How to optimise the bandwidths and the dimension of latent spaces in the KCCA and A-CCA machine learning algorithms for statistical matching purposes?

Auteur : Magermans, Céline
Promoteur(s) : Heuchenne, Cédric
Faculté : HEC-Ecole de gestion de l'Université de Liège
Diplôme : Master en ingénieur de gestion, à finalité spécialisée en Financial Engineering
Année académique : 2023-2024
URI/URL : http://hdl.handle.net/2268.2/21316

HOW TO OPTIMISE THE BANDWIDTHS AND THE DIMENSION OF LATENT SPACES IN THE KCCA AND A-CCA MACHINE LEARNING ALGORITHMS FOR STATISTICAL MATCHING PURPOSES?

Jury:
Supervisor:
Cédric HEUCHENNE
Readers:
Malka GUILLOT
Maren ULM

Master thesis by
Céline MAGERMANS
For a master's degree in business engineering, specialisation in financial engineering
Academic year: 2023/2024

Acknowledgments

First of all, I would like to express my gratitude to HEC Liège for the five years of enriching experience I have had there. It has provided me with a high-quality academic environment that has fostered my professional and personal development.

I would like to thank my family and friends for their unfailing support throughout my studies. Their presence and encouragement have been essential pillars for me.

My warmest thanks go to all the people who reviewed my work and provided me with constructive feedback, enabling me to improve the quality of this thesis.

I am also grateful to the jury for taking the time to assess my master thesis.

Finally, I would like to express my sincere thanks to my supervisor, Mr Heuchenne, for his exceptional guidance throughout the writing of this thesis. His constant availability to answer my questions, his sound advice and his clear explanations were invaluable.

Table of contents

List of figures

List of tables

List of abbreviations

A-CCA: Autoencoder and Canonical Correlation Analysis

AI: Artificial Intelligence

ANN: Artificial Neural Network

CCA: Canonical Correlation Analysis

CVM: Cramer-Von Mises statistic

KCCA: Kernel Canonical Correlation Analysis

RsMSE: Root standardised Mean Squared Error

Super-OM: Super-Organising Map

1 Introduction

1.1 Current context

In the current era of Big Data, information has become ubiquitous, flooding our daily lives through a multitude of sources. Bajaj and Ramteke (2014) define Big Data as "*a large amount of data which requires new technologies and architectures to make possible to extract value from it by capturing and analysis process*." Big Data refers to the profiling of extensive datasets (Fan et al., 2014) and is also known by the 3Vs model: Volume, Variety and Velocity (Chen et al., 2014). Volume denotes the vast quantity of data being produced and available. The diverse array of data types (such as photos, text, and audio) along with their multitude of sources collectively represent the Variety within Big Data (Bajaj & Ramteke, 2014). Velocity describes the speed at which Big Data is collected and processed (Chen et al. 2014).

Big Data originate from a plethora of diverse sources, particularly with the advent of the Internet of Things (Chen et al., 2014). For instance, photos shared on social networks, data from wearable sensors, GPS signals, and website cookies all contribute to the generation of Big Data, potentially containing valuable information (Bajaj & Ramteke, 2014). Consequently, one of the challenges associated with Big Data is its inherent heterogeneity, wherein information stems from varied sources, making aggregation challenging, especially considering that the populations of these sources likely do not align (Fan et al., 2014).

This ground-breaking technology is also opening up unparalleled opportunities for stakeholders worldwide. It enables the extraction of essential information for commercial, medical, scientific, and other applications. For instance, a company could utilise this data to gain deeper insights into the characteristics of its customers and tailor advertising efforts accordingly (Bajaj & Ramteke, 2014).

Despite the abundance of available information, challenges arise when attempting to consolidate data. This study will therefore address the issue of data heterogeneity. As previously discussed, a vast amount of data is accessible and generated on the internet, through surveys, interviews, and by connected devices, resulting in a proliferation of data sources (Van Der Putten et al., 2002). Both private and public organisations leverage this data to gather pertinent information for analysis and to draw conclusions on specific subjects of interest.

Nonetheless, the essential variables needed for a particular study are frequently dispersed across multiple databases. However, conducting a new survey to collect all necessary variables for a comprehensive analysis is not only cost-prohibitive but also time-consuming. Hence, a more pragmatic and efficient solution lies in integrating relevant databases. By merging databases, the research process is streamlined, and all essential variables are readily accessible within a single, consolidated database. This approach not only conserves resources but also facilitates a more cohesive and coherent analysis. This is the reason why integration of diverse data sources has become a critical aspect of contemporary research and analysis across various fields.

Statistical matching, also known as data fusion or synthetical matching (Annoye et al., 2024), is a technique used to merge data from various sources that cannot be perfectly matched. This method is

particularly valuable when dealing with disparate datasets that cover related aspects but are not inherently linked. Moreover, the improvement of algorithms, particularly through the integration of machine learning techniques, plays a crucial role in enhancing the accuracy and efficiency of statistical matching. Machine learning algorithms can learn and adapt to complex relationships between variables, providing more sophisticated insights into the interplay between the relationship in study.

Three methods are exposed by the paper of Annoye et al. (2024): "*the Kernel Canonical Correlation analysis (KCCA), the Super-Organising Map (Super-OM) and Autoencoders and Canonical Correlation Analysis (A-CCA)*". The first method reduces the dimensionality and uses the kernel trick. This approach also allows detecting matching between texts and images. Then, the Super-OM method uses the ANN (Artificial Neural Network) to have a low dimensionality. Finally, the A-CCA also uses an encoder, the ANN. In other words, the data is compressed by an encoder and then decompressed by a decoder. This process creates low dimensionality. The Canonical Correlation Analysis approach is then applied for statistical matching.

There are also other techniques unrelated to machine learning. These include parametric multivariate distributions, regression, and hot deck. However, it has been proven that machine learning methods provide better results.

1.2 Objective of the thesis

The focus of my master's thesis is to enhance the performance of a machine learning algorithm utilised for statistical matching purposes by the Belgian government. A team from the University of Liège collaborates with the state with the objective to develop algorithms capable of effectively merging multiple databases to fulfil the government's requirements. For instance, if the government aims at assessing the impact of rising petrol prices on the population, it necessitates linking a database containing information on living conditions with another database focusing on mobility patterns. This integration process, known as statistical matching, relies on the application of algorithms.

The team has already developed machine learning techniques for this purpose, but further analyses are required for certain hyperparameters, including the bandwidths and the dimensions of latent spaces in the Kernel Canonical Correlation Analysis (KCCA) and the Autoencoder and Canonical Correlation Analysis (A-CCA) methods. Consequently, this master's thesis aims at optimising these hyperparameters to improve the algorithm's ability to match databases accurately and predict non-common variables more effectively. By achieving this objective, the project team will be able to verify the quality and coherence of their work by analysing performance as these hyperparameters vary, thus ensuring robust and reliable results. Moreover, the Belgian government will gain enhanced capabilities to analyse various relationships and make informed decisions based on comprehensive data analysis.

Therefore, the objective of this thesis is to answer to following question:

"How to optimise the bandwidths and the dimension of latent spaces in the KCCA and A-CCA machine learning algorithms for statistical matching purposes?"

To address this research question, we generated a database in R and followed a Grid Search strategy to optimise the hyperparameters. The performance of the different models was evaluated using the Cramer-Von Mises statistic and the Root standardised Mean Squared Error (RsMSE). The results show

that optimising the bandwidth hyperparameters and the dimension of the latent spaces has an impact on the two-performance metrics of the KCCA and A-CCA algorithms. The relationships between the values of the hyperparameters and the evolution of the two metrics can be established; however, the direction of these relationships differs between the two approaches. Furthermore, the results vary depending on whether we seek to optimise the Cramer-Von Mises statistic or the RsMSE, suggesting that a compromise is necessary. These findings make a significant contribution to the field of machine learning, offering new perspectives for the optimisation of statistical matching algorithms.

This thesis will be structured as follow: in chapter 2, the literature review section will delve into the concept of statistical matching from various perspectives, exploring different techniques before transitioning to the discussion on machine learning techniques. Subsequently, the chapter 3 on the methodology used will outline the approaches employed in this study, focusing on the two specific techniques under examination and their hyperparameters to be optimised. Then, the results will be presented in chapter 4 and discussed, analysed, and interpreted in chapter 5. Finally, the chapter 6, “Conclusions”, will provide a summary of the research, mention its limitations, and suggest some avenues for future research in the field.

2 Literature review

In this literature review, we will begin with a presentation of statistical matching, briefly covering its historical development, applications, and the advantages and challenges. We will then go into more detail on this concept, also known as data fusion, providing a detailed explanation of how it works, and outlining some of the techniques associated with this field. Next, we will introduce the concept of machine learning by discussing a few key concepts and highlighting the link between machine learning and statistical matching. Finally, we will detail the machine learning techniques used for data fusion, two of which will have some hyperparameters optimised in order to meet the objectives of this thesis.

2.1 Introduction to statistical matching

As already explained in the introduction, the volume of data has been increasing exponentially and it becomes increasingly important to have the knowledge to handle them and to take use of them in a beneficial way, for companies, states, organisations or whatever. However, the topic to analyse requires sometimes (or better said, almost always) data from different sources (Radner et al., 1980). Indeed, not only is the number of available data growing exponentially, but the number of sources is rising even as well (Van Der Putten et al., 2002).

A variety of potential solutions may be put forth to address this issue. One potential solution would be to create a new survey in order to construct a database comprising all the requisite variables for the study. A second potential solution would be to employ a variety of imputation techniques to estimate the missing values. Ultimately, an alternative solution, which will be adopted for this study, is the combination of the disparate data sources, a process referred to as data fusion, statistical matching, or synthetic matching (Radner et al., 1980).

In this literature, the term statistical matching will be defined by addressing various points such as its history and benefits. Next, several traditional statistical matching techniques will be presented. We will then briefly describe what machine learning is before going on describing the related statistical matching methods.

2.1.1 Difference between statistical and exact matching

Before anything else, it is crucial to distinguish between statistical and exact matching. On the one hand, exact matching is defined by Radner et al. (1980) as "*a match in which the linkage of data for the same unit (e.g., person) from the different files is sought*". This implies that the units in the sources are identical (Eurostat, 2013). On the other hand, statistical matching does not aim to identify an exact match between the two sources, given the inherent difficulty in doing so (Eurostat, 2013; Radner et al., 1980). This is because the data sets may contain similar but not identical observations. The primary distinction lies in the degree of overlap between the two populations (Eurostat, 2013).

The implementation of exact matching or record linkage is a highly complex process, largely due to the inherent unlikeliness of two databases containing an identical population. This is particularly true when the sources of the data are comprised of very large populations, where the information has been collected independently (Annoye et al., 2024). Furthermore, record linkage necessitates the presence of variables that serve to identify the units within the population, including national identifiers, names,

and addresses (Annoye et al., 2024; Radner et al., 1980). However, this is not always feasible when the data must remain anonymous and private (Data Fusion, 2023) and comply with GDPR regulations.

2.1.2 History

2.1.2.1 The emergence of statistical matching

Obviously, exact matching existed before statistical matching, as it required less statistical effort and almost only had to rely on the variable used for identification in the sources. Statistical matching came about in response to the shortcomings and limitations of record linkage.

In the mid-1960s, the Bureau of Economic Analysis of the United States Department of Commerce wanted to conduct a study of the characteristics of individuals in its population in relation to the taxes levied on them (D'Orazio et al., 2006). Their aim was to obtain a database with socio-demographic information. However, there was no database containing all the necessary information. Meanwhile, the variables were present in two diverse sources. Therefore, they decided to merge the United States Tax File dating from 1966 with the Survey of Economic Opportunities carried out in 1967 (Eurostat, 2013). At the same time, statistical matching was also used by the Brookings Institution to obtain a database of all taxpayers (Radner et al., 1980). During the 1970s, other applications were made concerning income and other socio-demographic subjects (Ruggles & Ruggles, 1974).

Subsequently, statistical matching became increasingly visible within the media industry (Rässler, 2002). In fact, this technique played a key role in media targeting during that era and continues to be influential today. Through the analysis and consolidation of survey data on consumption, household preferences, and behaviour, various advertising strategies could be formulated and implemented. Statistical matching facilitated the optimisation of advertising budgets by directing efforts towards the most profitable consumers and delivering tailored content to them (O'Brien, 1991; Rässler, 2004; Van Der Putten et al., 2002).

Thus, statistical matching has seen significant development in the spheres of politics and economics, as well as playing a crucial role in the media (Van Der Putten et al., 2002). This technique has also paved the way for numerous studies. Following extensive research in the areas mentioned above, a wide range of socio-demographic and psychological studies using data fusion have been undertaken (Ruggles & Ruggles, 1974). For example, Gavin (1985) found correlations between socio-demographic characteristics and individual health.

2.1.2.2 Current use of statistical matching in various fields

As time progresses, this technique is gaining increasing popularity across all sectors. As the quantity of information gathered with the advent of the internet continues to grow, organisations are seeking methods of utilising this data for a variety of purposes, including commercial, scientific, and other applications. In addition, technologies are developing very rapidly and the emergence of the Internet of things (Chen et al., 2014) reinforces this rapid growth in information and its sources (Van Der Putten et al., 2002).

Since its inception, statistical matching has been used mainly in research, in economics and politics. Indeed, this method makes it possible to explore various scenarios, a requirement which is increasingly demanding in today's context (Eurostat, 2013). Many European countries are working to understand

the relationship between household expenditure and income, but this information is usually derived from two separate databases that need to be merged (Annoye et al., 2024). Other examples of recent studies on quality of life and the labour market are developed in the Eurostat report (2013).

With the advent of rapid technological advances, statistical matching is set to become increasingly important in a variety of sectors, including automotive, healthcare, and smart cities. The field of autonomous vehicles provides an illustrative example of the importance of data fusion from a range of sensors in determining the optimal trajectory and ensuring safe and efficient navigation. Furthermore, in healthcare, the comprehensive analysis of large patient datasets promises to elucidate the complex underlying factors contributing to various diseases, potentially revolutionising methods of diagnosis and treatment. Such advances underline the transformative potential of statistical matching as a fundamental tool for driving innovation and addressing complex challenges in multifaceted industries (Data Fusion, 2023).

As can be observed, the utilisation of statistical matching techniques is becoming an increasingly crucial aspect, with applications across a range of sectors. This underscores the value of conducting a comprehensive examination of this technique to enhance its efficacy.

2.1.3 Benefits and challenges of the statistical matching

Statistical matching techniques offer a wide array of advantages, ranging from their ability to integrate disparate datasets and enhance data completeness to facilitating robust analyses and enabling informed decision-making. However, alongside these benefits, they also introduce a host of complexities and obstacles that must be navigated.

2.1.3.1 Challenges of the statistical matching

A first challenge linked to the growing amount of data is the management of gigantic volumes of information (Data Fusion, 2023). It is evident that the undertaking of such an extensive data analysis requires a combination of necessary skills and sufficiently powerful software. To illustrate, a sample of the Belgian population may comprise up to 10,000 individuals and encompass twenty or more variables. It should be noted that this database must be merged with another of potentially similar dimensions. In the absence of expertise in data management, the processing of data becomes challenging. In such instances, the utilisation of software assumes considerable importance, particularly for the purpose of conducting in-depth analysis.

Linked to this, another significant challenge arises in managing the intricate calculations inherent in statistical matching techniques (Data Fusion, 2023). As elucidated above, data fusion methods are frequently applied to vast databases, necessitating the utilisation of rather sophisticated calculations. Consequently, it becomes imperative to effectively handle this complexity and utilise software robust enough to seamlessly support the entirety of the matching process.

Moreover, in the context of statistical matching, it is crucial to uphold data confidentiality (Data Fusion, 2023; Radner et al., 1980) and adhere to GDPR regulations (Gessendorfer et al., 2018). Consequently, meticulous attention must be paid to these regulations to ensure compliance. It is imperative to verify that data and information are anonymised, safeguarding private information and preserving anonymity, particularly pertinent when analysing sensitive datasets such as financial or medical data.

Ultimately, statistical matching can be vulnerable to biases throughout the data alignment procedure. A significant illustration of this is the risk posed by discrepancies in demographic characteristics among population subsets within the source datasets, which has the potential to introduce bias into the outcomes of statistical matching. These biases typically arise from inconsistencies (Gessendorfer et al., 2018). This challenge is intricately linked with the task of ensuring the quality of the amalgamated data, which entails managing missing, inconsistent, or erroneous data, all of which could significantly impact the matching outcomes (Data Fusion, 2023). Addressing the issue of heterogeneity between sources becomes imperative, encompassing considerations of data format, structure, and content (Ruggles & Ruggles, 1974). Consequently, robust methodologies are required to identify, quantify, and mitigate biases, ensuring the reliability and validity of the matched data.

2.1.3.2 Benefits of data fusion

Data fusion presents a plethora of notable benefits, among which is the invaluable capacity of statistical matching to merge two distinct data sources into a unified, comprehensive database, encompassing all pertinent information for the study (Rässler, 2004; Van Der Putten et al., 2002). This consolidation streamlines the execution of thorough analyses using pre-existing data, thereby maximising the utilisation of collected information (Eurostat, 2013). With access to a comprehensive database, information becomes readily accessible and decision-making processes are enhanced (Data Fusion, 2023). This seamless integration of data not only facilitates more efficient decision-making but also fosters a deeper understanding of complex phenomena through comprehensive data exploration and analysis.

Moreover, merging several sources circumvents the necessity of conducting a new survey to gather requisite information for the desired study, even if said information is dispersed across different sources. Executing a new survey entails significant investments of time and resources, which can be conserved through the application of statistical matching techniques (D'Orazio et al., 2006; Radner et al., 1980). Data fusion facilitates the reduction of the number of questions and respondents in a questionnaire, thereby enhancing data quality (Van Der Putten, 2002). Longer surveys often result in increased data incompleteness and decreased respondent accuracy (D'Orazio et al., 2006). Thus, by leveraging data fusion methods, researchers can effectively optimise resource allocation and improve the overall efficiency of data collection processes, while simultaneously enhancing the quality and reliability of collected data.

Another notable advantage of statistical matching is and will be its utility as a 'what-if' measure, allowing researchers to explore diverse hypothetical scenarios. Through this approach, researchers can simulate alternative outcomes by manipulating input data within the matching model. This affords significant flexibility in evaluating the potential ramifications of various interventions, policies, or decisions based on existing data (Rässler, 2004). It is precisely due to this advantageous capability that Belgian authorities are endeavouring to optimise this algorithm. By harnessing the power of statistical matching for scenario analysis, policymakers and decision-makers can make more informed choices, anticipate potential outcomes, and devise strategies that align with desired objectives.

2.2 What is the statistical matching?

Statistical matching, alternatively referred to as data fusion or synthetic matching, stands as a data integration technique (D'Orazio et al., 2006) characterised by the amalgamation of data originating from disparate sources that may not inherently share identical variables (Rässler, 2004). The overarching objective of this methodology is to construct a more comprehensive and insightful dataset by aggregating information from diverse origins. This process enables analysts to synthesise a unified dataset that encapsulates a broader spectrum of information, thereby enhancing the richness and utility of the resultant data for subsequent analyses and decision-making endeavours.

In statistical matching, a crucial differentiation exists between the micro and macro approaches. At the micro level, the primary objective revolves around constructing a synthetic database derived from the available sources. This involves matching and comparing individual and complete data for each unit, thereby facilitating the exploration of relationships between individual units (D'Orazio et al., 2006). The micro approach is centred on analysing the specific characteristics of each unit (Eurostat, 2013) and investigating the interactions between them. By delving into the intricacies of individual units and their interplay, analysts can gain valuable insights into the nuanced dynamics within the dataset, allowing for detailed examinations of various phenomena at a granular level.

In the macro approach, the traditional merging of databases characteristic of statistical matching may not necessarily occur. Instead, this approach centres on utilising the data in its original form to estimate global characteristics of the variables of interest that are not observed together in each dataset (D'Orazio et al., 2006). This may entail leveraging the source files to estimate joint distribution functions, marginal functions, or correlation matrices of the variables of interest (Eurostat, 2013).

