pre-processing steps (omit the need of encoding system) [71]. A state of art gradient boosting tree algorithm can be applied to train these categorical features.

In recent research three decision trees have been broadly used relying on the gradient boosting concept. These are CatBoost (Categorical Boosting), XGBoost and LightGBM. Of these, CatBoost has proven its superior performance over the two other trees, as measured by accuracy of prediction and computation time [83]. CatBoost is a decision tree model that utilises gradient boosting techniques. It was developed to solve the main issue affecting the classical gradient boosting approach; i.e. *prediction shift*. Prediction shift is a term that was used by the developer of CatBoost to describe the leakage issue affecting the target value during the training process. The chief reason for the prediction shift is that it uses the same data samples at each boosting step (i.e. estimation of the gradient). To overcome this, CatBoost constructs a decision tree model by performing ordering boost. Order boost uses a random permutation of training samples to support the ordered boost calculation. When using this technique, CatBoost reduced the overfitting of the categorical features [83, 71, 44]. Furthermore, an essential detail of CatBoost is the construction of a new categorical features based on the various combinations of categorical variables supplied. This combination is used to ensure the capture of high-order dependency, such as joint information [83].

To summarise, CatBoost is a recent gradient boosting tree model that has two main advantages over the other three models. First, they handle categorical features differently by omitting the need for pre-processing and encoding scheme steps. In addition, they enrich the features dimension by introducing new features that are effective for gradually combining categorical features. Secondly, they enhance model accuracy by using the boost ordering technique to solve the prediction shift problem, and avoid overfitting. Moreover, based on the significant body of research conducted to compare gradient boosting techniques, CatBoost outperforms alternatives in terms of accuracy and time complexity [9, 44]. We therefore employ CatBoost as a based regression model to train categorical features.

6.2.2.2 Learning From Numerical Features

A Radial Basis Function network with FRIOC clustering approach is used to train the numerical features. A detailed description of the FRIOC approach was provided in Section 4.3.1. So here we offer only a brief description.

The concept of Forward Recursive Input-Output clustering was first proposed by [84] to identify centres for Mamdani fuzzy neural networks, in which both the input and the output are utilized for clustering. FRIOC applies a coarser and a finer clustering to manage smooth and high variable regions respectively. Firstly, it applies an input cluster to divide input space into a predefined number of clusters. Then further sub-clustering is performed for those clusters with an overflow output variance. This process is repeated and guided by a validity check step, as a way to detect whether the cluster variation is set at an acceptable level. This forward recursive method facilitates the identification of a sufficient number of centres to support their representation in the RBF network. After, completing the FRIOC approach, the resulting centres can be used to represent the neurons in the RBF network.

The final k outputted clusters and centers from applying The FRIOC approach with the k-means to the numerical features can be represented as:

$$Oci(i=1,2,\cdots,K) \quad \nu_i(i=1,2,\cdots,K)$$

Then, their corresponding widths $\sigma_i(i=1,2,\cdots,K)$ for the Gaussian kernel is computes as data deviation by (6.2).

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{N_{Oci}} \left\| X_j^{num} - \nu_i \right\|}{N_{Oci}}} \tag{6.2}$$

Where X_j^{num} is the data samples that are contained in a cluster Oci, and N_{Oci} is the total number of samples for that cluster. Then, these centres and widths represent the neurons in the RBF network.

A RBF network consists of an input layer, a hidden layer (containing hidden neurons with Gaussian kernel) and an output layer, as described previously in Section 2.5. The input layer will receive the numerical features X^num and feed them to the hidden layer where the Gaussian kernel is applied to perform the non-linear transformation as follows.

$$h_j(X^{num}) = \exp\left(-\frac{\sum_{i=1}^{N} ||X_i^{num} - v_{ji}||}{\sigma_{ji}}\right)^2 \tag{6.3}$$

Then, the output of RBF is computed by (6.4),

$$y_i^{num} = \sum_{j=1}^{K} w_j h_j(X^{num}) \tag{6.4}$$

where y_i^{num} is the output prediction from numerical features, K is the total number of neurons in the RBF network, and w_j is the the connection weight between the jth neuron in the hidden layer and the output layer. By applying the matrix notation the

(6.4) can be written as follows.

$$\mathbf{Y}^{num} = \mathbf{W}\mathbf{H} \tag{6.5}$$

Then we can apply the ordinary Least Square methods to find the optimum weight values as follows.

$$\mathbf{W} = \left[\mathbf{H}^T\mathbf{H}\right]^{-1}\mathbf{H}^T\mathbf{Y}^{num} \tag{6.6}$$

6.2.2.3 Learning From Textual Features

Text features must be mapped into a set of vectors based on the Word Embedding algorithms. Word/Sentence Embedding is a commonly used NLP procedure that converts words or sentences into a set of real numbers (vectors) used in machine learning and deep learning models.

The Bidirectional Encoder Representations from Transformers (BERT) is a recent state-of-the-art pre-trained deep network NLP model. For diverse areas in NLP, BERT [20] has achieved significant and state-of-the-art performance, including for text classification, questions and answers and translations. BERT has been pre-trained on two main tasks in NLP: mask language modelling and Next sentence prediction.

There are a number of pre-trained BERT models available, and among them, we use the BERT-base-uncased model to extract sentence embedding from textual features. This results in a 768-dimensional vector for every text sample.

The extracted sentences vectors are then trained using the FRIOC-RBF method. As previously described in Section 6.2.2.2, this method consists of two phases: the initialisation phase and the training phase. During the initialisation phase, FRIOC with K-means is used to determine the optimal number and location of RBF centres. RBF kernel widths are then defined based on (6.2). The second phase involves training an RBF with Gaussian kernels based on (6.3) and(6.4). Finally, the final prediction values can be determined by solving (6.5) with the Least Square method (6.6).

6.2.2.4 The Combination of the Learning Models

Finally, once all regression models have been individually trained, a linear regression model is applied to obtain the weights of each model and to obtain the final prediction value as follows.

$$\hat{y}_i = \sum_{m=1}^{M} \beta_{V_m} F_m(x_i^{V_m}) \tag{6.7}$$

The (6.7) can be expresses by using matrix notation as follows.

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathcal{Y} = \begin{bmatrix} F_1(x_1^{V_1}) & F_2(x_1^{V_2}) & \cdots & F_m(x_1^{V_m}) \\ F_2(x_2^{V_2}) & F_2(x_2^{V_2}) & \cdots & F_m(x_2^{V_m}) \\ \vdots & \vdots & \ddots & \vdots \\ F_1(x_n^{V_1}) & F_2(x_n^{V_2}) & \cdots & F_m(x_n^{V_m}) \end{bmatrix} \quad \vec{\beta} = \begin{bmatrix} \beta_{V_1} \\ \beta_{V_2} \\ \vdots \\ \beta_{V_m} \end{bmatrix} \tag{6.8}$$

where

$$\hat{Y} = \mathcal{Y}\vec{\beta} \tag{6.9}$$

which form a least square model where the coefficient vector $\vec{\beta}$ can be estimated by solving the following equation.

$$\vec{\beta} = \left(\mathcal{Y}^T\mathcal{Y}\right)^{-1}\mathcal{Y}^T\hat{Y} \tag{6.10}$$

6.2.3 Procedure of Hybrid-Regression Model

The learning steps is presented in this section in order to help understand how the Hybrid-Regression Model combines the diverse regression models demonstrated in

Section 6.2.2.

step 1 Determining the M diverse features that describe the heterogeneous dataset $\mathfrak{D}$.

step 2 Design and develop regression models $F_m(x^{V_m})$ for each feature m

step 3 Training $F_m(x^{V_m})$ using x^{V_m} and predicting y_{V_m}.

step 4 Solve equations (6.9) and (6.10) to determine the models' weights.

step 5 Calculate the final prediction value $\hat{y}_F$ by solving equation (6.7).