In order to illustrate the concept more clearly, it is helpful to consider two databases, A and B. Both databases share common variables, referred to as X, but also contain distinct variables. Variables present in database A but not in B are denoted as Y, while variables found in database B but not in A are labelled as Z. The objective of statistical matching is to merge these two datasets to create a single comprehensive database containing variables X, Y, and Z, which is called a "synthetical data set" (Annoye et al., 2024; D'Orazio et al., 2006; Eurostat, 2013; Gessendorfer et al., 2018).

Obtaining the synthetic dataset is not a straightforward task due to the necessity to estimate the non-common variables, presenting a challenge akin to dealing with missing data. The aim of statistical matching is to solve the problem of missing data in one source by using another data source. Several imputation techniques exist for missing data, such as replacement by the mean or median, imputation by model using regressions (Kim & Shao, 2013; Van Buuren, 2018), or imputation of data using the maximum likelihood method (Anderson, 1974). However, a difference between the two notions lies in the fact that for missing data, the objective is to fill gaps caused by non-responses in surveys and to allow statistical analyses to be performed, whereas for statistical matching, the objective is to find credible data for each observation in the recipient dataset using information from the donor dataset.

Before initiating the statistical matching process, D'Orazio et al. (2006) detail several steps to be followed in order to assess the coherence between the two databases and to examine the feasibility of statistical matching for the two sources to be matched. They suggest analysing the degree of

harmonisation and reconciliation between the two sources by addressing eight key points, also cited in the Eurostat report (2013):

1. Harmonisation on the definition of units
2. Harmonisation of reference period
3. Completion of population
4. Harmonisation of variables
5. Harmonisation of classification
6. Adjustment for measurement errors
7. Adjustment for missing data
8. Derivation variables

For further elaboration on the specific points to be examined, readers can refer to the comprehensive analysis provided by D'Orazio et al. (2006) and the Eurostat report (2013). Another critical consideration lies in evaluating the effectiveness and influence of common variables in predicting or amalgamating outcomes. This involves assessing the explanatory power of shared variables, which play a pivotal role in statistical matching processes. By examining the correlation and predictive capability of these common variables, researchers can gauge their contribution to the accuracy and reliability of outcome predictions or data integration (D'Orazio et al., 2006; Eurostat, 2013).

In the case of statistical matching, if the aim is to incorporate all variables from dataset A and augment it with attributes Z from dataset B, A acts as the "recipient" while B serves as the "donor." This can be explained by the fact that the information required to estimate variables $\hat{Z}$ in A is derived from B (D'Orazio et al., 2006; Gessendorfer et al., 2018; Van Der Putten et al., 2002). To mitigate bias during statistical matching processes, it is crucial for the individuals or observations in the donor dataset to align with those in the recipient dataset (Van Der Putten et al., 2002). This principle is illustrated graphically below, employing socio-demographic and mobility surveys as examples:

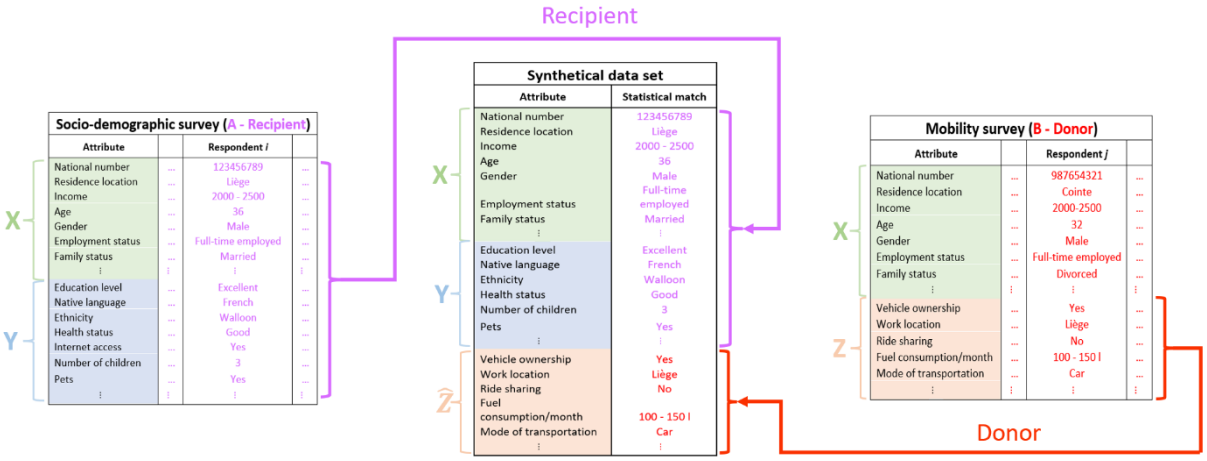


***Figure 1**: Illustration of the statistical matching principle*

As Radner et al. (1980) emphasise in their report, "*in a statistical match, the linkage of data for similar units rather than for the same unit is acceptable and expected*." This assertion underscores the objective of finding similar or matched observations in disparate data sources, rather than striving for exact matches for each individual observation, as such overlap is typically non-existent (Eurostat, 2013).

By adopting this approach, the utilisation of available data is maximised, thereby furnishing researchers with richer and more accurate information about the phenomenon under study.

2.2.1 Traditional statistical matching techniques

Since statistical matching was introduced, various methods have been developed to estimate the $\hat{Z}$ variables in the recipient. In this section, only the three best-known traditional approaches will be discussed, before moving on to machine learning techniques in the next section. Other techniques such as Bayesian and multiple imputations methods have been proposed and details about them are given in D'Orazio et al. (2006), Rässler (2002), Rubin (1987).

Traditional statistical matching methods include well-known approaches such as regression, the use of parametric multivariate distributions and the hot-deck method, as discussed by Aluja-Banet and co-authors in their 2007 paper. These techniques are widely used to harmonise data from diverse sources, allowing appropriate comparison and combination of the variables of interest. Obviously, other methods exist and so do combinations of several techniques (Annoye et al., 2024).

2.2.1.1.1 Regressions

In the context of statistical matching, the regression method is used to establish the relationship between the variables X and Z from the donor group (B). This step is crucial because it enables this relationship to be modelled so that the values of the $\hat{Z}$ variables in the recipient group can be inferred. In this way, the regression model $f(Z|X)$ is constructed from the data present in the donor in order to predict the missing variables $(\hat{Z})$ in the recipient (Aluja-Banet et al., 2007). In this case, the X variables are the independent variables, and the Z variables are the dependent variables (Ruggles and Ruggles, 1974). This regression model is defined as follows:

$$f(Z|X) = r(X) + \varepsilon$$

Where $r(X)$ represents the modelled relationship between the variables X and Z and ε is a random error term capturing the residual differences between the actual values for variables Z and predictions of variables $\hat{Z}$.

Then, once the relationship between X and Z has been established, this regression model is applied to the data in the recipient group (A) in order to predict the values of $\hat{Z}$ (Aluja-Banet et al., 2007). In other words, for each observation in the recipient dataset, the values of the X variables are used as inputs to the regression model to predict the values of the $\hat{Z}$ variables.

As Ruggles and Ruggles (1974) point out in their book, the reliability of this method depends essentially on the robustness of the relationship between the X and Z variables. It is crucial to note that in order to obtain accurate results, the variable to be imputed must be strongly associated with and strongly explained by the common X variables. In other words, the strength of the relationship between X and Z is crucial to the accuracy of the predictions.

In addition, it is worth highlighting the diversity of regression approaches available, each adapted to specific situations depending on the context as well as the underlying assumptions. Among these methods are linear or non-linear, parametric, or non-parametric regressions, as and other specific models that vary according to analytical needs. Examples include logistic to predict an outcome with

values 1 or 0 or linear regression for predicting continuous variables (Van Der Putten et al., 2002), polynomial regression and Lasso regression.

2.2.1.1.2 Parametric multivariate distribution

The objective of the parametric multivariate distribution is to estimate the missing Z variables in dataset A using the information provided by dataset B (Annoye et al., 2024). This approach is based on the joint modelling of the distributions of (X, Z) where X are the common variables taken in the survey B, the donor dataset.

The initial step involves modelling the multivariate distribution to jointly represent the common variables X sourced from dataset B, as well as the specific variables Z originating also from the donor. This approach assumed that these variables follow a common distribution, parameterised by θ. In essence, the observed and missing variables are postulated to be sampled from the same underlying distribution (Aluja-Banet et al., 2007). The parameters of this distribution are subsequently estimated using the data available in dataset B, employing techniques such as the maximum likelihood method or other parametric estimation methods (D'Orazio et al., 2006). Finally, information from dataset B about the Z variables is used to impute missing values in dataset A. This obviously requires the use of the estimated parametric distribution.

Mathematically, the parametric distribution is described as follows:

$$f(X, Z|\theta)$$

Where X represents the common variables in B, Z represents the variables specific to survey B that should be imputed in set A, and θ are the parameters of the multivariate distribution.

This distribution can be decomposed based on the fundamental assumption of conditional independence (D'Orazio et al., 2006) between the random variables X and Z, conditional on the parameter θ (Aluja-Banet et al., 2007):

$$f(X, Z|\theta) = f\left(Z|X, \theta_{Z|X}\right) f(X, \theta_X)$$

This decomposition of the distribution allows the parameters $\theta_{Z|X}$ and marginal θ_X to be estimated from the data that is already available in both sets. Finally, these parameters can be used to impute missing values of $\hat{Z}$ in set A (Aluja-Banet et al., 2007).

2.2.1.1.3 Hot deck

The hot-deck method is widely regarded as the most popular technique, as noted by various experts (Aluja-Banet et al., 2007; Eurostat (2013); Gessendorfer et al., 2018). The objective of this non-parametric method is to identify, for each observation in dataset A, a corresponding observation in dataset B that is similar in terms of their common variables X. Subsequently, the variables Z from the matched observation in B are transferred to the recipient dataset A (Eurostat, 2013). Thus, one notable advantage of this approach is its independence from assumptions regarding the distribution or the relationship between the X and Z data because, as Aluja-Banet et al. (2007) state, hot-deck is a « *data-based method* ».

However, in their study, Gessendorfer et al. (2018) shed light on a drawback of the hot-deck method, specifically emphasising the significance of the size of the donor set (B). Indeed, they underscore that a larger donor database enhances the probability of identifying robust similarities between the common variables X of dataset A and dataset B. Consequently, the values borrowed from dataset B to fill in the values of the $\hat{Z}$ variables in dataset A for the variables Z are more likely to be accurate.

Each of these approaches can be constrained or unconstrained, depending on whether an observation from the donor dataset can be used several times to impute values (unconstrained) or only once (constrained). There are several types of hot-deck techniques used for statistical matching, such as the random hot-deck and the distance hot-deck, or also known as the nearest neighbour donor in the constrained case (Aluja-Banet et al., 2007).

On the one hand, the random hot deck simply consists of choosing the donor observation at random from a subset of observations which share similar characteristics to the observation to be imputed in data set A, all by comparing the common variables X. Although this method is relatively simple, it can cause numerous biases due to its randomness. On the other hand, the distance hot-deck or nearest neighbour donor, involves calculating a distance function between the common variables of the two databases in order to determine the donor more accurately (Spaziani et al., 2019). Thus, the observation which minimises this distance will be chosen to associate these values in the recipient. The distance function can be calculated in diverse ways, e.g., Manhattan distance, Euclidean distance (Eurostat, 2013).

Finally, other methods combining the parametric and non-parametric approaches explained above exist and are also widely used for statistical matching purposes (Annoye et al., 2024; Eurostat, 2013).

2.3 Machine learning techniques

2.3.1 Introduction to machine learning

2.3.1.1 Definitions and link with statistical matching

According to Rebala et al. (2019) in their book "*An introduction to Machine Learning*", machine learning is defined as "*a field of computer science that studies algorithms and techniques for automating solutions to complex problems that are hard to program using conventional programming methods*". Meanwhile, Bi et al. (2019) see the concept as "*a branch of computer science that is largely concerned with enabling computers to 'learn' without being directly programmed*". Finally, as stated by D'Orazio et al. (2019), the concept of machine learning "*involves a broad set of techniques based on algorithms that learn from data*".

While machine learning has been present since the early days of computing in the 1950s and 1960s, its progress has significantly accelerated since the onset of the 21st century. This surge in advancement can be attributed to the remarkable growth in computing power and the widespread availability of the internet. These factors have played a pivotal role in facilitating the development and proliferation of machine learning techniques (Rebela et al., 2019). Currently, machine learning methods have become ubiquitous across numerous sectors, notably in marketing, economics, and finance, as they offer numerous advantages (Spaziani et al., 2019).

The concept of machine learning is often confused with that of artificial intelligence (AI) aims to imbue machines with intelligence through a range of approaches. In reality, machine learning is just one of the techniques included in the field of AI (Rebela et al., 2019).

Hence, it is feasible to identify the connection between statistical matching and machine learning, as both have to do with data analysis and prediction. Machine learning offers powerful functions that could be leveraged for data fusion and will improve statistical matching techniques. Algorithms, for example, streamline the process and enable learning directly from the data.

Thus, as defined by these different authors, machine learning makes it possible to take into account more complex relationships than the statistical techniques set out previously in this work (Spaziani et al., 2019). Indeed, machine learning has the power to learn from data by proposing more complex models which are therefore more representative of the situation, which is a major advantage. This also means that accuracy is even higher when the database is large (Rebala et al., 2019). In addition, machine learning makes it possible to manage large volumes of data, which is becoming essential in the age of Big Data and the exponential growth of available data (Bi et al., 2019). Furthermore, this technique does not rely on any hypothesis concerning the distribution of the data or the relationships between variables, but is based directly on the data, which avoids a fairly significant source of bias when these hypotheses are not verified. In this way, machine learning can adapt more easily to the data without any a priori specification.

The advantages outlined above show the importance of including machine learning techniques in the statistical matching process and are the reasons why they will be applied in this work.

2.3.1.2 Fundamentals concepts of machine learning

First and foremost, it is imperative to specify a few fundamental concepts which are essential to a proper understanding of machine learning. The initial distinction can be made between two types of data: labelled data and unlabelled data. Labelled data refers to data for which the target value to be predicted or identified is known, whereas unlabelled data lacks such provided target value (Bi et al., 2019; Rebala et al., 2019).

Next, there are distinct categories of machine learning techniques, four of which will be briefly explained. Firstly, supervised learning consists of receiving a database made up of labelled data and learning from this data using algorithms to define a model. The aim is that when the algorithm receives a new observation that does not exist in the initial database, a relatively accurate prediction can be made based on the characteristics of the previously studied data and the model established (Bi et al., 2019; Rebala et al., 2019). Regression techniques (linear and logistic) and decision trees are examples of methods belonging to supervised learning (Bi et al., 2019).

In unsupervised learning, the second category, the algorithm is provided with an unlabelled dataset. Its objective is to discern underlying patterns or similarities within the data, typically to form clusters or groups based on inherent structures (Bi et al., 2019; Rebala et al., 2019).

Between the two categories mentioned above lies semi-supervised learning, which receives a database composed of a mix of labelled and unlabelled data. Thus, initially, the aim will be to create clusters (link with unsupervised learning) before predicting the label of the unlabelled data (link with supervised

learning) thanks to the labelled data present in the subgroup (Bi et al., 2019; Rebala et al., 2019). The advantage of semi-supervised learning is that it saves time because there is no need to label each observation, but it is still very demanding (Bi et al., 2019).

Finally, reinforcement learning is a machine learning technique in which systems gradually learn to improve at a given task by experimenting with different actions and adjusting their behaviour according to the results obtained and the environment. By exploiting the knowledge gained from these initial attempts, reinforcement learning algorithms gradually optimise their strategies to achieve predefined goals (Bi et al., 2019; Rebala et al., 2019). This category is incredibly useful in dynamic environments where the number of possibilities is immense, as in the game of chess (Rebela et al., 2019).

Moreover, when optimising the hyperparameters in machine learning algorithms, it is crucial to understand the concepts of underfitting and overfitting. These phenomena are directly linked to the bias-variance trade-off, which represents the balance between two sources of error: bias (the error due to oversimplified assumptions in the model) and variance (the sensitivity of the model to fluctuations in the training data).

On the one hand, underfitting occurs when the model is too simple to capture the underlying trends in the training data (in this case, the donor, survey B). This results in inferior performance on both the training data and the test data (survey A), indicating that the model has not learned the essential relationships in the data well. In other words, a model with a high bias and low variance is likely to underlearn and this concept is represented on the left of the figure below. On the other hand, overfitting occurs when the model is too complex and learns not only the underlying trends but also the noise and specific anomalies in the training data. Such a model performs excellently on the training data but fails to generalise to new data. This means that it has low bias, but high variance and the situation is represented on the right below (Jabbar & Khan, 2014).

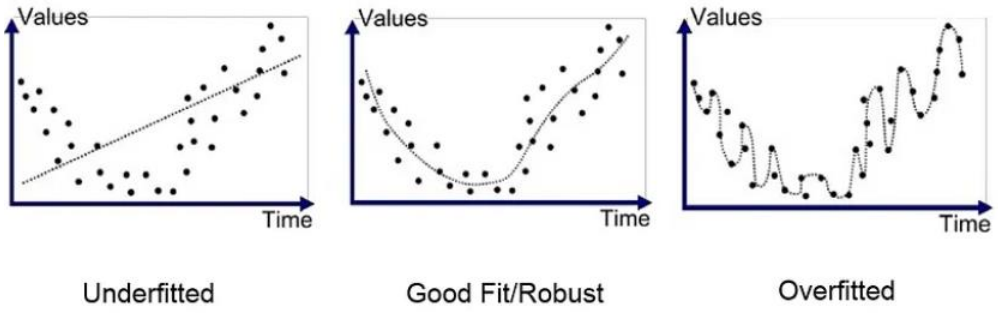


***Figure 2**: Illustration of the concepts of underfitting, good fit and overfitting (Bhande, 2018)*

To avoid these problems, it is essential to properly optimise the bandwidth hyperparameters and the dimension of the latent spaces. A good balance between the complexity of the model and its ability to generalise results in a minimisation of the Root standardised Mean Square Error (RsMSE) on the test dataset, ensuring that the model is both accurate and generalisable. The balance is represented by the following graph:

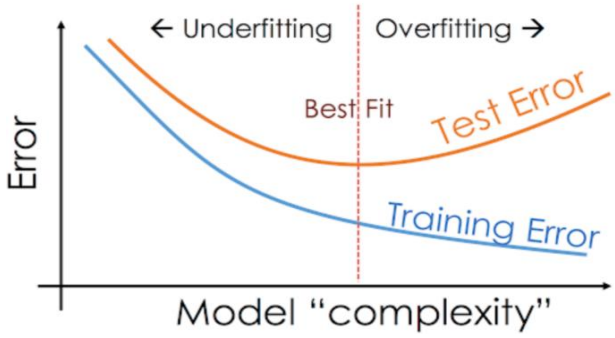


***Figure 3**: Illustration of the bias-variance trade-off showing underfitting, best fit and overfitting (Saxena, 2023)*

In conclusion, the best fit is at the point where the RsMSE is at its minimum, just before rising. On the left side of this point, the model would be underfitted, whereas to the right, it would be overfitted.

2.3.1.3 Well-known machine learning techniques

Briefly, the best-known machine learning techniques are regressions (logistic or linear), decision trees (random forests) and Artificial Neural Networks (ANNs). As explained above, regressions are used to model the relationships between variables on the basis of historical values in order to predict the independent variable. The use of machine learning makes it possible to model more complex relationships than traditional regression techniques.

Next, in the realm of machine learning, a decision tree algorithm operates by posing sequential questions based on available data to make decisions, effectively segmenting it into more homogenous subsets. Constructed recursively, the process stops when the data is suitably classified or upon meeting predefined stopping criteria to prevent overfitting (Rebela et al., 2019). Once established, the decision tree can predict outcomes for new data by applying its series of questions. This methodology is valued for its interpretability, facilitating the identification of significant variables in prediction or classification tasks (Bi et al., 2019; Rebela et al., 2019). Random forests, which consist of ensembles of decision trees, enhance prediction accuracy by aggregating the outputs of multiple trees (Rebela et al., 2019).

Finally, the Artificial Neural Network (ANN), inspired by the human brain's functioning, comprises multiple layers of interconnected neurons. The "input" layer represents independent variables, while the "hidden" layers and the "output" layer represent dependent variables (Bi et al., 2019). Information is transmitted between layers through synaptic weights, initially set randomly and then adjusted via back-propagation to minimise the disparity between predictions and actual values (Bi et al., 2019; Rebela et al., 2019). This adaptive approach enables the network to handle complex models and make predictions on new data after training. However, neural networks lack interpretability, making it challenging to understand the significance of individual variables (Bi et al., 2019).

2.3.2 Machine learning techniques with data fusion purposes

In this section, various machine learning techniques will be discussed in the context of data fusion. An initial approach by Spaziani et al. (2019) will be presented before continuing with the techniques developed in the article by Annoye et al. (2024), which will serve as a basis for this thesis.

2.3.2.1 A two-stage approach

In their paper "*Integration of Survey Data in R Based on Machine Learning*," Spaziani et al. (2019) present a two-step approach using machine learning techniques to combine two databases. The first step involves using machine learning methods to predict missing variables in each of the surveys. Thus, to predict the Y data from survey A, a model (regression or decision tree or another technique) needs to be developed based on the available data X. Then, this model is applied to database B to predict and impute the values of the Y variables. The same process is applied to predict and impute the values of the Z variables from B into A (Spaziani et al., 2019).

In the second phase, the traditional hot-deck technique, explained earlier, is employed to impute the Z variables in A. The predictions of $\hat{Y}$ and $\hat{Z}$ generated via machine learning techniques are used as matching criteria for the hot-deck method (Spaziani et al., 2019).

Although this approach has several advantages over traditional statistical matching techniques, Annoye et al. (2024) point out two limitations. The first one concerns the survey weights, which are not included in the approach outlined (Annoye et al., 2024). Survey weights are used to best represent the population studied and to give more or less importance to certain data. For example, if responses are missing for one unit, it will be given a lower weight. Survey weights therefore play a key role in limiting the presence of bias (Dever & Valliant, 2017). Secondly, this approach does not model an exhaustive relationship between common and non-common variables from A and B (Annoye et al., 2024). This shortcoming implies that the method may not fully capture the complex relationships between these variables, potentially leading to flawed or erroneous imputations.

For the reasons presented above and based on extensions of machine learning techniques, Annoye et al. (2024) have developed other statistical matching techniques. These techniques are the following: Kernel Canonical Correlation Analysis (KCCA), Autoencoder and Canonical Correlation Analysis (A-CCA) and Super-Organising Map (Super-OM). Super-OM will not be detailed in this work, but information is available in the article written by Annoye et al. (2024). It should be pointed out that the KCCA and A-CCA methods are derived from the Canonical Correlation Analysis (CCA) technique. In this literature review section, each of them will simply be introduced and the explanations of calculations for the CCA, KCCA and A-CCA approaches will be developed in the methodology section.

2.3.2.2 The Canonical Correlation Analysis method

The Canonical Correlation Analysis technique was developed in the 1930s by Harold Hotelling (1936) and involves analysing the relationships between two data sets. The objective of this approach is to maximise the correlation between variables from two databases using linear combinations known as "canonical variables". In this way, the CCA technique seeks to reduce the dimensionality of the two databases (Hotelling, 1936). However, as Asendorf (2015) points out in his thesis, the CCA approach is not a statistical matching algorithm. In fact, this technique provides transformations and correlations that highlight the common structure between the datasets, which can then be used for data fusion purposes (Asendorf, 2015).

The performance of the CCA approach is highly dependent on the volume of data available to train the model compared with the complexity (dimension) of the data. When there is a large amount of data and their complexity is lower, the CCA technique performs relatively well. On the contrary, when there

are few training samples, this approach may yield improbable and false results (Asendorf, 2015). To address this limitation, a number of extensions to CCA have been developed, such as KCCA, Sparse CCA, Regularised CCA and A-CCA (Annoye et al., 2024; Asendorf, 2015).

The Canonical Correlation Analysis method is an approach used in a wide variety of fields. Applications in the medical sector concerning, for example, genetic connections are detailed in Asendorf's thesis (2015). This method is also present in the fields of finance, marketing, music, and climatology, and is mainly known for its ability to find relationships between texts and images (Asendorf, 2015).

2.3.2.3 The Kernel Canonical Correlation Analysis approach

In the early 2000s, the Kernel Canonical Correlation Analysis (KCCA) approach was developed first by Lai and Fyfe (2000) and a later by Akaho (2006). As they explain in their article, the KCCA technique is an extension of the CCA technique that addresses one of its greatest limitations. Indeed, the KCCA approach is a machine learning method which makes it possible to model non-linear and therefore even more complex relationships using the kernel trick (Akaho, 2006; Lai and Fyfe, 2000). The kernel trick allows data to be implicitly transformed into a higher-dimensional (potentially infinite) feature space without having to calculate the coordinates of the points in this space. This allows algorithms to deal efficiently with non-linear problems by transforming them into linear problems in a higher dimensional space and applying the CCA method (Annoye et al., 2024; Lai and Fyfe, 2000) in this space.

Thanks to its many advantages, the KCCA method is a technique which is employed in a diverse range of fields. In cross-domain matching, Akaho (2006) talks about using the KCCA approach to link images to speech and vice versa. Indeed, given the high dimensionality of images and speeches, it is necessary to use higher-dimensional spaces to model non-linear relationships. Similarly, Shimodaira (2014) demonstrates in his paper the importance of the KCCA technique for matching images with text. It was only in 2020 that the KCCA approach made its appearance in the field of statistical matching thanks to Mitsuhiro and Hoshino (2020). However, the KCCA technique, which will be applied in this thesis, is derived from the article by Annoye et al. (2024). This method allows for the inclusion of both continuous and categorical variables, avoids incompatibilities with these categorical data, and takes survey weights into account, as specified by the authors.

2.3.2.4 The Autoencoder and Canonical Correlation Analysis method

The Autoencoder and Canonical Correlation Analysis technique is also an extension of the CCA approach and incorporates autoencoders. Although the term "autoencoder" is not explicitly mentioned, the underlying approach was developed by Rumelhart et al. (1986) in their article "*Learning representations by back-propagating errors*". The authors describe a neural network capable of compressing input data into an internal representation of reduced dimension, then reconstructing it from this representation, which corresponds to the functionality of an autoencoder. This is why, although the term itself is not used, the article is often associated with the introduction of autoencoders in the field of machine learning (Rumelhart et al., 1986). This approach is formalised in the definition proposed by Annoye et al. (2024):

"An autoencoder is an unsupervised neural network consisting of an encoder that compresses data efficiently by utilising the underlying structure therein, and a decoder, which decompresses the data into a representation that resembles the original version as closely as possible."

Thus, an autoencoder is composed of an encoder, a latent space, and a decoder. The encoders and decoders are defined to minimise the error between the original data and the reconstructed data (Annoye et al., 2024). The encoder and decoder are functions that are mostly in the form of neural networks (Michelucci, 2022). As Michelucci proves in his article, autoencoders are applied in various domains such as dimension reduction, data generation and image denoising.

In the specific framework of statistical matching, Luo et al. (2018) are the first to have made use of autoencoders. Indeed, they introduced a model based on autoencoders to improve the semantic consistency of responses in dialogue systems. By incorporating autoencoders, the model uses a statistical matching approach to learn and to evaluate the semantic dependency between utterances, thus generating more relevant and consistent responses in dialogues.

Finally, Annoye et al. (2024) developed the A-CCA method which involves using the autoencoder to compress the data and to represent them in a lower dimension in the latent space. Next, the CCA technique is applied to the compressed data in order to identify existing relationships between these variables and to maximise the correlation. Finally, the decoder is applied to the compressed data and attempts to reconstruct the original data (Annoye et al., 2024). The objective of the autoencoder is to minimise the difference between the input data and the reconstructed output data, while learning a useful and compact representation of the data.

2.3.2.5 Opening up the literature

In a recent publication, Annoye et al. (2024) presented the development of two novel techniques, KCCA and A-CCA, for statistical matching. Nevertheless, despite these notable advancements, further investigation is required to fully elucidate the potential of these two algorithms, particularly with regard to the optimisation of their hyperparameters. In particular, the dimensions of latent spaces and the bandwidth hyperparameters have yet to be fully explored with a view to obtaining optimal performance.

Specific details on bandwidths and the dimensions of latent spaces will be developed in the methodology section. These hyperparameters are of crucial importance for several reasons: they influence model flexibility, the ability to capture underlying patterns without overlearning, and robust generalisation to new data. Their optimisation is therefore essential to guarantee the effectiveness of the KCCA and A-CCA techniques in modelling relationships between complex data sets.

3 Methodology

This chapter will present the methodology that has been employed in the course of this study. Firstly, we will introduce the database that we have developed and that we will be working with. Next, we will explain the calculations related to the CCA, KCCA and A-CCA methods, as well as the management of categorical variables. Finally, we will discuss the measures that will be relevant for evaluating the performance of the predictions. It is also important to stress that all of this work will be carried out using RStudio software.

The approach adopted is mainly quantitative, relying on statistical analysis and machine learning algorithms to optimise hyperparameters and assess performance.

3.1 Presentation of the database

The data presented in this thesis was extracted from a database that was specifically constructed for the purpose of this study. This was necessary because access to information related to mobility and other domains is typically restricted due to the confidential nature of the personal data involved. The database was constructed within the R environment and comprises 5,000 rows and 30 columns, upon which dependency relationships were established.

To represent a variety of survey characteristics, both categorical and continuous data were included. The categorical data was generated using the "rbinom" function, which generates samples of random numbers following a binomial distribution. This function requires as parameters the number of observations to be generated, the number of trials in each Bernoulli trial, and the probability of success. On the other hand, continuous data was produced using the "rnorm" function, which generates samples of random numbers according to a normal distribution. It requires as parameters the number of observations to be generated, the mean and the standard deviation. The code implemented for the creation of the database is presented in Appendix A. A table showing the different variables, and their types is provided below:

Common variables		Non-common variables	
Variable	Type	Variable	Type
X1	Categorical	Y1	Categorical
X2	Continuous	Y2	Categorical
X3	Continuous	Y3	Categorical
X4	Continuous	Y4	Categorical
X5	Categorical	Y5	Continuous
X6	Continuous	Y6	Continuous
X7	Categorical	Y7	Continuous
X8	Categorical	Y8	Continuous
X9	Categorical	Y9	Continuous
X10	Continuous	Y10	Categorical
X11	Continuous	Y11	Categorical
X12	Continuous	Y12	Categorical
X13	Categorical	Y13	Categorical
X14	Categorical	Y14	Continuous
X15	Categorical	Y15	Continuous

***Table 1**: Type of the common and non-common variables*

For the purposes of this study, the database will be sub-sampled to allow the prediction of certain data from survey A. With this in mind, it was decided to hide the first 1,000 rows of the last 15 columns, corresponding to the top right-hand corner of the database. A visual representation of this operation is provided below:

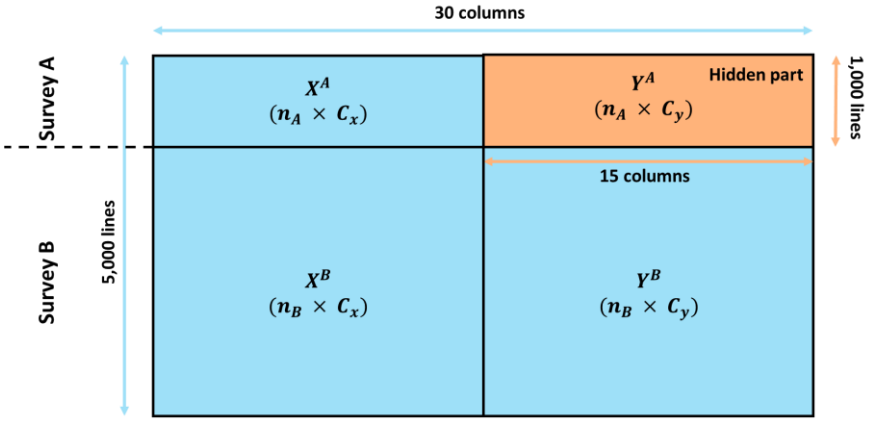


***Figure 4**: Illustration of the database and the hidden part to be predicted*

The database is therefore made up of two surveys, A (upper part of the figure – test sample) and B (lower part – training sample). Variables common to both surveys are represented by X, while variables specific to each survey are designated by Y. The number of rows in survey A are defined by n_A and n_B respectively, while the number of columns is denoted by C. In this context, the total number of rows, representing the number of individuals in each survey, is $1{,}000$ for the survey A and $4{,}000$ for the survey B. The number of variables (columns) of X and Y is 15 for both. Subsequently, it is normally necessary to use weight vectors for each observation in surveys A and B, defined respectively by w_A and w_B, whose associated diagonal matrices will be represented by W_A and W_B. However, since the variables are independent and identically distributed, weights are not needed in this case and will not be mentioned anymore. The values to be estimated are those in the "hidden part", denoted Y^A. In the code in the Appendix A, the data frames for $X^A, Y^A, X^B \; and \; Y^B$ are also defined.

Thus, thanks to the KCCA and A-CCA methods, a database comprising the sets of variables (X^A, Y^A), referred to in the literature as a "synthetic dataset", will be created. In this context, survey A will serve as the recipient, while survey B will be the donor. Then, the objective will be to optimise some hyperparameters of these two approaches in order to build the most accurate synthetical dataset possible, as close as possible to reality.

3.2 Explanations of both approaches based on the CCA method

In this section, we will begin by explaining the CCA method, which is necessary for the KCCA and A-CCA approaches. We will then develop the two-step procedure to impute the missing values in Y^A by handling categorical and continuous variables and detailing the two methods that will be implemented: KCCA and A-CCA. To do this, we will base ourselves on the article by Annoye et al (2024) in which they detail the calculations for this procedure. We will explain them in a literary manner to show our understanding.

3.2.1 Canonical Correlation Analysis (CCA)

Before going into more detail about the KCCA and A-CCA methods, it is necessary to focus first on the Canonical Correlation Analysis technique. As KCCA and A-CCA are extensions of the CCA approach, it is essential to firstly master the latter method.

As explained in the literature review, the main idea behind the CCA method is to maximise the correlation between linear combinations of X^B and Y^B variables. This is achieved by finding vectors a and b, known as canonical vectors, such that when X^B and Y^B are transformed by these vectors, the correlation between $X^B a$ and $Y^B b$ (the canonical variables) is maximised.

The constraints ensure that the solution is unique by controlling the variance of the transformed variables. Specifically, the constraints require that the variance of $X^B a$ and $Y^B b$, after transformation, is equal to one.

Once the canonical vectors, a and b, are derived from the CCA method, it is possible to process statistical matching by imputing the $\hat{Y}^A$. First, X^A, X^B and Y^B need to be centred and then, a kernel function, such as the Gaussian kernel, is applied to compute the pairwise distance among all elements of $X^B a$ and $X^A a$. One of the hyperparameters to be optimised in this thesis is the bandwidth hyperparameter used in the kernel.

Finally, the imputed values $\hat{Y}^A$ are obtained as a weighted mean of the variables in Y^B, where the weights are determined based on the distances between the transformed variables $X^B a$ and $X^A a$. These weights ensure that closer variables in the transformed space have a higher influence on the imputed values.

3.2.2 A two-step procedure

Normally, the two approaches consist of two stages. In the first stage, the categorical variables are imputed, thereby avoiding incompatibility errors between several variables. Indeed, in surveys concerning mobility and living conditions, variables relating to the place of residence and the municipality are included. Consequently, an individual residing in Alleur is incompatible with Liège as their municipality. In the second stage, the continuous variables are imputed with the KCCA or A-CCA techniques.

However, for the purposes of this thesis, the first phase will simply be described, but will not be studied or analysed. We will only focus on the second step, which solely consists of estimating the variables as if they were all continuous.

3.2.2.1 Handling categorical variables

In order to impute the categorical variables from $\hat{Y}^A$, it is necessary to construct the compatibility matrix Θ taking into account all the common variables, both X^A and X^B. For each row i (e.g. individual) in dataset A, its categorical variables are examined to find a similar row j (e.g. individual) in dataset B. If the i^{th} row in A has as counterpart the j^{th} row in B, Θ_{ij} is set to one or zero otherwise. If no counterpart is identified, the matching criteria are adjusted, and fewer variables are considered until a compatible counterpart is found. Finally, for each row in A, a row in B is randomly selected as the data source for imputation, considering a probability. This probability is defined considering compatibility,

the distance (similarity) calculated using the kernel between observations the i^{th} and j^{th} rows and the weight of observation j in B.

3.2.2.2 Two methods for handling continuous variables

The purpose of the second step is to impute the continuous variables (in our case, all the 15 variables to be imputed are considered continuous) using two different techniques. If the first step had taken place, the categorical variables imputed in this step would have had to be regarded as part of the common variables.

We will first describe the KCCA method in detail before turning to the A-CCA approach. To do this, we need to run the algorithms for the different methods on X^B and Y^B. These algorithms were supplied to us by my supervisor and developed by the project team.

3.2.2.2.1 The Kernel Canonical Correlation Analysis (KCCA) method

The KCCA approach differs from the CCA technique in being non-linear, which enables it to capture more precise relationships between the data. For the matrix X^B, each observation x_i^B is transformed to obtain a new vector $\Phi_x^B(X^B) = (\phi(\mathrm{x}_1), \dots, \phi(\mathrm{x}_n))$ which belongs to the Hilbert space H_x^B. Similarly, for the matrix Y^B, each observation y_i^B is transformed to obtain a new vector $\Phi_y^B(Y^B) = (\phi(\mathrm{y}_1), \dots, \phi(\mathrm{y}_n))$ which belongs to the Hilbert space H_y^B. A Hilbert space is a vector space with a scalar product and complete with respect to the norm induced by this scalar product.

Then, the inner products $\langle a|\Phi_x^B(X^B)\rangle$ and $\langle b|\Phi_y^B(Y^B)\rangle$ are calculated in Hilbert spaces and the correlation between these two products must be maximised while respecting the variance constraints and finding the vectors $a \in H_x^B$ and $b \in H_y^B$. The latter can be expressed as linear combinations of the transformed data. Thus, the correlation between $\sum_{i=1}^n \alpha_i \langle \phi_x^B(x_i^B)|\Phi_x^B(X^B)\rangle$ and $\sum_{i=1}^n \beta_i \langle \phi_y^B(y_i^B)|\Phi_y^B(Y^B)\rangle$ must be maximised, given that α_i and β_i are scalars.

Furthermore, Mercer's theorem allows inner products to be expressed in terms of positive definite symmetric kernels. These kernels are often expressed as Gramian matrices K_x^B and K_y^B, where the elements $(K_x^B)_{ij}$ and $(K_y^B)_{ij}$ correspond respectively to the kernel values $K_{h_x}^B(x_i^B, x_j^B)$ and $K_{h_y}^B(y_i^B, y_j^B)$, with h being the bandwidth hyperparameter.

The optimisation problem consists of finding α and β via a spectral decomposition problem. In order to prevent singularities and maintain the solution's uniqueness, a regularisation parameter γ is introduced.

Finally, to carry out the statistical matching, once α has been calculated, it is necessary to measure the distance between $K_{h_x}^B(x_i^B, x_j^B) \times \alpha$ and $K_{h_x}^A(x_i^A, x_j^B) \times \alpha$, then impute the values of $\hat{Y}^A$ by taking a weighted average of the variables in in Y^B, as was done in the CCA approach.

Therefore, the KCCA approach is distinguished by its capacity to discern complex dependencies between variables, a capability that stems from its non-linear nature. The utilisation of kernel functions facilitates the transformation of the original data into an infinite-dimensional Hilbert space, thereby enabling the more efficient detection of complex, non-linear relationships. Furthermore, the KCCA loss function is designed to optimise the correlation between projections of the data into this Hilbert space.