6.3 Mixed Numerical and Categorical Datasets

This experiment was set up by testing the model for mixed data, applying only numerical and categorical variables, and comparing these to recent models using the specific data type. This section provides additional information regarding the experiment and its results.

6.3.1 Datasets and Evaluation Metrics

Several benchmark datasets obtained from UCI [6] and Kaggle [1] (Section 3.2.1) provide a detailed description of the data. Datasets are split into training and testing samples, with respective sizes of 80%-20%. Mean Squared Error was used as a performance metric to assess the proposed model as in (6.11).

$$MSE = \frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{N} \tag{6.11}$$

[1] http://www.kaggle.com, accessed 07 August 2022

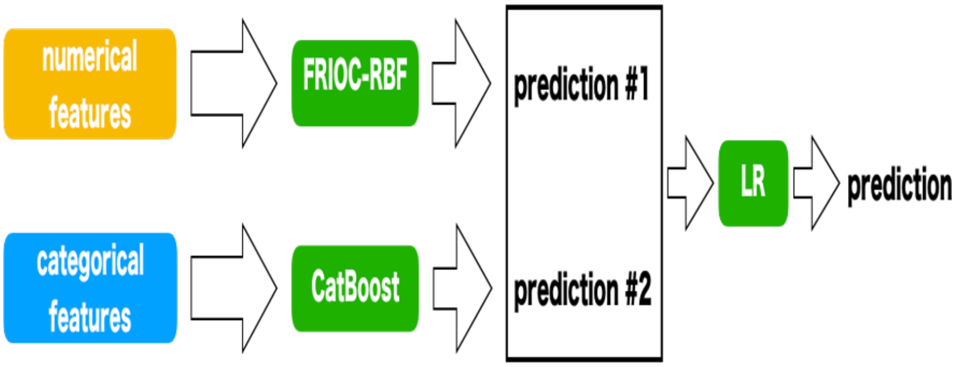

Figure 6.2: Hybrid-Regression model framework for mixed numerical and categorical dataset

6.3.2 Model Framework and Selection

The framework for this model is illustrated in Fig. 6.2. The datasets are characterised both numerically and categorically, and a regression model was selected to train and learn each feature. First, CatBoost trees can be used to train categorical features, when responses are denoted by $\hat{y}_{ct}$. Secondly, after normalization of the numerical features, the response value is produced by FRIOC-RBF with k-means as $\hat{y}_{nm}$. A linear regression model can then be applied to incorporate the results from the two learning models, to derive the final response values notated as $\hat{y}_F$.

6.3.3 Evaluation

Three different evaluation studies have been conducted to evaluate the performance of the proposed model in the following subsection.

Firstly, in Section 6.3.3.1 various baseline regression models are developed for comparison purposes. These models are: Linear Regression (LR), Decision Tree(CART), and Support Vector Regression (SVR). As these models can not directly deal with mixed datasets, we had to convert the categorical features to numerical using one-hot encoding.

Secondly, in Section 6.3.3.2 the performance of the proposed model is compared with a competing method. The hybrid regression tree model proposed in [53] is chosen for comparison purposes as they used the same mixed numerical and categorical datasets.

Finally, in Section 6.3.3.3 an ablation study has been performed by performing an overall features combination to investigate further each feature's contribution to the proposed model's performance.

6.3.3.1 Baseline models

In Table 6.1, we present the outcomes of some well-performed baseline regression models that were adopted to evaluate our proposed model. These models include Linear Regression, RandomForest, Decision Tree, Support Vector Regression (SVR) and CatBoost. For almost all the datasets, except for the Bike and Horse datasets, the Linear Regression model performed significantly worse than the others. CatBoost and Random Forest offer relatively similar results for the sets Nashville, Autos, House KDD, and Sales. The proposed model did, however, perform exceptionally well across most datasets; i.e. it reduced the MSE error for the Nashville, Autos, House, Horse, and Sales datasets by 90%.

6.3.3.2 Competing method

In comparison to the results published by [53], Table 6.2 shows the MSE test errors for both the hybrid model and the proposed model. The proposed model significantly outperformed the hybrid model across almost all the datasets analysed, according to the results. The MSE accuracy for the proposed model was 3.3668E+04, 1.0690E+03, 2.9761E+04, and 6.8467E+02 for Nashville, Autos, House, and Sales datasets, respectively. Meanwhile, the hybrid model results 2.71E+10, 4.97E+06, and 1.08E+09 for the same datasets. This improvement is significant when compared to the KDD and Horse datasets, with the proposed model resulting in a value of 3.7781E+00

Table 6.1: Testing MSE error in mixed numerical and categorical datasets as a comparison of Hybrid-Regression model against the baseline models (Linear Regression, Random Forest, Support Vector Regression (SVR), CatBoost, and Decision Tree). Highlighted values indicate the best performance model.

Dataset	Random Forest	Linear Regression	Decision Tree	SVR	CatBoost	Hybrid-Regression model
Nashville	1.6827E+09	2.3185E+32	3.1654E+09	3.1654E+09	1.4280E+09	**3.3668E+04**
Autos	1.8025E+06	1.3674E+22	1.8693E+06	1.8693E+06	1.4646E+06	**1.0690E+03**
House	2.5218E+09	6.6658E+28	7.6086E+09	7.6086E+09	1.5722E+09	**2.9761E+04**
Horse	2.6858E+01	1.0843E+00	1.5973E+00	1.5973E+00	3.0206E+02	**8.8720E-01**
Bike	3.4913E+03	**0**	1.4064E+04	1.4064E+04	7.4359E+01	1.3744E+00
KDD	2.8061E+01	6.4496E+21	3.8539E+01	3.8539E+01	2.7920E+01	**3.7781E+00**
Sale	7.5500E+05	7.2436E+06	1.2336E+06	1.2336E+06	6.6784E+05	**6.8467E+02**

and 8.872E-01, and the Hybrid model yielding an equivalent value of 8.00E+01 and 1.13E+00.

Table 6.2: Testing MSE error in the mixed numerical and categorical datasets for the Hybrid model and the proposed Hybrid-Regression model. Highlighted values indicate the best performance model.

Dataset	Hybrid model	Proposed model
Nashville	2.71E+10	**3.3668E+04**
Autos	4.97E+06	**1.0690E+03**
House	1.08E+09	**2.9761E+04**
Horse	1.13E+00	**8.872E-01**
Bike	**31.13E-03**	1.3744E+00
KDD	8.00E+01	**3.7781E+00**
Sale	1.81E+05	**6.8467E+02**

6.3.3.3 Ablation Study

The selected learning model was also used to measure each feature's performance in a subsequent experiment. Table 6.3 shows the MSE testing error for training the numerical features with FRIOC-RBF with K-means, and the categorical features with the CatBoost tree. The results from the various models proved to be similar, except for the Horse and Bike datasets, where the CatBoost model performed significantly worse than the FRIOC-RBF model; meanwhile the opposite was true for the Sales and the KDD datasets. Despite this, the linear combination for both models generated significantly stronger results than either model did separately. Considering the Nashville, Autos, and Sales datasets, the MSE score fell drastically to 3.3668E+04, 1.0690E+03, and 6.8467E+02, respectively, and this finding was similar for the other datasets.

The selected learning model is used to measure each feature's performance in a subsequent experiment. Table 6.3 shows the MSE testing error of training the numerical features with FRIOC-RBF with K-means and the categorical features with CatBoost tree. The results from the various models are similar except for the Horse and Bike datasets, where the Catboost model performed significantly worse than the FRIOC-RBF model, while the opposite was true for the Sales and the KDD datasets. Even so, the linear combination of both models shows significantly stronger results than either model separately. Considering the Nashville, Autos, and Sales datasets, the MSE score has decreased drastically to 3.3668E+04, 1.0690E+03, and 6.8467E+02, respectively, and this can also be said for the other datasets.