This direct maximisation of correlative relationships reveals more subtle underlying dependencies between data sets.

3.2.2.2.2 The Autoencoder and Canonical Correlation Analysis (A-CCA) approach

The ACCA approach relies on autoencoders, as discussed in the literature review. Autoencoders are designed to compress data using an encoder, reducing them to a lower-dimensional latent space. Then, a decoder reconstructs the data, aiming to closely match the original input. The goal is to minimise the mean-squared error between the original and reconstructed data, achieved by training both the encoder and decoder accordingly.

In this method, the first step involves the application of different autoencoders on the datasets X^B and Y^B in order to compress them into their respective latent spaces, denoted φ_X^B and φ_Y^B, which represent a reduced-dimensional version of the original data. Next, the CCA procedure is applied to these latent spaces to maximise the correlation between the canonical variables $\varphi_X^B a$ and $\varphi_Y^B b$ (a and b are still the canonical vectors), while respecting variance constraints.

To perform statistical matching, a kernel function is again applied to calculate the distance between $\varphi_X^B a$ and $\varphi_X^A a$. The values of $\hat{Y}^A$ are imputed as the weighted mean of the variables in Y^B, computed like in the CCA method.

The A-CCA approach is focused on the accurate reproduction of individual data using autoencoders whose loss function is designed to minimise the reconstruction error, which is to say, the difference between the original data and the reconstructed data. This results in a compact and accurate representation of the data, which is particularly useful for the replication of individual data sets. Nevertheless, although this approach can capture some non-linearities during compression, the canonical correlation applied in latent space remains linear. This may, therefore, limit the ability of the A-CCA technique to capture more complex dependencies between variables. The primary objective of this method is, therefore, to provide an accurate and compact representation of the original data set, rather than to optimise the correlative relationships between data sets.

3.2.3 Specification of the hyperparameters to be optimised

The objective of this work is to optimise the bandwidth hyperparameters and the dimension of latent spaces for both, the KCCA and A-CCA approaches, in order to provide more accurate imputations of $\hat{Y}^A$.

All bandwidth hyperparameters will be denoted by "p_2 …" (2 for the second phase).

3.2.3.1 Specifications of the hyperparameters to be optimised for the KCCA method

The first hyperparameters to be optimised in the KCCA approach come into play when Mercer's theorem is applied. They represent the bandwidths of the kernels used in the KCCA, which replace the scalar products in Hilbert space. These hyperparameters must be adjusted for both X^B and Y^B, giving a total of two hyperparameters to optimise: $p_2 h_x \; and \; p_2 h_y$. More specifically, the bandwidth hyperparameter is used to determine the shape and width of the Gaussian kernels (or other types of kernels) used to measure the similarity between pairs of points in Hilbert spaces. A poorly chosen bandwidth can lead to overfitting or underfitting of the data.

Then, in the KCCA approach, the latent space is a subspace of the Hilbert space H_x^B (or H_y^B) where X^B (or Y^B) is projected. In one dimension, there is only one component, the vector $K_x^B\alpha$ (or $K_y^B\beta$). However, it is possible to choose different α and β when spectrally decomposing the KCCA. For example, for a dimension three, there are three different α and β, giving $K_x^B\alpha_1, K_x^B\alpha_2, K_x^B\alpha_3$ and $K_y^B\beta_1, K_y^B\beta_2, K_y^B\beta_3$. This represents the latent space and its dimension to be optimised is defined by d. It determines the complexity of the KCCA model by specifying the number of dimensions in which the relationships between the X^B and Y^B data sets are captured. A higher number of latent dimensions allows a richer and more detailed representation of the relationships between the data but can also increase the risk of overfitting if the model is too complex for the amount of data available.

Finally, during the statistical matching phase at the end of the algorithm, a kernel function is again used to calculate the similarity between $K_{h_x}^B(x_i^B, x_j^B) \times \alpha$ and $K_{h_x}^A(x_i^A, x_j^B) \times \alpha$. This function requires a bandwidth hyperparameter to be set: $p_2 h$.

3.2.3.2 Specification of the hyperparameters to be optimised in the A-CCA approach

Firstly, the initial step in the A-CCA method is to apply a separate encoder to X^B and Y^B to project them into their respective latent spaces, φ_X^B and φ_Y^B. The dimension of these latent spaces must be defined: $p_2 lat_x, and\ p_2 lat_y$. The choice of latent space dimension can have an impact on the performance of the A-CCA approach in terms of the ability to discover correlations between datasets and the quality of the representation of compressed data. A higher latent space dimension can capture finer information about the data, but it can also increase the complexity of the model and lead to overlearning. On the other hand, a lower dimension can lead to a loss of essential information, but it can also simplify the model and improve its generalisation.

Moreover, when the CCA technique is applied to the data within the latent space, it is possible to have multiple vectors a and b in order to maximise the correlation between $\varphi_X^B a$ and $\varphi_Y^B b$. This is also recognised as a latent space, denoted as d, which is constrained to have a dimensionality that is either less than or equivalent to that of $\varphi_X^B a$ and $\varphi_Y^B b$.

Finally, in the data fusion stage, a kernel function is required to measure the distance between $\varphi_X^B a$ and $\varphi_X^A a$, which involves optimising its bandwidth. This hyperparameter needs to be specified: $p_2 h$.

3.3 Procedure to optimise these hyperparameters

In this section, the practical aspect of this work, focusing on the optimisation of the bandwidth hyperparameters and the dimension of latent spaces in the KCCA and A-CCA machine learning algorithms is discussed in the context of statistical matching. As already specified, the analyse will focus on the second step of the procedure concerning the continuous variable. The objective is to demonstrate the strategy used to adjust these crucial hyperparameters in order to obtain optimum performance during statistical matching.

The initial step involves naming the columns in the database. The first 15 variables, being common, are called "$X1, X2, \ldots, X15$" and the last 15, being non-common, are called "$Y1, Y2, \ldots, Y15$". Additionally, distinct data frames have been created to compartmentalise the dataset into its constituent parts: $XA, XB, YA, and\ YB$, where "X" corresponds to the common variables, "Y" to the non-common

variables, "A" to survey A data, and "B" to survey B data. Then, it becomes necessary to define the recipient and donor data frames, referred to as "$df.rec" = XA$ and "$df.don" = XB + YB$ in the code.

3.3.1 Kernel Canonical Correlation Analysis

As detailed above, the hyperparameters to optimise concerning the Kernel Canonical Correlation analysis are the following: the bandwidth of the kernels (p_2h_x, p_2h_y), the dimension of latent space (d), and the bandwidth hyperparameter in the kernel function used for the statistical matching (p_2h). These hyperparameters were optimised in two stages: first, the hyperparameters d and p_2h, then the hyperparameters p_2h_x, p_2h_y.

Other hyperparameters are considered in the KCCA approach but are not studied in this thesis. We have therefore decided to leave the default values. The hyperparameters not studied are listed in the table below, to ensure the reproducibility of the study:

Hyperparameters	Values
p2_g	0.00002
p2_n_combs	10
p2_objmethod	wsRMSE
p2_kernel_predict	gauss
n_fold	5
scaling	z-score
tuning_type	two h
type_predict	loop

***Table 2**: Value of unstudied hyperparameters in the KCCA approach*

In order to optimise d and p_2h, we chose to follow the Grid Search strategy. This hyperparameter optimisation technique involves specifying a grid of possible values for each model hyperparameter and then systematically evaluating the model for each combination of these values. This approach allows for determining the combination of hyperparameters that yields the best model performance according to one (or more) predefined metric(s), which will be explained later in this work.

Initially, we specified the following range of possible values for d: $1, 2, 3, 4, 5$ and for p_2h: $0.01, 0.11, 0.21, 0.31, 0.41, 0.51$. For each pair of hyperparameters (d, p_2h), we evaluated the model using the metrics that will be defined later, primarily focusing on the Root standardised Mean Squared Error (RsMSE). The objective was to minimise the RsMSE, indicating smaller errors in $\hat{Y}^A$ predictions. We subsequently extended the range for d to 7 and added 0.00051, 0.00076, 0.61, 0.71, 0.81 as possible values for p_2h. This allowed for observing a rise in the RsMSE and ensuring optimisation. The hyperparameters p_2h_x, p_2h_y were chosen through cross-validation to minimise the RsMSE among the following values: $0.01, 0.36, 0.71, 1.06$, which will later provide an indication of the optimum values for this hyperparameter.

Once d and p_2h have been optimised, we still need to find the optimum values for p_2h_x and p_2h_y. To do this, we selected the three pairs that minimise the RsMSE and the three pairs that minimise the CVM_C_NC (Cramer-Von Mises statistics, which will be defined in section 3.3.3) from the previous phase and provided several possible values based on those retained. First, we searched for the optimal

value for p_2h_y, again aiming to minimise the RsMSE and the CVM_C_NC respectively and continued testing values until this metric started to increase. Then, we applied the same principle to p_2h_x.

3.3.2 Autoencoder and Canonical Correlation Analysis

In the A-CCA method, the hyperparameters to be optimised are the followings: the dimension of the latent spaces $p_2lat_x \; and \; p_2lat_y$ in which the data are projected after the encoder, the dimension of latent space when applying the CCA (d), and the bandwidth hyperparameter in the kernel function used for the statistical matching (p_2h).

As with the KCCA method, other hyperparameters need to be defined for the A-CCA approach and will not be studied in this work. To ensure reproducibility of the results, the default values for the various hyperparameters not studied are given in the table below:

Hyperparameters	Values
scaling	z-score
p2_nlayers	2L
p2_epochs	200L
p2_batch_size	32L
p2_u	from 20L to 100L by 5L
p2_lr_min	-4
p2_lr_max	-1
p2_penL1_min	-6
p2_penL1_max	0
p2_penL2_min	-6
p2_penL2_max	0
p2_n_combs	110L
n_fold	5L
seed_phase2	123

***Table 3**: Value of unstudied hyperparameters in the A-CCA approach*

In addition, as the algorithm also uses the Python language, it was necessary to create a virtual environment to run the code.

For this method, we first optimised the dimensions of the various latent spaces, as they have a greater impact than the bandwidths derived from scalar products in the KCCA approach. Again, we followed the Grid Search strategy. We chose identical values for $p_2lat_x \; and \; p_2lat_y$, ranging from $1 \; to \; 15$ inclusive, with 15 being the maximum since there are 15 common and 15 non-common variables. For d, the range of possible values was set from $1 \; to \; 5$. Once again, each pair model $(d, p_2lat_x/p_2lat_y)$ was run and two performance metrics were calculated, with the main objective of observing an increase in RsMSE. Regarding the bandwidth hyperparameter p_2h, it was optimised through cross-validation to obtain an initial estimate of its optimal value among the following options: $0.1, 0.3 \; and \; 0.7$.

Secondly, the hyperparameter p_2h also needs to be optimised. To do this, we took again the three pairs $(d, p_2lat_x/p_2lat_y)$, which minimise the RsMSE and the three which minimise the CVM_C_NC. Then, we kept the value of p_2h obtained by cross-validation and tried slightly higher and lower values. We continued until the RsMSE and the CVM_C_NC, respectively, stopped decreasing and started increasing again.

3.3.3 Performance analysis

The final stage of the analysis consists of studying the performance of the imputations of $\hat{Y}^A$ in survey A using the methods described above. To do this, several performance measures have been used to analyse the quality of the estimates of $\hat{Y}^A$ by comparing them with the original data, Y^A.

3.3.3.1 Root standardised Mean Squared Error (RsMSE)

The Root standardised Mean Square Error (RsMSE) is a measure of the performance of a predictive model which considers both the accuracy of the predictions and the dispersion of the data. This measure is obtained by standardising the prediction errors in relation to the variance of the observed data, then taking the square root of this standardised value. It provides an indication of the standardised mean error, which makes it possible to compare the model's performance on different scales of data. The aim is to keep the values as low as possible. The formula is as follows:

$$RsMSE = \sqrt{\sum_{i=1}^{n} \left(\frac{y_i^A - \hat{y}_i^A}{(\sigma^A)^2} \right)^2}$$

Where n represents the number of observations, y_i^A denotes the original data, $\hat{y}_i^A$ is the prediction imputed in survey A and sigma is computed as follows: $(\sigma^A)^2 = \sum_{i=1}^{n} [y_i^A - (\sum_{i=1}^{n} y_i^A)]^2$

3.3.3.2 Cramer-Von Mises: bivariate and multivariate

The Cramer-Von Mises statistic represents the difference between the bivariate cumulative distribution function of two variables from the predictions and the bivariate cumulative distribution function of the same two variables in the original data. In the prediction dataset, the common and non-common variables are included. The outcome is a matrix containing all the statistics for each variable pair, where smaller values indicate better predictions. To obtain a single value from the output matrix, we took the average of these values. The result of this performance calculation is a single value, which should be as close to zero as possible. This suggests that the distribution of these variables in the predictions and in the original database is highly similar. In other words, a low value indicates that the empirical distribution is similar to the hypothetical distribution. The Cramer-Von Mises criterion is defined as:

$$CVM = \int_{\mathbb{R}} \int_{\mathbb{R}} [\hat{F}_{n_A}(x,y) - \hat{G}_{n_B}(x,y)]^2 \, dH_{n_A+n_B}(x,y)$$

Where $\hat{F}_{n_A}(x,y), \hat{G}_{n_B}(x,y)$ and $H_{n_A+n_B}(x,y)$ represent the bivariate empirical distributions of two variables X and Y in data sets A, B and $A+B$ respectively.

For this thesis, we calculated two Cramer-Von Mises statistics. The first consists of taking as variables, one from the non-common variables and one from the common variables. In this way, we studied the bivariate cumulative distribution function of a variable that did not have to be estimated (the common one "$X\ldots$") with one that went through statistical matching (the non-common one "$Y\ldots$"). The second statistics studied therefore the bivariate distribution of two non-common variables ("$Y\ldots$") which will have gone through the statistical matching procedure. The first statistics is denoted "CVM_C_NC" whereas the second one is named "CVM_NC_NC".

Finally, the CVM statistic for multivariate case follows the same principle as the one above but calculates the difference between the multivariate distribution of all the predicted and original variables, between $X^A Y^A$ and $X^A \hat{Y}^A$, and no longer the bivariate distribution. The result of this performance calculation ("CVM_all") therefore has only one value, which must be as close to zero as possible. As we did for the bivariate case, we also computed the statistic between only the non-common variables, between Y^A and $\hat{Y}^A$ ("CVM_all2").

4 Results

This chapter presents the results obtained by following the methodology described above and aims to address the research question: *"How to optimise the dimensions of latent spaces and bandwidth hyperparameters for two machine learning techniques?"*

The optimisation of these hyperparameters is crucial as it directly impacts the performance and accuracy of the KCCA and A-CCA algorithms in statistical matching tasks. Proper tuning of the bandwidth hyperparameters and the dimension of latent spaces can significantly enhance the algorithms' ability to estimate more accurately the non-common variables ($\hat{Y}^A$).

We will begin by analysing the results for the KCCA approach, detailing the impact of varying bandwidth hyperparameters and the dimension of latent spaces on the algorithm's performance. Subsequently, we will focus on the A-CCA approach, highlighting how these hyperparameters influence its efficacy.

The experiments were conducted using the database created by us, the algorithms provided by our supervisor and the R software. Tables and graphs will facilitate a clearer understanding of the results and will relate the various hyperparameters studied to determine which are optimal. This comprehensive analysis will help identify the configurations that maximise the performance of both algorithms under different conditions.

4.1 Results of the KCCA approach

For the KCCA approach, we first analysed the results obtained for optimising the hyperparameters d and p_2h before looking at the results for optimising $p_2h_x \; and \; p_2h_y$. For each of these two steps, the performance metrics of RsMSE and the Cramer-Von Mises criteria will be analysed in relation to the hyperparameters to be optimised. All the results are presented in the table in the Appendix B.

First of all, we would like to briefly analyse the two Cramer-Von Mises statistics introduced in the methodology: CVM_C_NC and CVM_NC_NC. At first glance, their values appeared to evolve in a similar manner. Consequently, we decided to create graphs illustrating their relationship as a function of the value of p_2h for each value of d. Additionally, we produced graphs showing their relationship as a function of the dimension d for each value of p_2h. These graphs are presented in Appendices C and D respectively, with CVM_NC_NC on the left axis (in blue) and CVM_C_NC on the right (in green).

Looking at the graphs, it is clear that they confirm our initial thoughts. In fact, the lines are superimposed and follow exactly the same trends, with the only difference being the scale of their values. Specifically, CVM_NC_NC values are approximately twice as much as those of CVM_C_NC. Although different, this scale difference is logical because the error in the distributions is greater when the statistic is calculated on two non-common variables Y estimated via the statistical matching technique, compared to when it is calculated with one common X (and therefore, non-estimated variable) and one non-common Y variable.

Given that these two metrics evolve in the same way, it is unnecessary to retain both for subsequent analyses. Thus, we have made the arbitrary choice to keep the CVM_C_NC statistic.

Moreover, we computed two multivariate Cramer-von Mises statistics to examine the similarity of the distributions of all common and non-common variables in the first stage (CVM_all) and only the non-common variables in the second stage (CVM_all2). As this metric is not yet algorithmically complete and is too complex to calculate, it will not be further analysed in this thesis. However, for informational purposes, we have included the graphs in the Appendices E and F, respectively. These graphs illustrate the evolution of these two metrics as a function of p_2h for each value of d. Then, they depict the relationship between these two metrics and the dimension d for each value of p_2h.

4.1.1 Step 1: results of the optimisation of d and p_{2h}

As explained in the methodology, the first hyperparameters to be optimised were the dimension of the latent space d during the spectral decomposition of the KCCA, and the bandwidth hyperparameter p_2h in the statistical matching phase. The graphs below will demonstrate which values optimise these hyperparameters and how the RsMSE and the Cramer-Von Mises criteria fluctuate in relation to them.

First, we studied the relationship between the bandwidth hyperparameter p_2h, the RsMSE and the Cramer-Von Mises statistics for each value of d.