Table 6.3: Testing MSE for Hybrid-Regression model in mixed datasets with different combinations of features.

Dataset	Numerical features	Categorical features	Linear regression
Nashville	1.8271E+09	2.4505E+09	**3.3668E+04**
Autos	1.7738E+06	1.7544E+06	**1.0690E+03**
House	4.1806E+09	2.0932E+09	**2.9761E+04**
Horse	2.0740E+00	3.0967E+02	**1.0455E+00**
Bike	3.2722E+00	2.9426E+04	**1.3744E+00**
KDD	2.9528E+02	3.3684E+01	**3.7781E+00**
Sale	1.2464E+06	9.1486E+05	**6.8467E+02**

6.4 Heterogeneous Dataset Experiment: SMP

The experiment evaluates the proposed model against a well-known regression model and a computing model on heterogeneous data with numerical, categorical, and textual variables. Detailed information about the experiment is provided in this section.

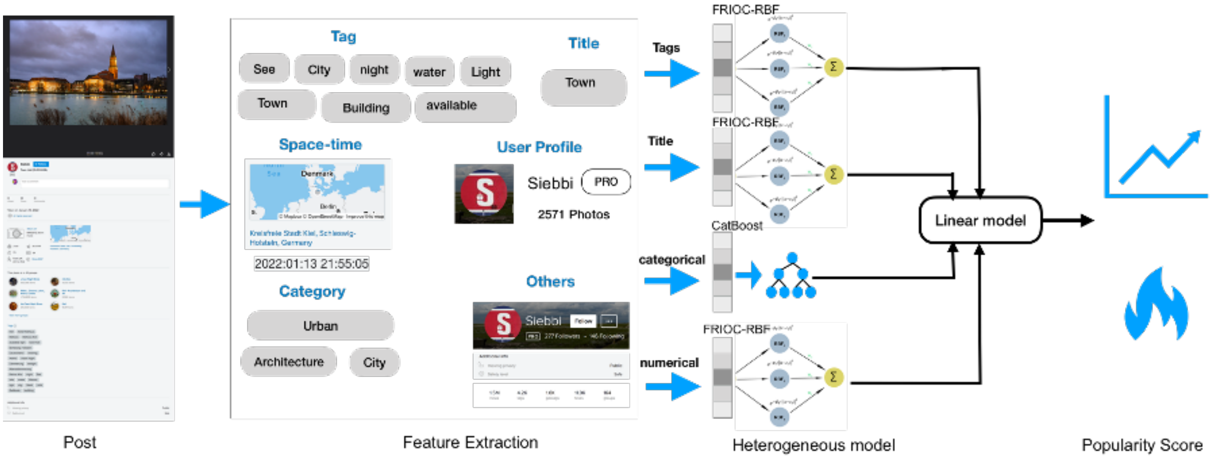

Figure 6.3: Illustration of the Hybrid-Regression model framework for SMP dataset.

6.4.1 Datasets and Evaluation Metrics

The Social Media Prediction (SMP) dataset was collected from Flickr, one of the largest media sharing websites. The dataset includes about 300K samples, each describing a single social post. The primary aim of collecting the data is to predict the popularity score for unseen posts. For evaluation purposes, we divided the data up into 80%-20% training and testing samples, respectively.

For the evaluation metrics, the Spearman ranking correlation (Spearman's Rho), and the Mean Absolute Error (MAE) was adopted to evaluate the proposed model performance (6.12). Spearman's Rho (SR) is a ranking correlation metric ranging from 0 to 1, (6.13) and the highest value indicates a better performance and can be computed as follows:

$$MAE = \frac{\sum_{i=1}^{N} |\hat{y}_i - y_i|}{N} \tag{6.12}$$

$$SR = \frac{1}{N-1} \sum_{k=1}^{N} \left(\frac{y_k - \bar{y}}{\sigma_y} \right) \left(\frac{\hat{y}_k - \bar{\hat{y}}_k}{\sigma_{\hat{y}_k}} \right) \tag{6.13}$$

where n indicates the total number of samples, $\hat{y}_k$ and y_k are the predicted and true popularity values, respectively, and $\hat{\bar{y}}_k,\bar{y}_k$ are the mean and the variance of the corresponding target values. In addition the MAE is defined as shown in Eq.(6.14).

$$MAE = \frac{1}{n}\sum_{k=1}^{n} |\hat{y}_k - y_k| \tag{6.14}$$

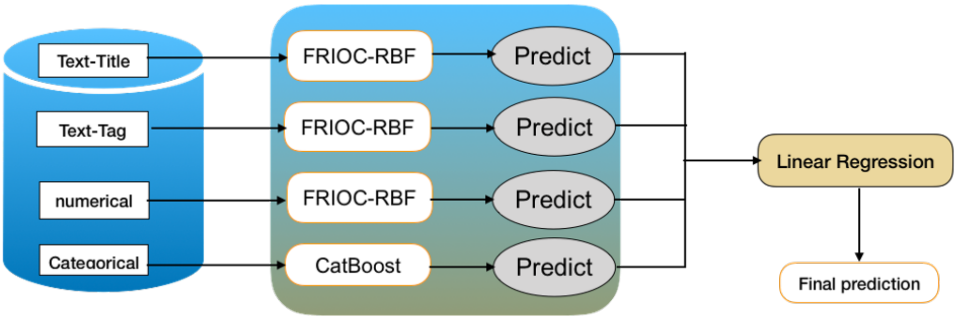

Figure 6.4: Hybrid-Regression model framework for SMP dataset.

6.4.2 Model Framework and Selection

In this section, the chief component of the proposed model for the SMP dataset is illustrated as shown in Fig. 6.3. The model is mainly for features extraction and is a heterogeneous regression model. Initially, four main features were extracted from the Flickr dataset and post. Some of these features were constructed by crawling the user's profile, while the others were computed based on the current information (for additional details, please refer to Section 3.2.2). The results described two text features, Tags and Title, a numerical and categorical features.

As shown in Fig. 6.4, for each of these features, a regression model was selected for the purpose of training and learning, as follows. Firstly, the categorical features are fed into and trained with the CatBoost tree, and its response value notated as $\hat{y}_{ct}$. Secondly, the additional features are trained by constructing a FRIOC-RBF with a k-means clustering approach, separately, and their response values notated as $\hat{y}_{nm}$,$\hat{y}_{ti}$ and, $\hat{y}_{ta}$, for numerical, Title and Tags features respectively. Finally, a linear regression model is applied to the responses of the four learning models to assign a weighting for each model outcome, and the final prediction reported as in (6.7).

6.4.3 Evaluation

Three different evaluation studies have been conducted to evaluate the performance of the proposed model in the following subsection.

Firstly, in Section 6.4.3.1 various baseline regression models are developed for comparison purposes. These models are: Linear Regression (LR), Random Forest (RF), Decision Tree (DT), XGBoost, and CatBoost.

Secondly, in Section 6.4.3.2 the performance of the proposed model is compared with the results released via the SMP competition website[2]

Finally, in Section 6.4.3.3 an ablation study has been performed by performing an overall features combination to investigate further each feature's contribution to the proposed model's performance.

6.4.3.1 Baseline models

A number of well-performed baseline regression model were used to evaluate the proposed model. Fig. 6.5 shows the corresponding outcomes. When analysing all the methods, the first thing we note is the poor performance of the Random Forest compared to the baselines. Moreover, we can observe that the proposed model outperforms the other models in terms of MAE and SR, and competitive results are achieved by CatBoost and XGBoost models, taking advantage of the designed features.

[2] https://smp-challenge.com/2020/leaderboard.html,accessed 07 August 2022

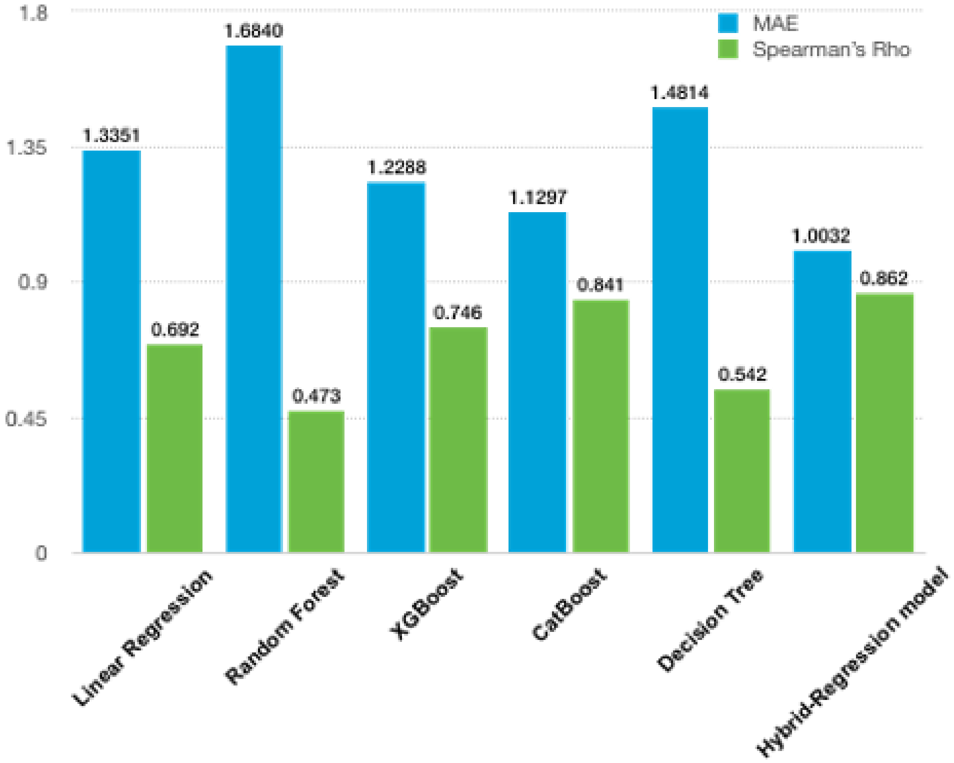

Figure 6.5: Testing error of the proposed Hybrid-Regression model against the baseline models in terms of MAE and SR. MAE-Mean Absolute Error, SR-Spearman's Rho.

6.4.3.2 Competing results

We compared the performance of our model to the results on the leader board page of the competitors website. Table 6.5 shows the top three MAE and SR scores had been achieved. The top two winning teams applied the TFIDF and word2vec scheme to extract the word embedding vectors from the textual features. While adopting BERT for the purpose of textual feature extraction greatly influences the performance of the proposed model.

6.4.3.3 Ablation Study

In this section a further experiment is conducted to evaluate the impact of different types of features through an ablation study, involving removing one type of feature from the method at a time. The results are as displayed in Table 6.4. Accordingly, we reached the following conclusions: Firstly, all the features have a positive impact on the model's performance. However, the categorical features have a large impact on the performance, while the numerical and tags features participate equally in model performance, with the impact of the title feature on performance being relatively small.

6.5 Summary

A regression model, based on a linear combination of trained models for each type of data, was presented in this chapter to train heterogeneous data. We evaluated the proposed model considering two levels of heterogeneity: mixed numerical and categorical data and a social media prediction dataset with four different data types. In addition to using the FRIOC-RBF with the k-mean clustering approach to train the numerical and

Table 6.4: Proposed Hybrid-Regression model performance for different combinations of features is shown as MAE and SR testing results. MAE-Mean Absolute Error, SR-Spearman's Rho. The X indicates which feature or features were used. The highlighted values indicates the best performance.

numerical	X				X	X		X	X
categorical		X				X	X	X	X
Tag			X		X		X	X	X
Title				X	X	X	X		X
SR	0.5399	0.8428	0.5505	0.4417	0.6675	0.8632	0.8610	0.8661	**0.8682**
MAE	1.5976	1.0925	1.5638	1.705	1.3866	1.0251	1.0293	1.0118	**1.0032**

Table 6.5: Proposed Hybrid-Regression model performance compared with top three result announced in SMP leaderboard for different combinations of features is shown as MAE and SR testing results. MAE-Mean Absolute Error, SR-Spearman's Rho.

Team	SR	MAE
ecnu_aida	0.7040	1.3707
USTC CrossModal Robot	0.6744	1.3586
UESTC IntelliGame Lab	0.6506	1.3935
Hybrid-Regression model	**0.8622**	**1.0218**

textual data, the CatBoost gradient boosting tree was used to train the categorical features. The results obtained demonstrate: 1) the combined performance of the models is better than the performance of each model individually; and 2) the proposed model showed a significantly positive performance with the mixed data types, and satisfactory results with the SMP dataset. In summary, the proposed model can be applied to any kind and level of heterogeneity in data. Future work will primarily focus on applying the proposed model to scenarios integrating more data types.

Chapter 7

Conclusions and Future Work

This study has developed a variety of regression approaches to learn from heterogeneous datasets. The aim is to predict continuous values from datasets described by various features represented by multiple data types. Learning from heterogeneous data have been shown to be complicated machine learning problems. Section 7.1 summarizes the major contributions of this thesis, while Section 7.2 examines possible future research in this research area.

7.1 Thesis Contributions and Discussion

Overall, the research in this thesis focuses on the regression problems posed by heterogeneous data with different data types. This topic is very important in areas such as medicine, recommendation systems, social prediction, and energy.

Regarding the first and second objectives, a heterogeneous distance measurement based on an attribute-weighted scheme is defined and used to train the radial basis function network. The forward recursive input output clustering approach is conducted as structured learning for the RBF network. These two objectives are fulfilled in Chapter 4.

Table 7.1: Testing MSE Error in the Mixed numerical and categorical datasets for the proposed models. Highlighted values indicate the best performance model. MSE-Mean Squared Error

Dataset	RBF with HDM	HRBF	Hybrid-Regression Model
Nashville	2.08E+09	2.6519E+09	**3.3668E+04**
Autos	1.80E+06	1.7178E+06	**1.0690E+03**
House	3.93E+09	3.2698E+09	**2.9761E+04**
Horse	1.11E+00	1.4797E+00	**8.872E-01**
Bike	6.80E-03	4.1673E+00	1.3744E+00
KDD	3.98E+01	6.9699E+01	**3.7781E+00**
Sale	9.26E+05	8.7790E+05	**6.8467E+02**

For the third objective, the Heterogeneous Radial Basis Function (HRBF) regression model presented in Chapter 5 is developed to learn from heterogeneous datasets without defining a heterogeneous distance measurement.

In Chapter 6, a simple hybrid-regression model is presented that linearly combines the results of the multi-regression model to achieve the fourth objective.

The following subsections will summarise and discuss these models' weaknesses, strengths and outcomes.

7.1.1 An Input-Output Clustering Approach to The Structure Learning of Radial Basis Function Networks with Heterogeneous Data

In this model, a RBF regression model is developed using a unified distance measure for heterogeneity data. The FRIOC approach is followed to initialise the RBF network. FRIOC is proven to be a qualified approach for a complex system with various output variations as it uses a coarser cluster for smooth or linear regions and a finer cluster for a variable region. In addition, FRIOC can be classified as a supervised clustering approach as it uses both predictor and response variables during the clustering process. This approach determines the optimal number and location of RBF centres.

Furthermore, the idea behind this approach is to define a unified distance measure that can calculate the distance between heterogeneous samples. An attribute-weighted distance measure was developed and tested on two heterogeneous datasets: mixed numerical and categorical datasets and a social media dataset with numerical, categorical and textual features. The first set of results obtained from the experiment were promising results, in some cases, led to better results than those from baselines and competing models.