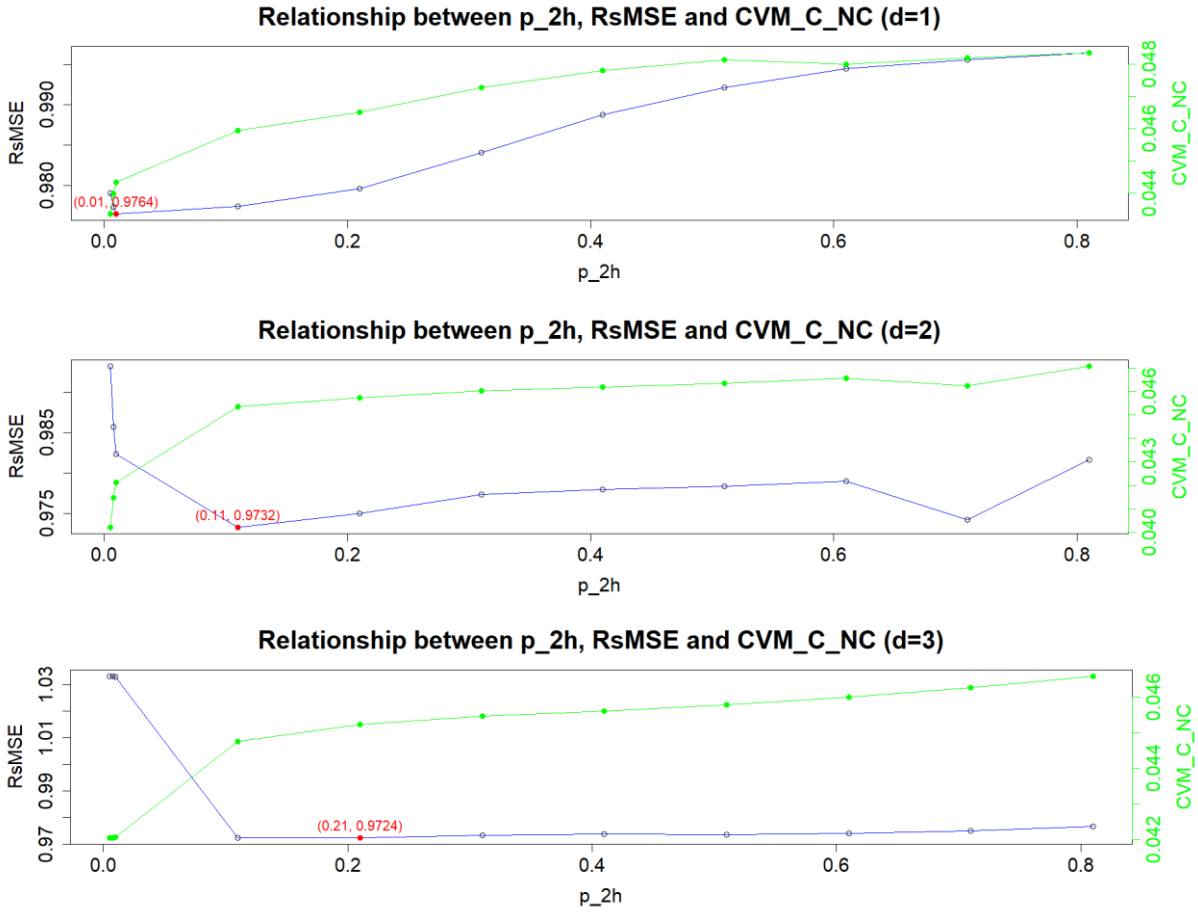


***Figure 5**: Relationship between the bandwidth hyperparameter ($p_{2h} = 0.0051,\ 0.0076,\ 0.01,\ 0.11,\ 0.21,\ 0.31,\ 0.41,\ 0.51,\ 0.61$), the CVM_C_NC and the RsMSE for $d = 1, 2, 3$*

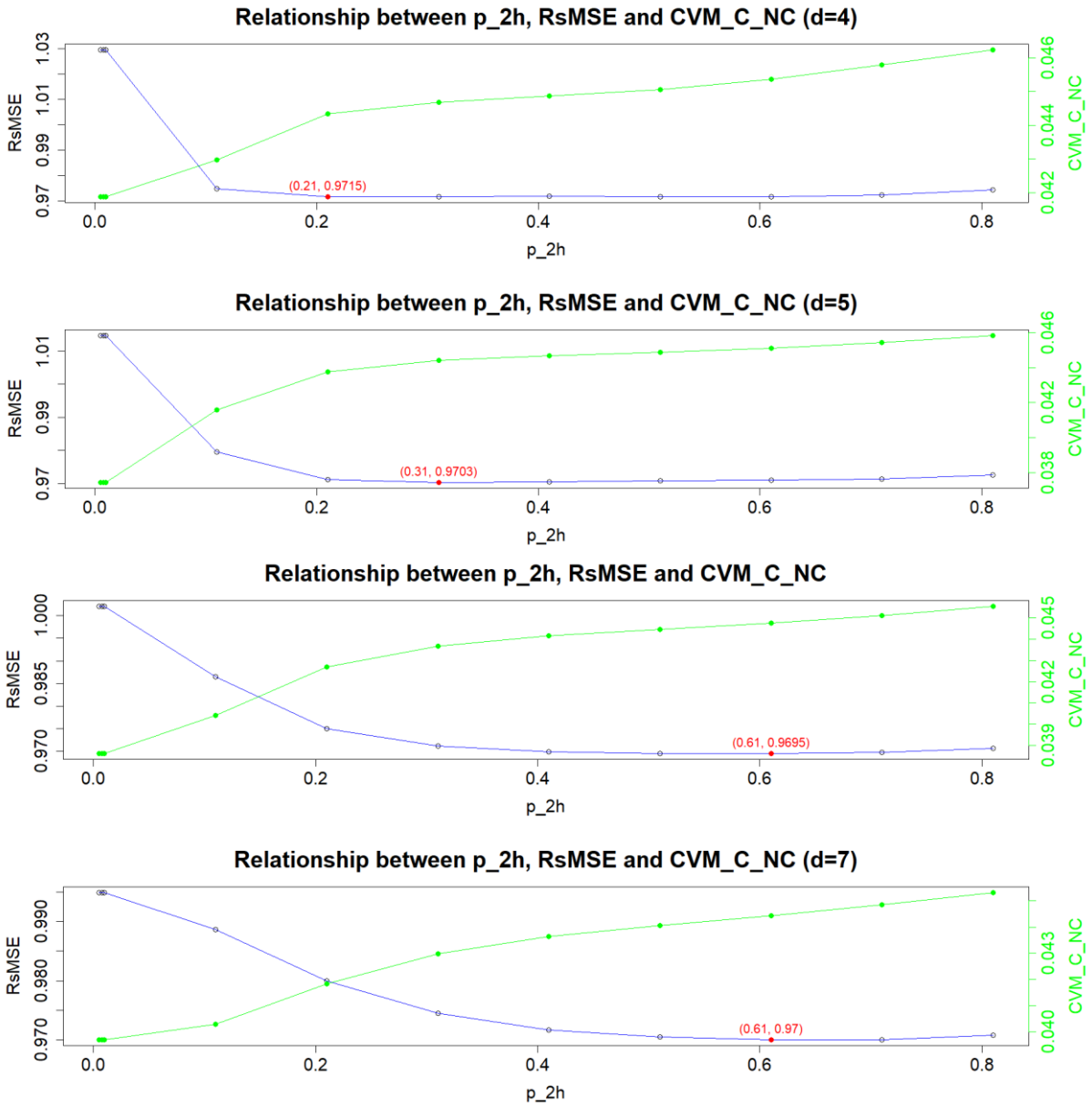


***Figure 6**: Relationship between the bandwidth hyperparameter ($p_{2h} = 0.0051, 0.0076, 0.01, 0.11, 0.21, 0.31, 0.41, 0.51, 0.61$), the CVM_C_NC and the RsMSE for $d = 4, 5, 6, 7$*

The graphs illustrate the relationship between the bandwidth p_2h, the Root squared Mean Squared Error (RsMSE) (blue curve), and the Cramer-Von Mises statistic ($CVM_{C_{NC}}$) (green curve) for different dimensions of the latent space d. Each red dot represents the pair (p_2h ; $RsMSE$) for which the RsMSE is the lowest. Here is an integrated analysis of both the RsMSE and CVM_C_NC results for the dimensions $d = 1\ to\ d = 7$.

For $d = 1$, the RsMSE increases almost linearly with increasing bandwidth p_2h, which suggests overfitting. Lower values of p_2h minimise the RsMSE, with its optimum value remarkably close to zero. The CVM_C_NC also increases with p_2h, suggesting that higher p_2h leads to higher values of this statistic, meaning bigger difference between the empirical and the estimating distributions. The green curve is more convex, indicating a smoother increase compared to RsMSE.

When $d = 2$, the RsMSE initially decreases as p_2h increases (underfitting), reaches a minimum and then, starts to increase slightly again (overfitting). The optimum point is observed when p_2h is approximately 0.1, with an RsMSE of 0.9372, representing the minimum reached. The CVM_C_NC increases sharply at the beginning and then, stabilises as p_2h increases, indicating better performance in terms of the Cramer-von Mises statistic at really lower values and stabilisation at higher values.

In the cases of $d = 3$ and $d = 4$, the RsMSE falls rapidly for initial values of p_2h and remains virtually constant after a certain value. For both dimensions, a value of p_2h around 0.2 seems optimal. This stability after the initial drop indicates that the choice of p_2h is less critical beyond this value and shows a stable trend for higher dimensions. The CVM_C_NC increases sharply between $p_2h = 0.01$ and $p_2h = 0.1$ for $d = 3$ and $p_2h = 0.2$ for $d = 4$. Then, it continues to increase with p_2h, but in a more stable and slight way, showing a divergent trend of RsMSE, indicating that the Cramer-Von Mises statistic becomes less favourable as p_2h increases.

When $d = 5$, the RsMSE decreases sharply for the first values of p_2h (underfitting) and then stabilises. The optimum value for p_2h is around 0.3, with the RsMSE stabilising thereafter. The CVM_C_NC increases steadily with p_2h, showing a clear trade-off with RsMSE. Higher p_2h leads to better RsMSE but worse CVM_C_NC.

For $d = 6$ and $d = 7$, the RsMSE decreases progressively with increasing p_2h until reaching a stable value around 0.5. The curve shows a continuous improvement up to this value, followed by stabilisation. Again, the CVM_C_NC increases steadily, similar to lower dimensions, with the best RsMSE performance around $p_2h = 0.5$ leading to higher CVM_C_NC values.

In conclusion, the RsMSE shows an increasing trend with the bandwidth hyperparameter p_2h, with different optimal values for different dimensions of the latent space d. Lower dimensions favor lower p_2h values, while higher dimensions show better stability at higher p_2h values. Regarding the Cramer-von Mises statistic, it generally increases with p_2h, suggesting a trade-off between minimising the RsMSE and maintaining lower CVM_C_NC values.

So, for each value of d, the table on the left shows the values of p_2h that minimise the RsMSE and the corresponding CVM_C_NC. On the right, the table represents the values p_2h which minimise the CVM_C_NC and the corresponding RsMSE.

d	p_2h	RsMSE	CVM_C_NC
1	0.01	0.97643903	0.04433659
2	0.11	0.97324435	0.04535497
3	0.21	0.97241528	0.04522858
4	0.21	0.97152487	0.04434243
5	0.31	0.97034796	0.04441377
6	0.61	0.96949255	0.04477321
7	0.61	0.97001816	0.04444051

***Table 5**: p_2h values minimising the RsMSE for each d and the corresponding CVM_C_NC in the KCCA method*

d	p_2h	RsMSE	CVM_C_NC
1	0.0051	0.97905404	0.04336115
2	0.0051	0.99323252	0.04021099
3	0.0051	1.0329563	0.04205199
4	0.01	1.02945862	0.04188252
5	0.01	1.01458959	0.03745322
6	0.01	1.00207712	0.03862195
7	0.01	0.99487972	0.03969673

***Table 4**: p_2h values minimising the CVM_C_NC for each d and the corresponding RsMSE in the KCCA method*

On the one hand, the three best pairs $(d\,;p_2h)$ which minimise the RsMSE are $(d=5\,;p_2h=0.31)$, $(d=6\,;p_2h=0.61),(d=7\,;p_2h=0.61)$. On the other hand, the pairs $(d=5\,;p_2h=0.01),(d=6\,;p_2h=0.01),(d=7\,;p_2h=0.01)$ are the most effective in minimising the CVM_C_NC.

Then, we studied the relationship between d, the RsMSE and the Cramer-Von Mises statistics (CVM_C_NC) for each value of p_2h.

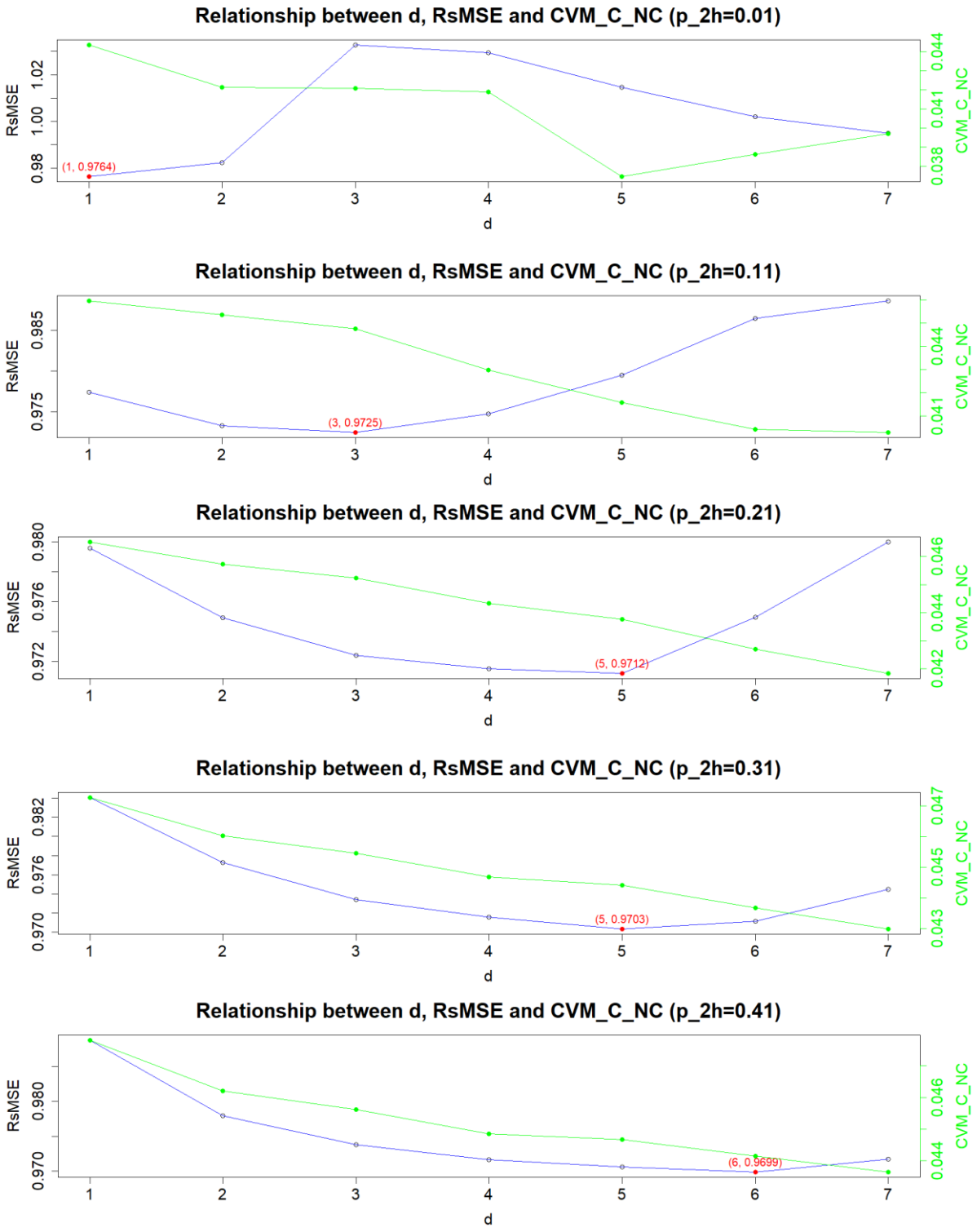


***Figure 7**: Relationship between the dimension of the latent space ($d = 1,2,3,4,5,6,7$), CVM_C_NC and the RsMSE for $p\_2h = 0.01, 0.11, 0.21, 0.31, 0.41$*

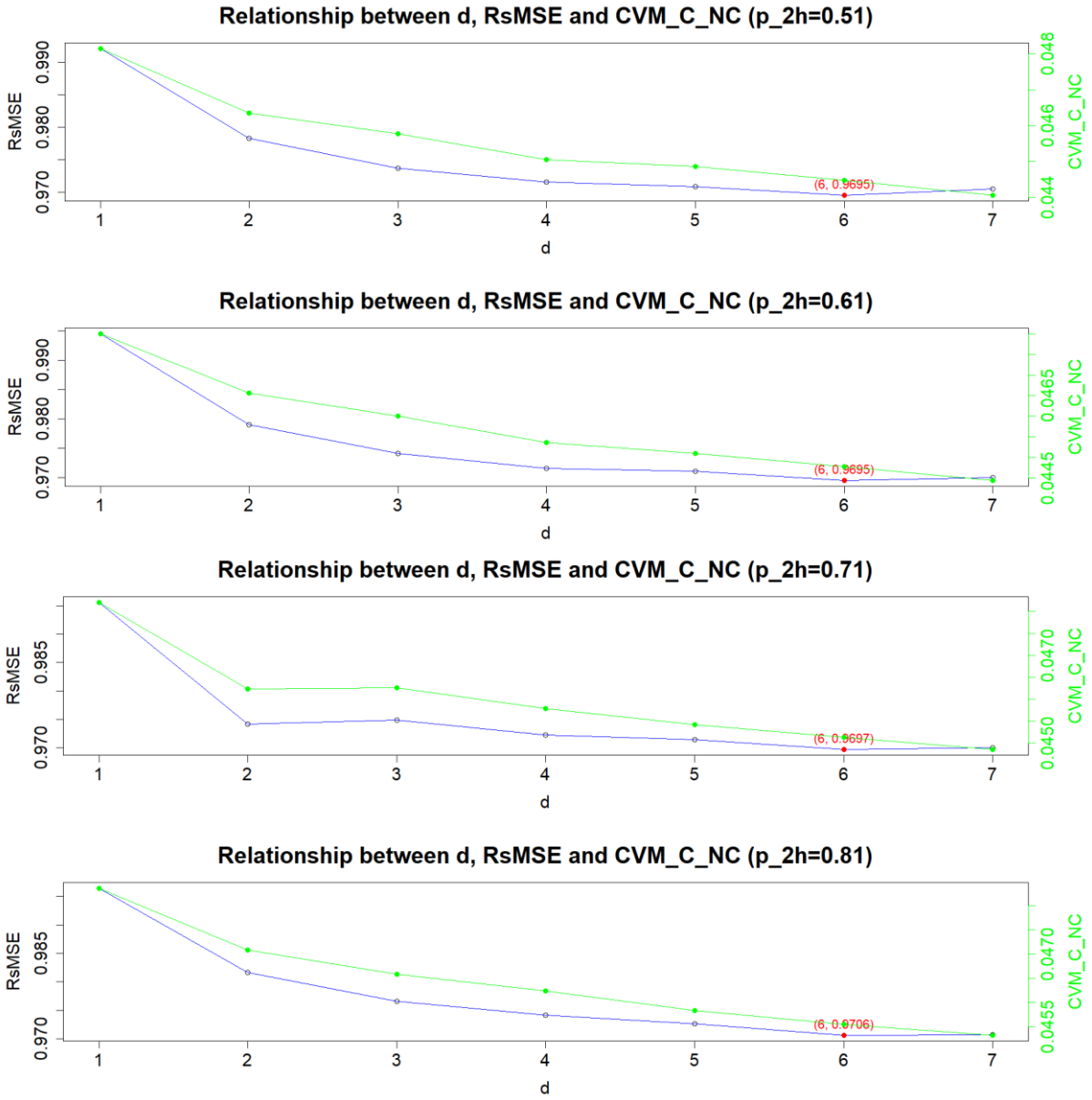


***Figure 8**: Relationship between the dimension of the latent space ($d = 1, 2, 3, 4, 5, 6, 7$), CVM_C_NC and the RsMSE for $p\_2h = 0.41, 0.51, 0.61, 0.71, 0.81$*

The graphs illustrate the relationship between the dimension of the latent space d and the Root standardised Mean Squared Error (RsMSE) (blue curve), as well as the Cramer-von Mises statistic (CVM_C_NC) (green curve) for different values of the bandwidth p_2h. Each red dot represents the minimum RsMSE value for a given pair (d ; $RsMSE$). Here is an integrated analysis of both the RsMSE and CVM_C_NC results for the bandwidths $p_2h = 0.01,\ 0.11,\ 0.21,\ 0.31,\ 0.41,\ 0.51,\ 0.61$.

For $p_2h = 0.01$, the RsMSE varies non-linearly with the dimension of d. There is an increase in RsMSE for the first few dimensions, followed by a decrease. The optimum value of the RsMSE is reached for $d = 1$ with an RsMSE 0.9764, which is the lowest. After that, the RsMSE increases and then decreases again, suggesting variability in performance efficiency with very small values of p_2h. The green curve shows a variable pattern with d, initially decreasing, then stabilising before decreasing again and finally increasing. The lowest value is at $d = 5$.

When $p_2h = 0.11$, the RsMSE shows a U-shaped curve, with an initial decrease (underfitting situation), reaching a minimum (the optimum point), then a gradual increase (overfitting situation). The optimum value of RsMSE is reached for $d = 3$ with an RsMSE of 0.9752, suggesting that for this value of p_2h, a higher dimension is beneficial. The CVC_C_NC decreases consistently with d, suggesting better performance in terms of CVC_C_NC as dimension increases, diverging from the RsMSE trend.