In the second phase of the experiment, where the heterogeneity level of the dataset has increased, the results were not as good as expected. There may be several reasons for this. First, the defined distance measure used the weighted attribute scheme, where each feature/variable is assigned a weight based on the number of described attributes. As a result, this measure emphasises the features with the largest number of attributes, regardless of whether they are true. Although it worked well with mixed numerical and categorical data where the number of attributes is similar, in the SMP dataset, where the number of represented attributes varies as the data dimensionality increased, its performance decreased. As a result of the attribute weight scheme, features with the largest number of attributes receive a high weight, even though they may not be the most significant features. Secondly, extremely high data dimensionality has a negative influence on distance measurement. As dimensionality increases, the complexity of distance measurement and computation costs increase.

Table 7.1 summarises the results from the three models proposed in this thesis. A comparison of the results obtained from this model with the other proposed models, shows that whilst its performance is low for most datasets, it performs well in the bike dataset with an MSE score of $6.80E-03$. Overall, the RBF regression model with unified distance measure performed well with low heterogeneity but not with high data dimensions.

7.1.2 Learning Heterogeneous Data Based on Heterogeneous Radial Basis Function Network

The Heterogeneous Radial Basis Function (HRBF) regression model presented in Chapter 5 can be applied to heterogeneous datasets without defining a distance measure. Each feature or variable is represented by a set of nodes in the hidden layer of the HRBF. An appropriate clustering and distance measure was used for each feature in conjunction with the FRIOC approach to construct these nodes.

By constructing a heterogeneous Radial Basis Function (HRBF) network with heterogeneous nodes in the network's hidden layer, it was possible to efficiently extend the RBF's ability to cope with diverse data types. This model made it unnecessary to define a unified distance measure and unify data types using transformation or encoding techniques, which reduced the number of steps required to pre-process the heterogeneous data. Furthermore, the use of the FRIOC approach during the initialisation phase helped to simultaneously and effectively identify the number, location, and width of RBF kernels for each data type. Moreover, modifying and updating the nodes representing a feature in the RBF hidden layer can be carried out easily without impacting on the other representative nodes.

Compared to the previous one and its competitors, this model produced better results. Nevertheless, the model was found to be inefficient at learning from categorical features when the effect of features on the model performance was investigated.

the model performance decreased when learning from categorical features as the data dimentionality incresed. when the effect of features on the model performance was investigated.

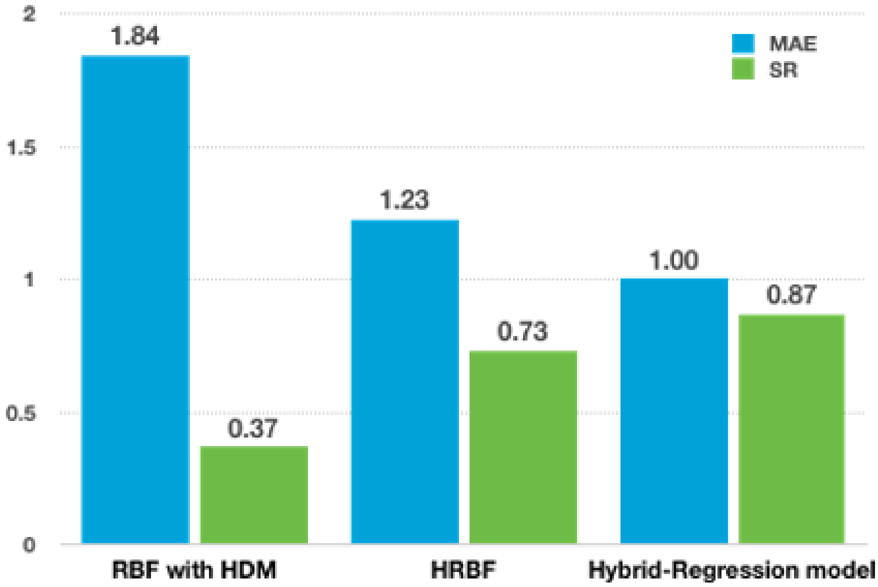

Figure 7.1: Testing error for the proposed regression models. MAE- Mean Absolute Error. SR- Spearman Correlation

7.1.3 Hybrid-Regression Model Prediction Based on Heterogeneous Data

In the absence of a universal machine learning method that can be applied to all types of problems and data, machine learning struggles to manage sets of heterogeneous data. Chapter 6 proposed a regression model that can learn from diverse data types by aggregating outcomes from various models associated with each data type to deliver a final prediction. First, a mixed numerical and categorical dataset was examined, and then a dataset with additional heterogeneity was used for the purpose of evaluation. Results from both experiments demonstrated that it this approach is effective, and its performance was significantly positive.

The majority of machine learning models are tailored to specific problems or data sources; using the most robust model for each data type is a very effective method. The following are the advantages of this approach: 1) By developing a model for each type of data, information can be extracted efficiently and the underlying knowledge revealed optimally. The CatBoost tree, for example, significantly impacts a model's performance when used to train categorical features. 2) Aggregating the results from different models is more effective than any single specific model. For example, the combined results are superior to each model individually with mixed data. 3) During the learning process, a model can be updated and modified at any time, and multiple processes run simultaneously.

Furthermore, the model eliminates the need to unify the data types and to define a distance measure in order to train heterogeneous data. With this strategy, the full knowledge provided by each feature is extracted, taking advantage of the well-performed regression model for each data type. In addition, the information can be extracted efficiently and the underlying knowledge revealed optimally by developing a model for each type of data. For example, the CatBoost tree significantly impacts the model's performance when used to train categorical features.

In addition, the use of the FRIOC approach to identify the RBF centres is beneficial as it eliminates the need for a trial and error process and determines the optimal number and location of centres. Moreover, it utilizes the knowledge and information provided by the output space.

Compared with the previous models, this model has significantly increased the learner's performance. The mixed numerical and categorical dataset dramatically decreased the MSE error for almost all the datasets, as shown in Table 7.1. In the SMP data, the evaluation metrics proved the effectiveness of this model compared with the other competitors. In Fig. 7.1, the high value of SR indicates the high correlation between the true and the predicted target values.

This model proved to be effective, and its results showed that with suitable models and simple approaches, learning problems for heterogeneous data can be solved quite easily.

7.2 Future Work

This study has proposed RBF-based approaches for training heterogeneous data. Despite some achievements, there are still areas for improvement in learning performance. The following ideas are proposed for future work:

- The proposed models in this thesis are based on RBF networks with a Gaussian kernel. The least square method was used to compute the optimal connection weights between the hidden and last layers in these models. In future, a more advanced technique should be considered to find the optimal weight values, such as genetic algorithms with practical Swarm optimisation or gradient descent.

- The distance measure proposed in Chapter 4 is based on defining weights for each feature, and the weight definition is based on the number of attributes and the construction of the features. It assigns higher weights to features with a greater number of attributes. Therefore, a more accurate weighting scheme should be explored such as swarm-based optimizer. Additionally, the performance of this model depends on what is learned during the clustering step. As a clustering model was designed to train mixed numerical and categorical data types, the model did not perform as expected. Therefore, it is necessary to develop a clustering model that can learn from large heterogeneous datasets or to adopt one from previous research into this topic.

- A more robust representation of the categorical features should be considered, as the performance of the regression model depends on the representation of the categorical features. In future, we will try to represent these features similarly to word embedding.

- The heterogeneous RBF model presented in Chapter 5 consists of nodes in the hidden layer with the same types of data as those in the heterogeneous dataset.

However, due to the high computational cost of the model, it can only be applied to a few different types of data. For that in future, a feature reduction techniques may applied to decrease the model complicity.