The RsMSE for $p_2h = 0.21, 0.31, 0.41, 0.51, 0.61, 0.71\ and\ 0.81$ follow a similar trend and take on a U-shaped form. In fact, an initial decrease in RsMSE is observed for the first values of d (underfitting case) before stabilising and finally increasing slightly (overfitting situation). For $p_2h = 0.21$ and $p_2h = 0.31$, the optimum dimension is $d = 5$ while for $p_2h = 0.41, 0.51, 0.61, 0.71\ and\ 0.81$, the optimum dimension is found at $d = 6$. This confirms that for higher values of p_2h, the dimensions of the latent space must be larger to minimise the RsMSE.

Regarding the CVM_C_NC for $p_2h = 0.21, 0.31\ and\ 0.41$, the green curve shows a consistent decrease with d. For $p_2h = 0.51, 0.61, 0.71\ and\ 0.81$, the CVM_C_NC decreases with d as well and show a slight stabilisation for higher dimensions of the latent space d.

In conclusion, for $p_2h = 0.1$, the RsMSE shows non-linearity with varying dimensions d, but for higher p_2h values, the RsMSE demonstrates a convex, U-shaped pattern, stabilising after certain dimensions. Concerning the CVM_C_NC, it generally decreases with increasing d, showing better alignment with RsMSE at higher dimensions for larger p_2h values.

In order to effectively optimise the performance of the KCCA algorithm in statistical matching tasks, it is essential to simultaneously adjust the hyperparameters p_2h and d. This integrated approach makes it possible to find the optimal combinations that maximise the accuracy and efficiency of the matching, underlining the importance of considering these hyperparameters synergistically for optimal results.

The following table on the left shows the dimensions of the latent space d that minimises the RsMSE for each value of the bandwidth hyperparameter p_2h and the corresponding CVM_C_NC. On the right, the table demonstrates the dimensions of the latent space d, which minimise the CVM_C_NC and the corresponding RsMSE.

p_2h	d	RsMSE	CVM_C_NC
0.01	1	0.97643903	0.04433659
0.11	3	0.97245081	0.04475069
0.21	5	0.97121182	0.04377147
0.31	5	0.97034796	0.04441377
0.41	6	0.96986976	0.04415795
0.51	6	0.96952296	0.04447152
0.61	6	0.96949255	0.04477321
0.71	6	0.96974415	0.04512671
0.81	6	0.97063921	0.04554658

***Table 7**: dimension d minimising the RsMSE for each p_2h values and the corresponding CVM_C_NC for the KCCA method*

p_2h	d	RsMSE	CVM_C_NC
0.01	5	1.01458959	0.03745322
0.11	7	0.98862122	0.04028877
0.21	7	0.98000338	0.04183872
0.31	7	0.97449551	0.04298634
0.41	7	0.9716909	0.04364932
0.51	7	0.97051676	0.04406417
0.61	7	0.97001816	0.04444051
0.71	7	0.9700257	0.04485431
0.81	7	0.97077701	0.04531666

***Table 6**: dimension d minimising the CVM_C_NC for each p_2h values and the corresponding RsMSE for the KCCA method*

From this table, the three pairs $(d\,;p_2h)$ which minimise the RsMSE are the followings: $(d = 6\,;p_2h = 0.51), (d = 6\,;p_2h = 0.61), (d = 6\,;p_2h = 0.71)$. Concerning the CVM_C_NC, the

pairs $(d = 5\,; p_2h = 0.01), (d = 7\,; p_2h = 0.11), (d = 7\,; p_2h = 0.21)$ are the most effective to minimise the Cramer-Von Mises statistics.

4.1.2 Step 2: results of the optimisation of p_2h_x and p_2h_y

Having found the values of d and p_2h that optimise the performance of the algorithm for the KCCA method, we still need to find the optimal values of the dimensions of the bandwidth hyperparameters p_2h_x and p_2h_y. As a reminder, they represent the bandwidths of the kernels used in the KCCA method, which replace the scalar products in the Hilbert space. To do this, we selected the three pairs $(d\,;\ p_2h)$ which minimise the RsMSE and the three pairs which minimise the CVM_C_NC. Then, we used the best values of p_2h_x and p_2h_y which had been defined by the cross-validation strategy.

Moreover, to ensure that the application of the cross-validation technique was consistent and accurate, we repeated the procedure outlined below using pairs of $(d\,; p_2h)$ that resulted in higher RsMSEs, indicating lower performance. The results of these tests are presented in tables in the Appendix G. For these pairs, adjusting the p_2h_x and p_2h_y hyperparameters to optimise them did not result in RsMSE values lower than the best RsMSEs previously obtained. Consequently, this analysis confirms that the cross-validation technique is appropriate and effective for these hyperparameters. This conclusion is further supported by the fact that, despite attempts to optimise the hyperparameters, the performance metrics did not surpass those achieved earlier.

4.1.2.1 Results of the optimisation by minimising the Root standardised Mean Squared Error

The results of the initial step were analysed, and the three most successful couples, in terms of minimising RsMSE, are presented in the following table.

d	p_2h	p_2hx	p_2hy	RsMSE	CVM_C_NC
6	0.51	0.01	0.01	0.96952296	0.04447152
6	0.61	0.01	0.01	0.96949255	0.04477321
6	0.71	0.01	0.01	0.96974415	0.04512671

***Table 8**: The three pairs with the lowest RsMSE in the KCCA approach*

The 3D graphs below visualise the relationship between the performance metric RsMSE (represented on the blue axis), and the bandwidth hyperparameters p_2h_x (on the red axis) and p_2h_y (on the green axis) in the Kernel Canonical Correlation Analysis (KCCA). The results are presented in the Appendix H.

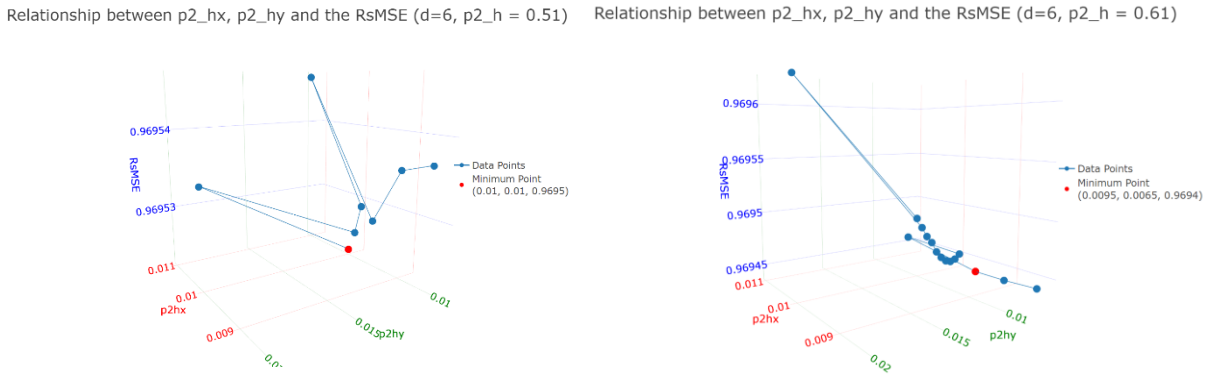


***Figure 9**: Relationship between p_2h_x, p_2h_y and the RsMSE for $d = 6$ and $p_2h = 0.51$ on the left and for $d = 6$ and $p_2h = 0.61$ on the right*

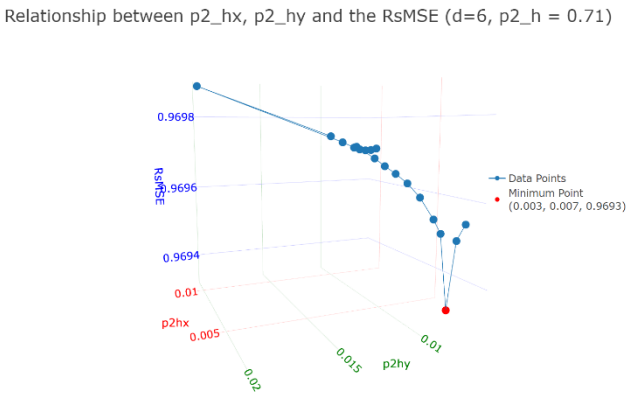


***Figure 10**: Relationship between p_2h_x, p_2h_y and the RsMSE for $d = 6$ and $p_2h = 0.71$*

Upon analysing these three graphs, it becomes apparent that the bandwidth hyperparameters p_2h_x and p_2h_y affect the RsMSE, and thus the performance of the algorithm. Certain combinations of these two hyperparameters result in a lower RsMSE, although these differences are very slight. For $p_2h = 0.51$, none of the hyperparameter values found through cross-validation improves the model's performance by reducing the RsMSE. On the contrary, all other combinations increase this performance metric.

For $p_2h = 0.61$ and $p_2h = 0.71$, the data points show a similar general trend: the RsMSE decreases, and therefore performance increases, as p_2h_x and p_2h_y decrease. However, their minimum points have different values, and the sensitivity of the RsMSE is slightly more pronounced for $p_2h = 0.71$. This means that for this value ofp_2h, small variations in the hyperparameters p_2h_x and p_2h_y can lead to more significant differences in the model's performance.

In summary, optimising the bandwidth hyperparameters p_2h_x and p_2h_y is crucial for minimising the RsMSE and improving the algorithm's performance. Variations in p_2h show different sensitivities, with optimal performance achieved for specific values of p_2h_x and p_2h_y. The table below demonstrates for each of these three pairs the optimal values of both hyperparameters and their RsMSE.

d	p_2h	p_2hx	p_2hy	RsMSE	CVM_C_NC
6	0.51	0.01	0.01	0.96952296	0.04447152
6	0.61	0.0095	0.0065	0.96944154	0.04473360
6	0.71	0.003	0.007	0.96933382	0.04463136

***Table 9**: Optimal values of the hyperparameters minimising the RsMSE and the corresponding CVM_C_NC in the KCCA method*

4.1.2.2 Results of the optimisation by minimising the Cramer-Von Mises statistic

In this section, we will analyse the results concerning the optimisation of the p_2h_x and p_2h_y from the Cramer-Von Mises perspective. The three pairs that minimised this metric are presented in the table below with the values determined through cross-validation for p_2h_x and p_2h_y. The results of the optimisation are presented in the table in the Appendix I.

d	p_2h	p_2hx	p_2hy	RsMSE	CVM_C_NC
5	0.01	0.01	1.06	1.01458959	0.03745322
6	0.01	0.01	0.01	1.00207712	0.03862195
7	0.01	0.01	0.01	0.99487972	0.03969673

***Table 10**: The three pairs with the lowest CVM_C_NC in the KCCA approach*

The 3D graphs below illustrate the relationship between the Cramer-Von Mises statistic (CVM_C_NC, represented on the blue axis), the bandwidth hyperparameters p_2h_x (on the red axis) and p_2h_y (on the green axis) in the KCCA method for each pair listed above.

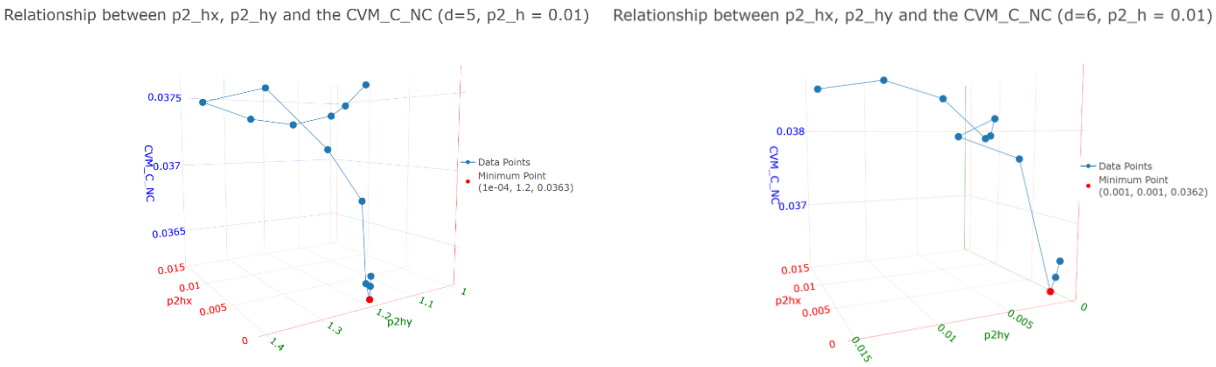


***Figure 11**: Relationship between p_2h_x, p_2h_y and the CVM_C_NC for $d = 5$ and $p_2h = 0.01$ on the left and for $d = 6$ and $p_2h = 0.01$ on the right*

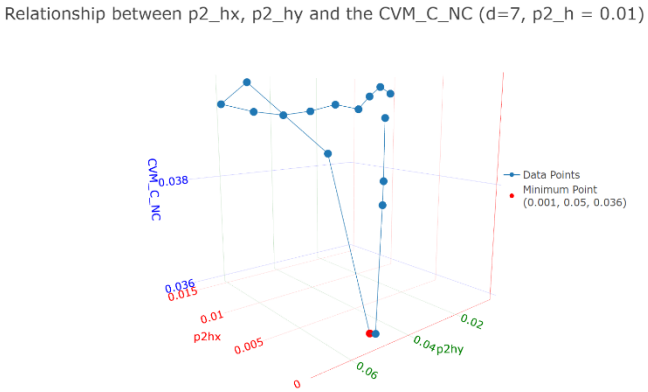


***Figure 12**: Relationship between p_2h_x, p_2h_y and the CVM_C_NC for $d = 7$ and $p_2h = 0.01$*

Upon analysing these graphs, we observe that the Cramer-von Mises statistic is influenced by variations in p_2h_x and p_2h_y. Each graph reveals a combination that minimises this statistic, thereby enhancing the algorithm's performance in maintaining dependencies between variables.

For the cases where $d = 5$ and $d = 7$, the CVM_C_NC decreases as p_2h_y increases and p_2h_x decreases, reaching their minimal points at $(0.0001\,;\,1.2)$ and $(0.001\,;\,0.05)$ respectively, beyond which the CVM_C_NC rises again. Conversely, when $d = 6$, the Cramer-von Mises statistic decreases as both p_2h_x and p_2h_y decrease. Once the minimum value is achieved, this metric begins to increase.

In conclusion, the analysis of these graphs demonstrates that the optimal performance of the KCCA algorithm, as measured by the Cramer-Von Mises statistic, is achieved by carefully selecting the

bandwidth hyperparameters p_2h_x and p_2h_y. Beyond these optimal points, the performance metric worsens. The table below presents the optimal points for each graph:

d	p_2h	p_2hx	p_2hy	RsMSE	CVM_C_NC
5	0.01	0.0001	1.2	1.04417036	0.0362753
6	0.01	0.001	0.001	1.03230632	0.0362205
7	0.01	0.001	0.05	1.03139704	0.03599899

***Table 11**: Optimal values of the hyperparameters minimising the CVM_C_NC and the corresponding RsMSE in the KCCA approach*

4.2 Results of the A-CCA approach

The results of the optimising hyperparameters for the A-CCA method (presented in the Appendix J) will be analysed in two stages. First, we will analyse the optimal values for the dimension of the latent space when the CCA technique is applied (d), as well as the dimensions of the latent spaces (φ_X^B and φ_Y^B) when the encoder is applied for the common (X^B) and non-common (Y^B) variables, denoted as $p_2lat_x \; and \; p_2lat_y$, respectively. Next, we will perform an analysis to optimise the bandwidth hyperparameter involved in the statistical matching phase, p_2h. Similar to the KCCA technique, the RsMSE and the Cramer-Von Mises criteria will be used as performance metrics.

As was the case with the KCCA approach, the two metrics, CVM_C_NC and CVM_NC_NC, appear to be very similar. Therefore, we conducted the same analysis as before, creating graphs showing these two metrics, once as a function of d and again as a function of $p_2lat_x \; and \; p_2lat_y$. These graphs are included in the Appendices K and L, respectively. The conclusions are identical: the two metrics follow the same trend and evolve together, with the only difference being their scale, explained in the same way as for the KCCA approach. To remain consistent, we decided to only keep the CVM_C_NC metric for the rest of the analyses.

In addition, we also calculated the two multivariate Cramer-von Mises statistics for the A-CCA approach. For the same reasons explained for the KCCA method, these two metrics will not be further discussed, and the graphs are provided in the Appendices M and N for information.

4.2.1 Step 1: results of the optimisation of d and $p_2lat_x \; and \; p_2lat_y$

In this section, we will analyse the results obtained from the algorithms when optimising the hyperparameters d and $p_2lat_x \; and \; p_2lat_y$. They demonstrate the relationships between the different hyperparameters and allow us to find the values that optimise the performance of the algorithm and the A-CCA technique used.

Firstly, we studied the relationship between the dimension of the latent spaces for $p_2lat_x \; and \; p_2lat_y$, the RsMSE (the blue curve) and the CVM_C_NC (the green curve) for each value of d. Each red dot represents the pair (p_2lat_x/p_2lat_y ; $RsMSE$) for which the RsMSE is the lowest. Here is an integrated analysis of both the RsMSE and CVM_C_NC results for the dimensions $d = 1 \; to \; d = 5$.

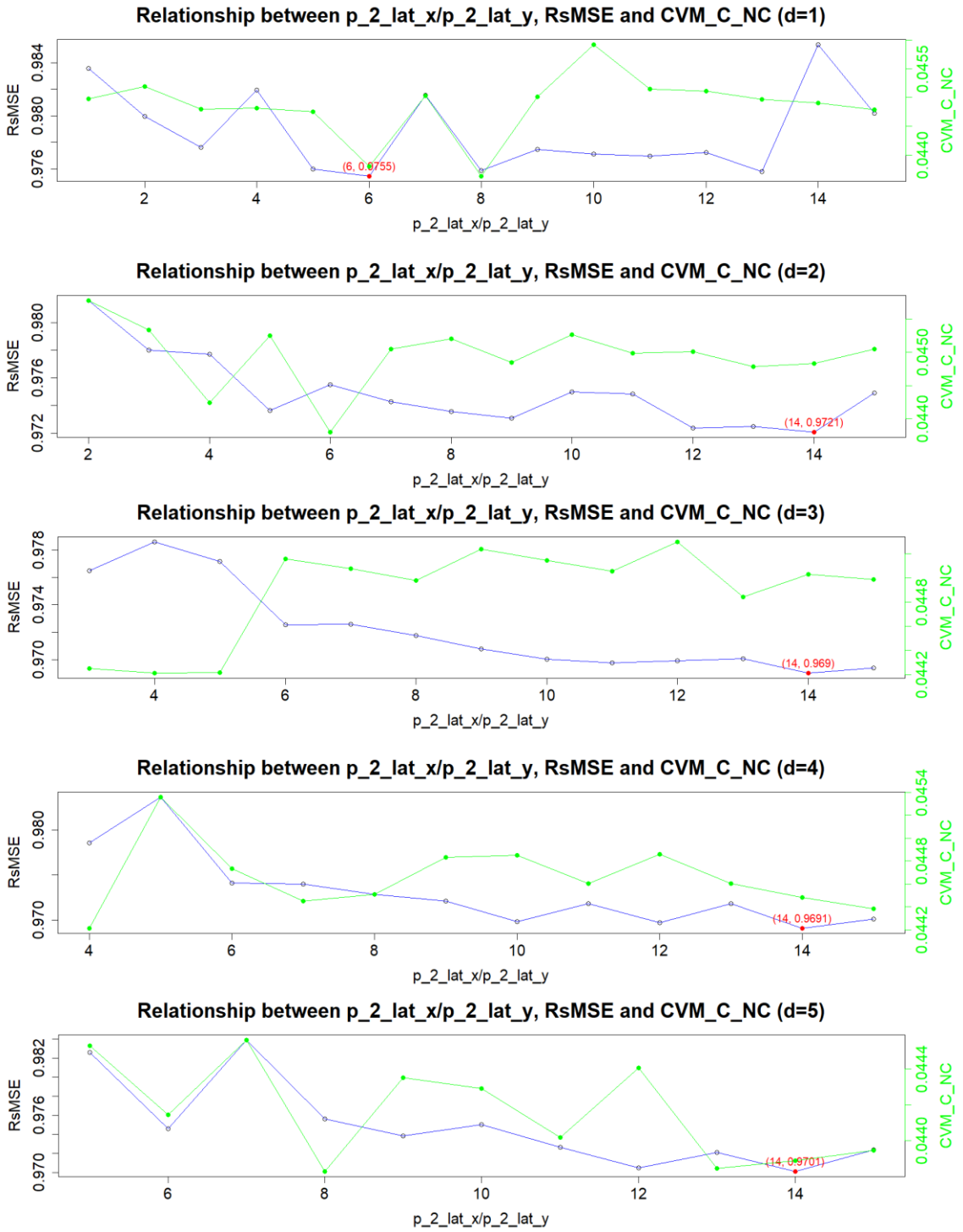


***Figure 13**: Relationship between the dimension of the latent spaces p_2lat_x, p_2lat_y, the RsMSE and the CVM_C_NC for $d = 1, 2, 3, 4\ and\ 5$*

For $d = 1$, the RsMSE varies significantly without any discernible trend, fluctuating between $0.976\ and\ 0.982$. Although these values show slight differences, it is not possible to observe a clear relationship between these hyperparameters. The optimum point is identified when the RsMSE is at its lowest, which occurs at $p_2lat_x/p_2lat_y = 6$. A similar pattern is observed with the Cramer-Von-Mises statistic, which also fluctuates but exhibits a slight upward trend. Again, the range of values remains very narrow.

When d is set to a value of two, the RsMSE initially shows a very downward trend, followed by a slight decrease with a few upturns. For its part, CVM_C_NC has no definite trend and fluctuates between $0.045\ and\ 0.055$.

For $d\ =\ 3$, the RsMSE shows a clear downward trend and is slightly U-shaped, with its minimum point at $p_2lat_x/p_2lat_y\ = 14$. The CVM_C_NC first increases before stabilising at around 0.045.

In the case where $d\ =\ 4$, the RsMSE fluctuates with a rather decreasing trend before stabilising and fluctuating more slightly to reach a minimum RsMSE when $p_2lat_x/p_2lat_y\ = 14$. Once again, CVM_C_NC fluctuates with no real trend, with lower values as the dimension of $p_2lat_x\ and\ p_2lat_y$ increases.

Finally, when $d\ =\ 5$, the RsMSE fluctuates quite strongly for low dimension of $p_2lat_x\ and\ p_2lat_y$ before stabilising somewhat as the dimensions of the latent spaces $p_2lat_x/p_2lat_y\ = 14$ and finding an optimum value of 14. As for CVM_C_NC, it fluctuates significantly, with no decipherable trend.

In conclusion, the analysis of the graphs suggests that the optimisation of the hyperparameters $p_2lat_x\ and\ p_2lat_y\ = 14$ must be carried out by balancing the values of RsMSE and CVM_C_NC to obtain efficient dimension reduction while preserving the structure of the data, with a preference for $p_2lat_x/p_2lat_y\ = 14$ for the d values studied.

The following table on the right shows the optimal values for the dimensions of the latent spaces $p_2lat_x\ and\ p_2lat_y$ which minimise the RsMSE and the corresponding CVM_C_NC for each value of d. The table on the left presents the optimal values for the same hyperparameters which minimise the CVM_C_NC and the corresponding RsMSE.

d	p_2_lat_x/p_2_lat_y	RsMSE	CVM_C_NC
1	8	0.97589503	0.04362909
2	6	0.97550459	0.04380251
3	4	0.97859419	0.04421293
4	4	0.97859419	0.04421293
5	8	0.97560768	0.04382965

Table 13: *Dimension of p_2lat_x/p_2lat_y minimising the CVM_C_NC for each value of d and the corresponding RsMSE in the A-CCA approach*

d	p_2_lat_x/p_2_lat_y	RsMSE	CVM_C_NC
1	6	0.97545452	0.04380102
2	14	0.97207616	0.04483187
3	14	0.96901617	0.04503044
4	14	0.96906588	0.04448076
5	14	0.97012742	0.04388928

Table 12*: Dimension of p_2lat_x/p_2lat_y minimising the RsMSE for each value of d and the corresponding CVM_C_NC in the A-CCA approach*

The three pairs $(d\ ;\ p_2lat_x/p_2lat_y)$ with the lowest RsMSE are the following: $(d\ =\ 3\ ;\ p_2lat_x/p_2lat_y = 14)$, $(d\ =\ 4\ ;\ p_2lat_x/p_2lat_y = 14)\ and\ (d\ =\ 5\ ;\ p_2lat_x/p_2lat_y = 14)$ whereas the three pairs which minimise the CVM_C_NC are $(d\ =\ 1\ ;\ p_2lat_x/p_2lat_y = 8)$, $(d\ =\ 2\ ;\ p_2lat_x/p_2lat_y = 6)\ and\ (d\ =\ 5\ ;\ p_2lat_x/p_2lat_y = 8)$.

Secondly, we analysed the relationship between the dimension of the latent space d, RsMSE and Cramer-Von Mises statistic for each dimension of latent spaces p_2lat_x/p_2lat_y, ranging from 5 to 15. Again, the red dots represent the pair $(d\ ;\ RsMSE)$ with the lowest RsMSE. The results are plotted below.

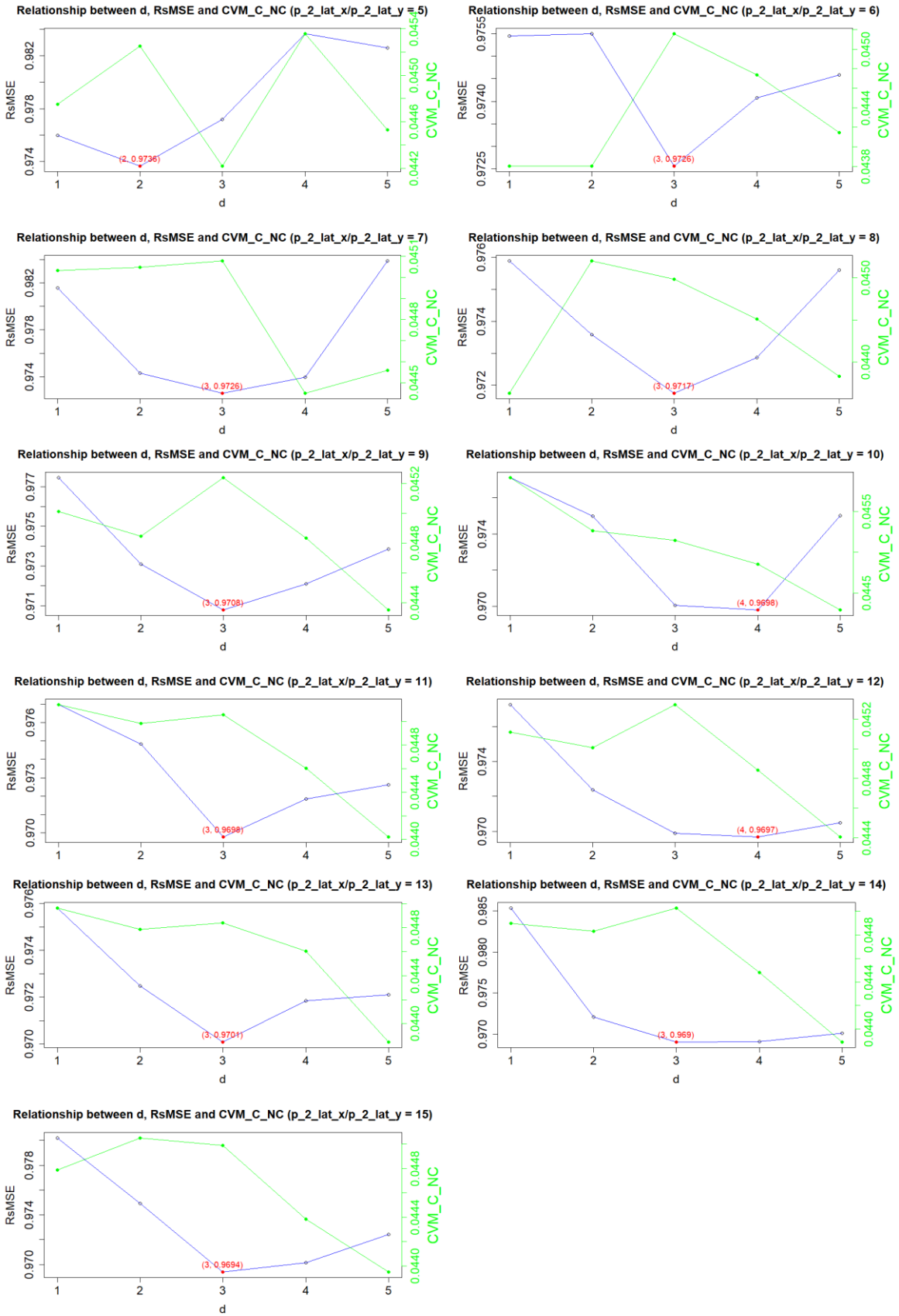


Figure 14: *Relationship between the dimension of the latent spaces* d, *the RsMSE and the CVM_C_NC for* $p_2lat_x, p_2lat_y = 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 \; and \; 15$

The graphs presented above illustrate remarkably similar trends, particularly in relation to the RsMSE curve. This curve exhibits a consistent trajectory for each dimension of p_2lat_x/p_2lat_y, except for $p_2lat_x/p_2lat_y = 5$, which deviates slightly with an additional decline at the end of the trajectory. For all other dimensions of the latent spaces p_2lat_x/p_2lat_y, the RsMSE initially decreases as d increases (underfitting), before rising (overfitting) again after a certain point. This inflection point, highlighted in red on the graphs, signifies the minimum value of the RsMSE, thereby determining the optimal dimension d. The curves exhibit a convex shape.

The Cramer-von Mises curve (CVM_C_NC), however, does not follow a uniform pattern. For $p_2lat_x/p_2lat_y = 5$, the curve demonstrates fluctuations without a clear trend. Conversely, for $p_2lat_x/p_2lat_y = 6$ and $p_2lat_x/p_2lat_y = 8$, the CVM_C_NC rises sharply before falling more gently. For the other values of p_2lat_x/p_2lat_y, the CVM_C_NC shows an overall downward trend with increasing d, without reaching a minimum. This suggests that increasing the dimension of the latent space enhances the alignment of the distributions of the latent variables, as measured by the Cramer-Von Mises statistic.

In conclusion, there is a clear trade-off between RsMSE and CVM_C_NC. Generally, when the RsMSE reaches its minimum value, the CVM_C_NC tends to be relatively high, and thus not at its minimum. It is noteworthy that, despite variations, the range of fluctuations remains relatively small. The graphs suggest that a d dimension between 3 and 5 often proves to be the most optimal. This range offers a favourable compromise between minimising reconstruction error and maintaining statistical quality.

The values of d that minimise the RsMSE for each value of p_2lat_x/p_2lat_y and the CVM_C_NC are shown in the right table whereas those which minimise the CVM_C_NC and the corresponding RsMSE are presented in the left table.

p_2_lat_x/p_2_lat_y	d	RsMSE	CVM_C_NC
5	3	0.97717816	0.04421802
6	1	0.97545452	0.04380102
7	4	0.97396798	0.04445039
8	1	0.97589503	0.04362909
9	5	0.97386637	0.04435141
10	5	0.97503427	0.04429091
11	5	0.97263231	0.04401833
12	5	0.97049084	0.04440585
13	5	0.97210940	0.04384583
14	5	0.97012742	0.04388928
15	5	0.97240679	0.04394768

***Table 15**: Dimension of d minimising the CVM_C_NC for each value of p_2lat_x/p_2lat_y and the corresponding RsMSE in the A-CCA approach*

p_2_lat_x/p_2_lat_y	d	RsMSE	CVM_C_NC
5	2	0.97364917	0.04524948
6	3	0.97255804	0.04515918
7	3	0.97257693	0.04507901
8	3	0.97174023	0.04498079
9	3	0.97078040	0.04523993
10	4	0.96978120	0.04485230
11	4	0.97184003	0.04460411
12	4	0.96967678	0.04485837
13	4	0.97184003	0.04460411
14	3	0.96901617	0.04503044
15	3	0.96937601	0.04498851

***Table 14**: Dimension of d minimising the RsMSE for each value of p_2lat_x/p_2lat_y and the corresponding CVM_C_NC in the A-CCA approach*

Thus, the three pairs $(d\,;\, p_2lat_x/p_2lat_y)$ with the lowest RsMSE are the following: $(d = 4\,;\, p_2lat_x/p_2lat_y = 12)$, $(d = 3\,;\, p_2lat_x/p_2lat_y = 14)$ and $(d = 3\,;\, p_2lat_x/p_2lat_y = 15)$. The three pairs, which minimise the CVM_C_NC are $(d = 1;\, p_2lat_x/p_2lat_y = 6)$, $(d = 1\,;\, p_2lat_x/p_2lat_y = 8)$ and $(d = 5\,;\, p_2lat_x/p_2lat_y = 13)$.

4.2.2 Step 2: results of the optimisation of p_2h

Now that the optimal dimensions of the various latent spaces have been defined, we need to analyse the results for the optimisation of the bandwidth used in the statistical matching phase. To achieve this, we selected the three pairs $(d\,; p_2lat_x/p_2lat_y)$ that minimised the RsMSE and the three pairs that minimised the Cramer-Von-Mises statistic in the first step and focused on optimising the bandwidth hyperparameter for each of these six pairs.

Moreover, as with the KCCA approach, we tested the cross-validation technique by optimising the p_2h hyperparameter for pairs $(d\,;\, p_2lat_x/p_2lat_y\,)$ with lower RsMSE to see if they could obtain better RsMSE than the other pairs. By analysing the tables with the results found in the Appendix O, it is possible to conclude that the cross-validation technique is consistent since none of these "less good" couples managed to obtain better RsMSE.

4.2.2.1 Results of the optimisation by minimising the Root standardised Mean Squared Error

We first analysed the results focusing on the RsMSE. Therefore, we decided to optimise the hyperparameter p_{2h} for these three pairs, which minimise the RsMSE:

d	p_2_lat_x/p_2_lat_y	p_2h	RsMSE	CVM_C_NC
3	14	0.7	0.96901617	0.04503044
3	15	0.7	0.96937601	0.04498851
4	14	0.7	0.96906588	0.04448076

***Table 16**: The three pairs with the lowest RsMSE in the A-CCA method*

Using the cross-validation technique results for hyperparameter p_2h obtained in the first step and noted in the table above, we explored additional upper and lower values to determine if the RsMSE could be further reduced. We continued this iterative process until an increase in the RsMSE was observed for the three selected points. Tables in the Appendix P show all the results of this process.

This systematic approach ensures that the bandwidth hyperparameter is fine-tuned for optimal performance, leveraging the most promising dimensions identified earlier. By iterating until the RsMSE no longer decreases, we can confidently identify the best settings for the bandwidth, thus enhancing the overall accuracy and reliability of the statistical matching phase. The results are shown in the graph below, which illustrates the relationship between the bandwidth hyperparameter and the RsMSE.

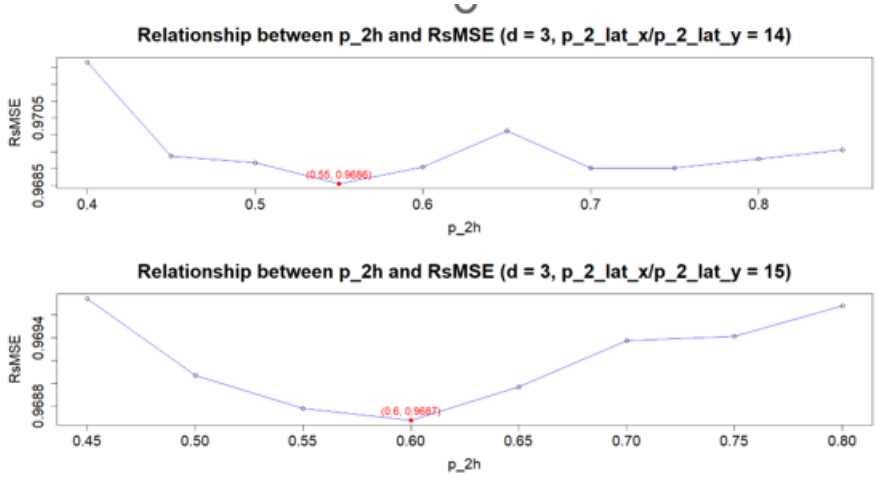


***Figure 15**: Relationship between p_2h and the RsMSE for $d\ =\ 3\,;\ p_2lat_x = p_2lat_y =\ 14$ in the top graph, $d\ =\ 3,\ p_2lat_x = p_2lat_y\ =\ 15$ in the bottom graph and $d\ =\ 4$*

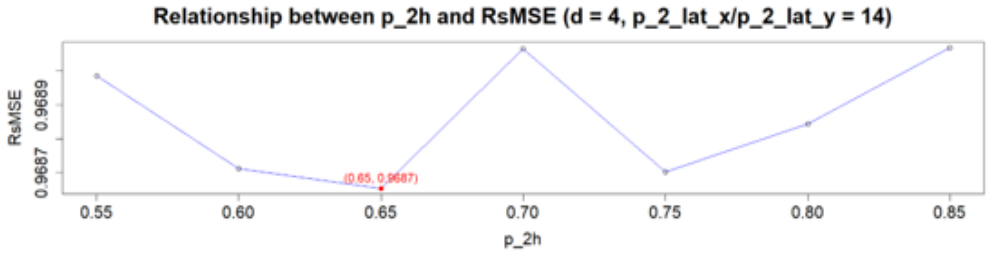


***Figure 16**: Relationship between p_2h and the RsMSE for $d = 4$; $p_2lat_x = p_2lat_y = 14$*

The subsequent three pairs $((d = 3; p_2lat_x = p_2lat_y = 14); (d = 3; p_2lat_x = p_2lat_y = 15); (d = 4; p_2lat_x = p_2lat_y = 14))$ exhibit a comparable trend. Specifically, the RsMSE initially decreases as the p_2h hyperparameter increases (underfitting situation), reaching a minimum at values of $0.55, 0.60, and\ 0.65$ respectively. Following this minimum point, the RsMSE begins to rise again (overfitting case), indicating that the optimal bandwidth hyperparameters correspond to the points where the minimum RsMSE is reached.

From the graphs, it is evident that the optimal bandwidth hyperparameter (p_2h) varies depending on the specific dimensions of the latent spaces. The identified optimal values $(0.55, 0.60, and\ 0.65$ for the respective pairs) provide the lowest RsMSE, indicating the best fit for the model. These results are crucial for fine-tuning the bandwidth hyperparameter to ensure minimal reconstruction error and enhance the model's accuracy and reliability in the statistical matching phase.

d	p_2_lat_x/p_2_lat_y	p_2h2	RsMSE	CVM_C_NC
3	14	0.55	0.96855524	0.04453943
3	15	0.60	0.96867608	0.04466020
4	14	0.65	0.96865352	0.04433593

***Table 17**: Optimal values of the hyperparameters minimising the RsMSE and the corresponding CVM_C_NC for the A-CCA approach*

4.2.2.2 Results of the optimisation by minimising the Cramer-Von Mises statistic

Then, we analysed the results obtained for p_2h after minimising the CVM_C_NC. Therefore, we have identified and optimised the three pairs which yield the lowest values for this metric, as follows:

d	p_2_lat_x/p_2_lat_y	p_2h	RsMSE	CVM_C_NC
1	6	0.7	0.97545452	0.04380102
2	6	0.7	0.97550459	0.04380251
1	8	0.7	0.97589503	0.04362909

***Table 18**: The three pairs with the lowest CVM_C_NC in the A-CCA method*

To optimise the bandwidth hyperparameter, we used the results obtained by cross-validation in the first step and varied it to minimise the CVM_C_NC. The results of this analysis are shown below in graphs illustrating the relationship between the Cramer-Von Mises statistic and the p_2h value. The results are presented in the Appendix Q.