- A hybrid regression model was proposed in Chapter 6 based on two main models: FRIOC-RBF network and CatBoost. However, other models could be selected and combined to improve performance accuracy. Additionally, the model only tested linear combinations of the hybrid models, whereas other nonlinear combinations could be considered.

- Even though this thesis is about developing a simple and direct regression model for training heterogeneous data, it is necessary to assess the performance of the deep learning model as a solution for this data. Furthermore, more research should be carried out to include classification and clustering tasks.

- The models in these thesis only consider a limit number of different features. In future, a more complex dataset that contain more data types such as images and time series can be used to evaluated these models.

Bibliography

[1] Amir Ahmad and Lipika Dey. A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, 63(2):503–527, 2007.

[2] Saleh M Al-Alawi, Sabah A Abdul-Wahab, and Charles S Bakheit. Combining principal component regression and artificial neural networks for more accurate predictions of ground-level ozone. *Environmental Modelling & Software*, 23(4):396–403, 2008.

[3] Alex Alexandridis, Eva Chondrodima, Nikolaos Giannopoulos, and Haralambos Sarimveis. A fast and efficient method for training categorical radial basis function networks. *IEEE transactions on neural networks and learning systems*, 28(11):2831–2836, 2016.

[4] Najat Ali, Daniel Neagu, and Paul Trundle. Classification of heterogeneous data based on data type impact on similarity. In *UK Workshop on Computational Intelligence*, pages 252–263. Springer, 2018.

[5] Mohamed Alloghani, Dhiya Al-Jumeily, Jamila Mustafina, Abir Hussain, and Ahmed J Aljaaf. A systematic review on supervised and unsupervised machine learning algorithms for data science. *Supervised and unsupervised learning for data science*, pages 3–21, 2020.

[6] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[7] John M Bates and Clive WJ Granger. The combination of forecasts. *Journal of the Operational Research Society*, 20(4):451–468, 1969.

[8] Andreas Behr, Marco Giese, Katja Theune, et al. Early prediction of university dropouts-a random forest approach. *Jahrbücher für Nationalökonomie und Statistik*, 240(6):743–789, 2020.

[9] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54(3):1937–1967, 2021.

[10] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[11] David S Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

[12] Patricio Cerda, Gaël Varoquaux, and Balázs Kégl. Similarity encoding for learning with dirty categorical variables. *Machine Learning*, 107(8):1477–1494, 2018.

[13] Priyanga Chandrasekar, Kai Qian, Hossain Shahriar, and Prabir Bhattacharya. Improving the prediction accuracy of decision tree mining with data preprocessing. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 481–484. IEEE, 2017.

[14] C-L Chen, W-C Chen, and F-Y Chang. Hybrid learning algorithm for gaussian potential function networks. In *IEE Proceedings D-Control Theory and Applications*, volume 140, pages 442–448. IET, 1993.

[15] Junhong Chen, Dayong Liang, Zhanmo Zhu, Xiaojing Zhou, Zihan Ye, and Xiuyun Mo. Social media popularity prediction based on visual-textual features

with xgboost. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2692–2696, 2019.

[16] Qiong Chen, Mengxing Huang, and Hao Wang. A feature discretization method for classification of high-resolution remote sensing images in coastal areas. *IEEE Transactions on Geoscience and Remote Sensing*, 59(10):8584–8598, 2021.

[17] Yi-Hsun Cheng and Chun-Shin Lin. A learning algorithm for radial basis function networks: with the capability of adding and pruning neurons. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 2, pages 797–801. IEEE, 1994.

[18] C. F. N. Cowan, P. M. Grant, and Sheng Chen. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2(2):302–309, 1991.

[19] Christian Darken and John Moody. Fast adaptive k-means clustering: some empirical results. In *1990 IJCNN international joint conference on neural networks*, pages 233–238. IEEE, 1990.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[21] Keyan Ding, Ronggang Wang, and Shiqi Wang. Social media popularity prediction: A multiple feature fusion approach with deep neural networks. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2682–2686, 2019.

[22] Keyan Ding, Ronggang Wang, and Shiqi Wang. Social media popularity prediction: A multiple feature fusion approach with deep neural networks. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2682–2686, 2019.

[23] Carlos Eiras-Franco, Verónica Bolón-Canedo, Sabela Ramos, Jorge González-Domínguez, Amparo Alonso-Betanzos, and Juan Tourino. Multithreaded and spark parallelization of feature selection filters. *Journal of Computational Science*, 17:609–619, 2016.

[24] Basma Elsharkawy, Hatem Ahmed, and Rashed Salem. Semantic-based approach for solving the heterogeneity of clinical data. *IJCI. International Journal of Computers and Information*, 5(1):35–45, 2016.

[25] Usama Fayyad and Keki Irani. Multi-interval discretization of continuous-valued attributes for classification learning. 1993.

[26] Andrés García-Floriano, Cuauhtémoc López-Martín, Cornelio Yáñez-Márquez, and Alain Abran. Support vector regression for predicting software enhancement effort. *Information and Software Technology*, 97:99–109, 2018.

[27] Mehmetcan Gayberi and Sule Gunduz Oguducu. Popularity prediction of posts in social networks based on user, post and image features. In *Proceedings of the 11th International Conference on Management of Digital EcoSystems*, pages 9–15, 2019.

[28] Cheng Hian Goh. *Representing and reasoning about semantic conflicts in heterogeneous information systems*. PhD thesis, Massachusetts Institute of Technology, 1997.

[29] Eric Golinko, Thomas Sonderman, and Xingquan Zhu. Cnfl: categorical to numerical feature learning for clustering and classification. In *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pages 585–594. IEEE, 2017.

[30] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

[31] Victor H Grisales, José J Soriano, Sergio Barato, and Diana M Gonzalez. Robust agglomerative clustering algorithm for fuzzy modeling purposes. In *Proceedings of the 2004 American Control Conference*, volume 2, pages 1782–1787. IEEE, 2004.

[32] Cheng Guo and Felix Berkhahn. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*, 2016.

[33] Zahra Hajirahimi and Mehdi Khashei. Hybrid structures in time series modeling and forecasting: A review. *Engineering Applications of Artificial Intelligence*, 86:83–106, 2019.

[34] Hong-Gui Han, Wei Lu, Ying Hou, and Jun-Fei Qiao. An adaptive-pso-based self-organizing rbf neural network. *IEEE transactions on neural networks and learning systems*, 29(1):104–117, 2016.

[35] Hong-Gui Han, Wei Lu, Ying Hou, and Jun-Fei Qiao. An adaptive-pso-based self-organizing rbf neural network. *IEEE transactions on neural networks and learning systems*, 29(1):104–117, 2016.

[36] Honggui Han, Xiaolong Wu, Lu Zhang, Yu Tian, and Junfei Qiao. Self-organizing rbf neural network using an adaptive gradient multiobjective particle swarm optimization. *IEEE transactions on cybernetics*, 49(1):69–82, 2017.

[37] Sandhya Harikumar and PV Surya. K-medoid clustering for heterogeneous datasets. *Procedia Computer Science*, 70:226–237, 2015.

[38] Elena Hernández-Pereira, Juan A Suárez-Romero, Oscar Fontenla-Romero, and Amparo Alonso-Betanzos. Conversion methods for symbolic features: A comparison applied to an intrusion detection problem. *Expert Systems with Applications*, 36(7):10612–10617, 2009.

[39] Michele Hibon and Theodoros Evgeniou. To combine or not to combine: selecting among forecasts and their combinations. *International journal of forecasting*, 21(1):15–24, 2005.

[40] Feitao Huang, Junhong Chen, Zehang Lin, Peipei Kang, and Zhenguo Yang. Random forest exploiting post-related and user-related features for social media popularity prediction. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 2013–2017, 2018.

[41] Guang-Bin Huang, Paramasivan Saratchandran, and Narasimhan Sundararajan. An efficient sequential learning algorithm for growing and pruning rbf (gap-rbf) networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(6):2284–2292, 2004.

[42] Zhexue Huang. Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining,(PAKDD)*, pages 21–34. Citeseer, 1997.

[43] Monalisa Jena and Satchidananda Dehuri. Decisiontree for classification and regression: A state-of-the art review. *Informatica*, 44(4), 2020.

[44] Siddharth Jhaveri, Ishan Khedkar, Yash Kantharia, and Shree Jaswal. Success prediction using random forest, catboost, xgboost and adaboost for kickstarter

campaigns. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1170–1173. IEEE, 2019.

[45] Songlei Jian, Guansong Pang, Longbing Cao, Kai Lu, and Hang Gao. Cure: Flexible categorical data representation by hierarchical coupling learning. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):853–866, 2018.

[46] Rong Jin and Huan Liu. A novel approach to model generation for heterogeneous data classification. In *IJCAI*, volume 5, pages 746–751, 2005.

[47] Zhao Jing, Jianqiao Chen, and Xu Li. Rbf-ga: An adaptive radial basis function metamodeling with genetic algorithm for structural reliability analysis. *Reliability Engineering & System Safety*, 189:42–57, 2019.

[48] Monika Kalra and Niranjan Lal. Data mining of heterogeneous data with research challenges. In *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, pages 1–6. IEEE, 2016.

[49] Wladyslaw Kaminski and Pawel Strumillo. Kernel orthonormalization in radial basis function neural networks. *IEEE Transactions on Neural Networks*, 8(5):1177–1183, 1997.

[50] Peipei Kang, Zehang Lin, Shaohua Teng, Guipeng Zhang, Lingni Guo, and Wei Zhang. Catboost-based framework with additional user information for social media popularity prediction. In *Proceedings of the 27th ACM international conference on multimedia*, pages 2677–2681, 2019.

[51] Christoph Kern, Thomas Klausch, and Frauke Kreuter. Tree-based machine learning methods for survey research. In *Survey research methods*, volume 13, page 73. NIH Public Access, 2019.

[52] Aditya Khosla, Atish Das Sarma, and Raffay Hamid. What makes an image popular? In *Proceedings of the 23rd international conference on World wide web*, pages 867–876, 2014.

[53] Kyoungok Kim and Jung-sik Hong. A hybrid decision tree algorithm for mixed numeric and categorical data in regression analysis. *Pattern Recognition Letters*, 98:39–45, 2017.

[54] Won Kim and Jungyun Seo. Classifying schematic and data heterogeneity in multidatabase systems. *Computer*, 24(12):12–18, 1991.

[55] Kaitlin Kirasich, Trace Smith, and Bivin Sadler. Random forest vs logistic regression: binary classification for heterogeneous datasets. *SMU Data Science Review*, 1(3):9, 2018.

[56] Sotiris Kotsiantis and Dimitris Kanellopoulos. Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32(1):47–58, 2006.

[57] Miroslav Kubat. Decision trees can initialize radial-basis function networks. *IEEE Transactions on Neural Networks*, 9(5):813–821, 1998.

[58] Akshi Kumar, Vikrant Dabas, and Parul Hooda. Text classification algorithms for mining unstructured data: a swot analysis. *International Journal of Information Technology*, 12(4):1159–1169, 2020.

[59] Ludmila I Kuncheva. Initializing of an rbf network by a genetic algorithm. *Neurocomputing*, 14(3):273–288, 1997.

[60] Xin Lai, Yihong Zhang, and Wei Zhang. Hyfea: Winning solution to social media popularity prediction for multimedia grand challenge 2020. In *Proceedings*

of the 28th ACM International Conference on Multimedia, pages 4565–4569, 2020.

[61] Liuwu Li, Sihong Huang, Ziliang He, and Wenyin Liu. An effective text-based characterization combined with numerical features for social media headline prediction. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 2003–2007, 2018.

[62] Qiude Li, Qingyu Xiong, Shengfen Ji, Min Gao, Yang Yu, and Chao Wu. Multi-view heterogeneous fusion and embedding for categorical attributes on mixed data. *Soft Computing*, 24(14):10843–10863, 2020.

[63] Qiude Li, Qingyu Xiong, Shengfen Ji, Junhao Wen, Min Gao, Yang Yu, and Rui Xu. Using fine-tuned conditional probabilities for data transformation of nominal attributes. *Pattern Recognition Letters*, 128:107–114, 2019.

[64] Qiude Li, Qingyu Xiong, Shengfen Ji, Yang Yu, Chao Wu, and Hualing Yi. A method for mixed data classification base on rbf-elm network. *Neurocomputing*, 431:7–22, 2021.

[65] Haishan Liu and Dejing Dou. An exploration of understanding heterogeneity through data mining. In *Proceedings of KDD 2008 Workshop on Mining Multiple Information Sources*, pages 18–25, 2008.

[66] Huan Liu, Farhad Hussain, Chew Lim Tan, and Manoranjan Dash. Discretization: An enabling technique. *Data mining and knowledge discovery*, 6(4):393–423, 2002.

[67] David Lowe. Adaptive radial basis function nonlinearities, and the problem of generalisation. In *1989 First IEE International Conference on Artificial Neural Networks,(Conf. Publ. No. 313)*, pages 171–175. IET, 1989.

[68] Jinna Lv, Wu Liu, Meng Zhang, He Gong, Bin Wu, and Huadong Ma. Multi-feature fusion for predicting social media popularity. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1883–1888, 2017.

[69] K. Z. Mao. Rbf neural network center selection based on fisher ratio class separability measure. *IEEE Transactions on Neural Networks*, 13(5):1211–1217, 2002.

[70] Richard V McCarthy, Mary M McCarthy, Wendy Ceccucci, and Leila Halawi. Predictive models using decision trees. In *Applying Predictive Analytics*, pages 123–144. Springer, 2019.

[71] Daniele Micci-Barreca. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD Explorations Newsletter*, 3(1):27–32, 2001.

[72] John Moody and Christian J Darken. Fast learning in networks of locally-tuned processing units. *Neural computation*, 1(2):281–294, 1989.

[73] Adel Najafi-Marghmaleki, Afshin Tatar, Ali Barati-Harooni, Mohammad-Javad Choobineh, and Amir H Mohammadi. Ga-rbf model for prediction of dew point pressure in gas condensate reservoirs. *Journal of Molecular Liquids*, 223:979–986, 2016.

[74] John Neter, Michael H Kutner, Christopher J Nachtsheim, William Wasserman, et al. Applied linear statistical models. 1996.

[75] Byran O'Hora, Jerome Perera, and Anthony Brabazon. Designing radial basis function networks for classification using differential evolution. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 2932–2937. IEEE, 2006.

[76] K Okamato, Seiichi Ozawa, and Shigeo Abe. A fast incremental learning algorithm of rbf networks with long-term memory. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 1, pages 102–107. IEEE, 2003.

[77] Witold Pedrycz. Conditional fuzzy clustering in the design of radial basis function neural networks. *IEEE transactions on neural networks*, 9(4):601–612, 1998.

[78] Witold Pedrycz, Ho-Sung Park, and Sung-Kwun Oh. A granular-oriented development of functional radial basis function neural networks. *Neurocomputing*, 72(1-3):420–435, 2008.

[79] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[80] John Platt. A resource-allocating network for function interpolation. *Neural computation*, 3(2):213–225, 1991.

[81] János Podani. Extending gower's general coefficient of similarity to ordinal characters. *Taxon*, 48(2):331–340, 1999.

[82] Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[83] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *arXiv preprint arXiv:1706.09516*, 2017.

[84] Junfei Qiao, Wei Li, Xiao-Jun Zeng, and Honggui Han. Identification of fuzzy neural networks by forward recursive input-output clustering and accurate similarity analysis. *Applied soft computing*, 49:524–543, 2016.

[85] Sergio Ramirez-Gallego, Salvador Garcia, Hector Mourino-Talin, David Martinez-Rego, Verónica Bolón-Canedo, Amparo Alonso-Betanzos, Jose Manuel Benitez, and Francisco Herrera. Distributed entropy minimization discretizer for big data analysis under apache spark. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 2, pages 33–40. IEEE, 2015.

[86] Sergio Ramírez-Gallego, Salvador García, Héctor Mouriño-Talín, David Martínez-Rego, Verónica Bolón-Canedo, Amparo Alonso-Betanzos, José Manuel Benítez, and Francisco Herrera. Data discretization: taxonomy and big data challenge. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(1):5–21, 2016.

[87] Paulo Angelo Alves Resende and André Costa Drummond. A survey of random forest based methods for intrusion detection systems. *ACM Computing Surveys (CSUR)*, 51(3):1–36, 2018.

[88] Christoffer Riis, Damian Konrad Kowalczyk, and Lars Kai Hansen. On the limits to multi-modal popularity prediction on instagram-a new robust, efficient and explainable baseline. *arXiv preprint arXiv:2004.12482*, 2020.

[89] Gilbert Ritschard. Chaid and earlier supervised tree methods. *Contemporary issues in exploratory data mining in the behavioral sciences*, pages 48–74, 2013.

[90] Leszek Rutkowski, Maciej Jaworski, Lena Pietruczuk, and Piotr Duda. Decision trees for mining data streams based on the gaussian approximation. *IEEE Transactions on Knowledge and Data Engineering*, 26(1):108–119, 2013.

[91] Haralambos Sarimveis, Alex Alexandridis, George Tsekouras, and George Bafas. A fast and efficient algorithm for training radial basis function neural networks based on a fuzzy partition of the input space. *Industrial & engineering chemistry research*, 41(4):751–759, 2002.

[92] IP Schagen. Sequential exploration of unknown multi-dimensional functions as an aid to optimization. *IMA Journal of Numerical Analysis*, 4(3):337–347, 1984.

[93] Angelo Silverio, Pierpaolo Cavallo, Roberta De Rosa, and Gennaro Galasso. Big health data and cardiovascular diseases: a challenge for research, an opportunity for clinical care. *Frontiers in medicine*, 6:36, 2019.

[94] Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, and Vishanth Weerakkody. Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70:263–286, 2017.

[95] Donald F Specht. Probabilistic neural networks. *Neural networks*, 3(1):109–118, 1990.

[96] Antonino Staiano, Roberto Tagliaferri, and Witold Pedrycz. Improving rbf networks performance in regression tasks by means of a supervised fuzzy clustering. *Neurocomputing*, 69(13-15):1570–1581, 2006.

[97] Michio Sugeno and Takahiro Yasukawa. A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on fuzzy systems*, 1(1):7–31, 1993.

[98] R Tagliaferri, A Staiano, and D Scala. A supervised fuzzy clustering for radial basis function neural networks training. In *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, volume 3, pages 1804–1809. IEEE, 2001.

[99] David Taniar and Laura Irina Rusu. *Strategic Advancements in Utilizing Data Mining and Warehousing Technologies: New Concepts and Developments: New Concepts and Developments*. IGI Global, 2009.

[100] George E Tsekouras and John Tsimikas. On training rbf neural networks using input-output fuzzy clustering and particle swarm optimization. *Fuzzy Sets and Systems*, 221:65–89, 2013.

[101] Ioannis G Tsoulos, Nikolaos Anastasopoulos, Georgios Ntritsos, and Alexandros Tzallas. Train rbf networks with a hybrid genetic algorithm. *Evolutionary Intelligence*, pages 1–7, 2021.

[102] Zekeriya Uykan, Cuneyt Guzelis, M Ertugrul Çelebi, and Heikki N Koivo. Analysis of input-output clustering for determining centers of rbfn. *IEEE transactions on neural networks*, 11(4):851–858, 2000.

[103] Gancho Vachkov and Alok Sharma. Growing radial basis function network models. In *Asia-Pacific World Congress on Computer Science and Engineering*, pages 1–6. IEEE, 2014.

[104] Gancho Vachkov, Valentin Stoyanov, and Nikolinka Christova. Incremental rbf network models for nonlinear approximation and classification. In *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2015.

[105] Julio J Valdes. Extreme learning machines with heterogeneous data types. *Neurocomputing*, 277:38–52, 2018.

[106] S Elanayar Vt and Yung C Shin. Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems. *IEEE transactions on neural networks*, 5(4):594–603, 1994.

[107] Di Wang, Xiao Jun Zeng, and John A. Keane. A clustering algorithm for radial basis function neural network initialization. *Neurocomputing*, 77(1):144–155, 2012.

[108] Di Wang, Xiao-Jun Zeng, and John A Keane. A clustering algorithm for radial basis function neural network initialization. *Neurocomputing*, 77(1):144–155, 2012.

[109] Kai Wang, Penghui Wang, Xin Chen, Qiushi Huang, Zhendong Mao, and Yongdong Zhang. A feature generalization framework for social media popularity prediction. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4570–4574, 2020.

[110] Bruce A Whitehead. Genetic evolution of radial basis function coverage using orthogonal niches. *IEEE Transactions on Neural Networks*, 7(6):1525–1528, 1996.

[111] D Randall Wilson and Tony R Martinez. Improved heterogeneous distance functions. *Journal of artificial intelligence research*, 6:1–34, 1997.

[112] Andrew KC Wong and David KY Chiu. Synthesizing statistical knowledge from incomplete mixed-mode data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):796–805, 1987.

[113] Bo Wu, Wen-Huang Cheng, Peiye Liu, Bei Liu, Zhaoyang Zeng, and Jiebo Luo. Smp challenge: An overview of social media prediction challenge 2019. In *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.

[114] Ming Xu, Hao Chen, and Liwei Duan. A combined training algorithm for rbf neural network based on particle swarm optimization and gradient descent. In

2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS), pages 702–706. IEEE, 2020.

[115] Bing Yu and Xingshi He. Training radial basis function networks with differential evolution. In *In Proceedings of IEEE International Conference on Granular Computing*. Citeseer, 2006.

[116] Hao Yu, Tiantian Xie, Stanisław Paszczyñski, and Bogdan M Wilamowski. Advantages of radial basis function networks for dynamic system design. *IEEE Transactions on Industrial Electronics*, 58(12):5438–5450, 2011.

[117] Fan Zhang and Lauren J O'Donnell. Support vector regression. In *Machine Learning*, pages 123–140. Elsevier, 2020.

[118] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[119] Kai Zhang, Qiaojun Wang, Zhengzhang Chen, Ivan Marsic, Vipin Kumar, Guofei Jiang, and Jie Zhang. From categorical to numerical: Multiple transitive distance learning and embedding. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 46–54. SIAM, 2015.

[120] Yiqun Zhang and Yiu-Ming Cheung. Discretizing numerical attributes in decision tree for big data analysis. In *2014 IEEE International Conference on Data Mining Workshop*, pages 1150–1157. IEEE, 2014.

[121] Yunwei Zhang, Chunlin Gong, Hai Fang, Hua Su, Chunna Li, and Andrea Da Ronch. An efficient space division-based width optimization method for rbf network using fuzzy clustering algorithms. *Structural and Multidisciplinary Optimization*, 60(2):461–480, 2019.

[122] QM Zhu and SA Billings. Fast orthogonal identification of nonlinear stochastic models and radial basis function neural networks. *International Journal of Control*, 64(5):871–886, 1996.

[123] Djamel A Zighed, Shusaku Tsumoto, Zbigniew W Ras, and Hakim Hacid. *Mining complex data*, volume 165. Springer, 2008.