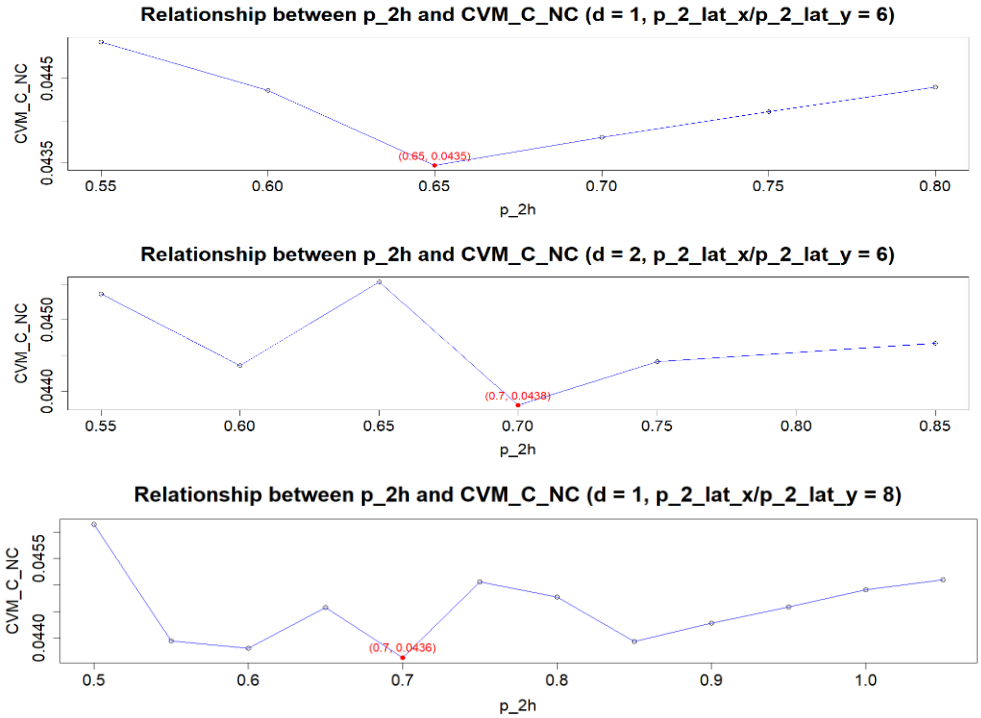


***Figure 17**: Relationship between p_2h and the CVM_C_NC for $d = 1$; $p_2lat_x/p_2lat_y = 6$ in the top graph, $d = 2$, $p_2lat_x = p_2lat_y = 6$ in the middle graph and $d = 1$; $p_2lat_x/p_2lat_y = 8$ in the bottom graph*

Upon analysing these graphs, we observe that the Cramer-von Mises statistic is dependent on the bandwidth hyperparameter p_2h. For the pair $(d = 1 ; p_2lat_x/p_2lat_y = 6)$, the CVM_C_NC decreases until it reaches the optimal point at $p_2h = 0.65$, after which it increases again. This suggests that maximum performance is achieved at this point, where the CVM_C_NC is at its minimum.

For the other two graphs, the behaviour of the Cramer-von Mises statistic is less linear and more complex compared to the first graph. We observe more fluctuations in CVM_C_NC as p_2h varies, with several local minima and maxima. Nevertheless, both graphs still exhibit a p_2h value that minimises the CVM_C_NC. The best performance is observed at $p_2h = 0.7$ for both.

In conclusion, the optimal values of the bandwidth hyperparameter p_2h vary with the dimension of the latent spaces, highlighting the importance of carefully tuning these hyperparameters to achieve optimal performance of the algorithm from the perspective of the Cramer-von Mises statistic.

d	p_2_lat_x/p_2_lat_y	p_2h	RsMSE	CVM_C_NC
1	6	0.65	0.97644962	0.04347295
2	6	0.70	0.97550459	0.04380251
1	8	0.70	0.97826579	0.04457400

***Table 19**: Optimal values of the hyperparameters minimising the CVM_C_NC and the corresponding RsMSE for the A-CCA method*

5 Discussion

Thanks to the results obtained in the previous section, it is possible to draw conclusions and attempt to answer the research question which, as a reminder, was: *"How to optimise bandwidth hyperparameters and the dimensions of the latent spaces in machine learning algorithms for the KCCA and A-CCA methods".*

This discussion is divided into two parts, the first for the KCCA approach and the next for the A-CCA method. For each of these techniques, the answer to the question will be set out, explaining the choice, and developing the ideas from different points of view, in particular depending on which performance metric we are considering: the Root standardised Mean Squared Error or the Cramer-Von Mises statistic.

It should first be noted that the RsMSE metric is designed to evaluate the precision of predictive outcomes. Accordingly, optimising RsMSE performance entails minimising the discrepancies between the predicted values $\widehat{Y^A}$ and the actual values Y^A. This is a highly sensitive metric, particularly in the presence of outliers. In contrast, the Cramer-Von Mises statistical metric (CVM_C_NC) compares the cumulative distribution functions of predicted and actual values. Its objective is to minimise the difference/distance between the two distributions, thereby ensuring similarity and, in part, preserving the dependencies between the variables.

5.1 Kernel Canonical Correlation Analysis

This section will examine the values of the various hyperparameters that enable the optimisation of algorithm performance in the context of the KCCA approach. The initial focus will be on the Root standardised Mean Squared Error metric, after which the Cramer-Von Mises statistic will be discussed and the differences in the results explained. Ultimately, conclusions can be drawn regarding the optimal hyperparameters.

Firstly, as a reminder of point 4.1.2.1, the values of the hyperparameters that maximise performance according to the RsMSE among the three possibilities are highlighted in the following table.

d	p_2h	p_2hx	p_2hy	RsMSE	CVM_C_NC
6	0.51	0.01	0.01	0.96952296	0.04447152
6	0.61	0.0095	0.0065	0.96944154	0.04473360
6	0.71	0.003	0.007	0.96933382	0.04463136

***Table 20**: Hyperparameter values maximising performance of the KCCA approach according to RsMSE (best values highlighted in red)*

From this table, it can be inferred that in order to optimise the performance of the Kernel Canonical Correlation Analysis machine learning algorithm from the RsMSE perspective, the latent space dimension d during spectral decomposition must be set to 6. The bandwidth hyperparameters of the kernel $p_2h_x \; and \; p_2h_y$, employed in the KCCA to replace the scalar products in Hilbert space, should be set to 0.003 and 0.007, respectively. Lastly, the bandwidth p_2h used during the statistical matching phase should be 0.71. In this case, the minimum RsMSE value was found to be 0.96933382. Given that the value is less than one, it can be inferred that the prediction error is less than the standard deviation of the data, which serves to demonstrate that the model in question is an accurate fit for the data set.

In this context, it is pertinent to highlight the value of the Cramer-Von Mises statistic, which is equivalent to 0.04463136.

Next, the table above from section 4.1.2.2 shows the optimum values for the various hyperparameters in order to minimise the Cramer-Von Mises statistic (CVM_C_NC), the optimal values of which are highlighted in red.

d	p_2h	p_2hx	p_2hy	RsMSE	CVM_C_NC
5	0.01	0.0001	1.2	1.04417036	0.0362753
6	0.01	0.001	0.001	1.03230632	0.0362205
7	0.01	0.001	0.05	1.03139704	0.03599899

***Table 21**: Hyperparameter values maximising performance of the KCCA approach according to CVM_C_NC (best values highlighted in red)*

From this analysis, we can conclude that to optimise the KCCA algorithm from the perspective of the Cramer-von Mises statistic, the dimension of the latent space d should be set to 7 and the bandwidth hyperparameter p_2h during the statistical matching phase should be 0.001. The bandwidth hyperparameters p_2h_x and p_2h_y should be 0.001 and 0.05, respectively. In these circumstances, the value of CVM_C_NC will reach a minimum of 0.035999, which indicates that the cumulative distribution functions of the model's predicted outcomes and the actual data are markedly similar. The model thus demonstrates an aptitude for preserving the global dependencies between the variables and performs well from the perspective of the Cramer-Von Mises statistic. Furthermore, it is important to note that the RsMSE value in this case is 1.03139704.

A review of the results as a function of the two-performance metrics previously mentioned reveals that the hyperparameters are not identical and exhibit opposing trends. It was observed that for each dimension of the latent space d, the RsMSE exhibited a convex shape and decreased as the bandwidth p_2h increased. Conversely, the CVM_C_NC was low for low values of p_2h and then increased sharply. Similarly, for each value of p_2h, the RsMSE metric exhibited a convex shape, initially decreasing as d increased before reaching a minimum and rising again. In contrast, the CVM_C_NC continued to decrease as the dimension of d increased.

Thus, we see that when the RsMSE is optimised, the Cramer-Von Mises statistic is relatively high (value 0.04463136 and its minima 0.03599899). These values show that the model performs well from the RsMSE point of view, whereas from the CVM_C_NC point of view, the distance between the empirical and original cumulative distribution functions is higher, indicating worse performance. Similarly, when we optimise according to this statistic, the RsMSE reaches almost its maximum (worth 1.03139704, while its minimum is equivalent to 0.96933382). In this case, this means that the dispersion of the predicted data is greater than that of the original data, indicating a worse performance of the model in terms of RsMSE.

This behaviour and this difference can mainly be explained by the different objectives of the two measures. The RsMSE focuses on the accuracy of predictions, while the Cramer-Von Mises statistic focuses on the distribution of predicted data compared to actual data. This divergence makes it more difficult to align the two objectives simultaneously.

In addition, a higher latent space dimension d better captures the dependencies between variables, preserves more structural features of the data and more accurately represents complex relationships, thus reducing the Cramer-Von Mises statistic. However, this can also lead to overfitting as the model tends to capture the noise present in the training data, increasing the RsMSE and computational complexity. In fact, by capturing the finest details, the model may lose generalisation and produce larger prediction errors.

In contrast, a lower dimension d simplifies the model by reducing the number of hyperparameters and allows better generalisation of the new data, thus avoiding overfitting. As a result, the RsMSE tends to be lower because the model makes more robust predictions. However, a lower dimension d only captures the principal components and may miss more complex relationships between variables, which will increase the value of the Cramer-Von Mises statistic.

Subsequently, the bandwidth hyperparameter p_2h determines the degree to which data points in close proximity exert an influence on the calculations performed by the Gaussian kernel function. A low p_2h value is more effective at capturing local dependencies and slight variations, as p_2h increases the sensitivity of the kernel function to minor discrepancies between $U_a \ and \ U_b$. Consequently, the Cramer-Von Mises statistic is diminished due to the increased similarity between the distributions of the real and imputed data, as well as the enhanced representation of local dependencies. Nevertheless, a low value may result in overfitting, as the capture of minute details may also encompass the capture of noise. This phenomenon can elevate the value of the RsMSE, as the model demonstrates suboptimal performance on the data to be imputed.

Conversely, a high value for the bandwidth p_2h results in a considerable reduction in the impact of variations, thereby rendering the kernel function less sensitive to minor differences. Consequently, the smoothed function results in a loss of detail and an increase in the Cramer-Von Mises statistic due to the increased distance between the cumulative distributions. With regard to the RsMSE point, a higher bandwidth permits generalisation through the smoothing of the smallest variations, thereby reducing the risk of overfitting. Consequently, the RsMSE is reduced, as the predictions are more robust and less susceptible to significant errors.

Finally, it is difficult to draw conclusions regarding the relationship between p_2h_x, p_2h_y and the algorithm's performance, given the complexities associated with these two variables. Indeed, our findings indicate that these values exert an influence on the evaluation metrics (RsMSE and CVM_C_NC). However, no discernible trend could be identified. The optimisation of these hyperparameters was conducted on an individual basis, and thus, no comprehensive analysis can be presented.

In conclusion, in order to optimise the performance of the algorithm for the Kernel Canonical Correlation Analysis approach, it is necessary to either choose between optimising the Root standardised Mean Squared Error and the Cramer-Von Mises statistic, or to find a balance. As previously stated, the RsMSE and CVM_C_NC metrics assess disparate aspects of a prediction model's performance, rendering simultaneous optimisation of these two metrics challenging, if not impossible.

Therefore, if the primary objective is to optimise RsMSE, the values of the hyperparameters that optimise the algorithm are as follows: $d = 6$; $p_2h = 0.71$; $p_2h_x = 0.003$; $p_2h_y = 0.007$. In

contrast, if the primary objective is to optimise the Cramer-Von Mises statistic while maintaining the data structure, the optimal values are: $d = 7$; $p_2h = 0.01$; $p_2h_x = 0.001$; $p_2h_y = 0.05$.

Ultimately, in order to enhance the efficacy of the algorithm by considering both metrics concurrently, it is essential to strike a balance between the precision of the predictions (minimisation of errors) and the preservation of the dependencies between the variables (minimisation of the distance between the empirical and predicted cumulative distribution functions). In order to determine the optimal value for the dimension of the latent space d, it is necessary to identify a value that is sufficiently high to capture as many dependencies between the variables as possible, while simultaneously ensuring that it is low enough to minimise prediction errors. Similarly, a compromise must be reached regarding the value of the bandwidth, whereby global dependencies are preserved while accurate and robust predictions are offered.

5.2 Autoencoder and Canonical Correlation Analysis

This section will delineate the requisite values of the various hyperparameters for optimising the algorithm for the Autoencoder and Canonical Correlation Analysis approach. It should be noted that the dimension of the latent spaces for each of the X^B and Y^B variables after the application of the encoders (p_2lat_x/p_2lat_y), the dimension of the latent space during the application of the CCA (d), and the bandwidth (p_2h) during the statistical matching phase need to be optimised. This analysis will begin with the optimal values from an RsMSE perspective and then from a Cramer-Von Mises statistical point of view, before elucidating the differences between the two.

Primarily, the values of the hyperparameters which optimise the algorithm's performance from the Root standardised Mean Squared Error perspective are illustrated in the aforementioned table, extracted from section 4.2.2.1. The optimal hyperparameter values are highlighted in red.

d	p_2_lat_x/p_2_lat_y	p_2h2	RsMSE	CVM_C_NC
3	14	0.55	0.96855524	0.04453943
3	15	0.60	0.96867608	0.04466020
4	14	0.65	0.96865352	0.04433593

***Table 22**: Hyperparameter values maximising performance of the A-CCA method according to RsMSE (best values highlighted in red)*

The data in this table allows us to conclude that in order to optimise the performance of the machine learning algorithm for the A-CCA approach in the RsMSE perspective, the dimension of the latent spaces (p_2lat_x/p_2lat_y) after applying the encoders should be 14, the dimension of the latent space d when applying the CCA should be 3, and the bandwidth hyperparameter (p_2h) required during the statistical matching phase should be 0.55. In this instance, the minimum value of the RsMSE is 0.96855524, which is less than one. A value less than one indicates a prediction error that is less than the standard deviation of the data, which demonstrates a good fit of the model to the data. In this instance, the objective of minimising RsMSE should be noted, along with the value of the Cramer-Von Mises statistic, which is equivalent to 0.04453943.

Furthermore, the values that optimise the performance of the A-CCA algorithm from the perspective of the Cramer-Von Mises statistic are presented in the table below, as detailed in section 4.2.2.1. The optimal combination of values from the three presented is highlighted in red.

d	p_2_lat_x/p_2_lat_y	p_2h	RsMSE	CVM_C_NC
1	6	0.65	0.97644962	0.04347295
2	6	0.70	0.97550459	0.04380251
1	8	0.70	0.97826579	0.04457400

***Table 23**: Hyperparameter values maximising performance of the A-CCA method according to CVM_C_NC (best values highlighted in red)*

An examination of the table indicates that in order to optimise the performance of the machine learning algorithm for the A-CCA approach from the perspective of the Cramer-Von Mises statistic, the dimension of the latent space p_2lat_x/p_2lat_y after the application of the encoders should be 6, the dimension of the latent space d after the application of the CCA should be 1, and the bandwidth hyperparameter p_2h required during the statistical matching phase should be 0.65. In accordance with the aforementioned conditions, the minimum value of the Cramer-Von Mises statistic is 0.04347295. The value of CVM_C_NC indicates that the data structure and the dependencies between variables have been effectively preserved. The empirical and real cumulative distribution functions exhibit a high degree of similarity, and the model demonstrates a satisfactory performance in terms of the Cramer-Von Mises statistic. In this instance, the RsMSE is 0.97644962.

These results show a difference in the value of the hyperparameters depending on what we are trying to minimise. In fact, we find that the dimension of the different latent spaces ($p_2lat_x/p_2lat_y \; and \; d$) must be higher when minimising the RsMSE than when trying to optimise from the point of view of the Cramer-Von Mises statistic.

First, it makes sense to have larger dimensions of p_2lat_x/p_2lat_y when we are trying to minimise the RsMSE than when we are trying to minimise the Cramer-Von Mises statistic. This is because when the autoencoder is used, the chosen dimensions do not help to describe the dependencies. Therefore, minimising the Cramer-Von Mises statistic does not imply obtaining large dimensions. However, for the RsMSE, larger dimensions chosen by the autoencoder allow a better reproduction of $X^B \; and \; Y^B$ individually, because the larger the dimension, the closer it is to the number of variables to be predicted (15 in this case). Therefore, increasing the dimension of p_2lat_x/p_2lat_y will not affect the CVM_C_NC, but will favour the RsMSE.

A second explanation can be provided concerning the relationship between the dimension of the latent spaces after the application of the encoders (p_2lat_x/p_2lat_y) and the latent space d in the context of the CCA approach. In the case of a large representation space for both $X^B \; and \; Y^B$ after the application of encoders φ_X^B and φ_Y^B, this implies that the resulting representations are of a high dimensionality. In order for CCA to be effective in such cases, the dimension of the latent space d, must be sufficiently large to capture the nuances and variations present in the high-dimensional representations. If the latent space is insufficiently large, it may be unable to capture the full complexity of the data after encoding, thereby limiting the effectiveness of the CCA technique in maximising correlations between representations of $X^B \; and \; Y^B$. A larger latent space provides sufficient capacity to model and maximise canonical correlations between information-rich representations of the data, thereby ensuring better model performance.

Subsequently, with regard to the bandwidth p_2h, the value of this hyperparameter must be lower when the objective is to minimise the RsMSE than when the objective is to minimise the Cramer-Von Mises statistic. Indeed, given that the A-CCA approach is primarily concerned with individual reproduction of $X^B \; and \; Y^B$, a reduction in p_2h is an effective means of minimising the RsMSE. Conversely, a greater value of this hyperparameter is necessary to minimise the Cramer-Von Mises statistic in order to reproduce the dependencies as accurately as possible, although this is not the actual outcome.

Nevertheless, it can be seen that the RsMSE and CVM_C_NC values exhibit only slight differences between the two cases. Indeed, the RsMSE reaches its minimum value of 0.9686 and is 0.9764 when the CVM_C_NC is optimised (a difference of 0.0078). In both cases, the metric is less than one, which demonstrates that the models perform well, as the prediction errors are less than the standard deviation of the real data. Similarly, the minimum value of CVM_C_NC is 0.04347, while its value when the RsMSE is optimised is 0.04454 (a difference of 0.00107), demonstrating that the two variables exhibit an almost identical level of performance.

Next, we identified a noteworthy point in our analysis of the relationship between the dimensions of the latent spaces, (p_2lat_x/p_2lat_y), and the two-performance metrics for each dimension of the latent space d. Indeed, an examination of these graphs (see figure 13) reveals a greater degree of fluctuations, although the scale is limited, the oscillatory patterns are discernible. These fluctuations may be attributed to the random selection of certain hyperparameters, including the learning rate and the number of units in the hidden layers of the neural network in the A-CCA approach. Consequently, these randomly selected values may not represent the optimal solution. One potential solution to this limitation would be to increase the number of random searches, that is, to test a greater number of hyperparameter combinations, thereby increasing the range of possibilities and the probability of identifying the optimal values.

A greater number of hyperparameter configurations can be assessed, thus increasing the coverage of the hyperparameter space, and enabling more effective exploration of potential model architectures. This reduces the probability of failing to identify hyperparameter combinations that could yield optimal performance. Consequently, this could result in a reduction in the fluctuations observed in the performance metrics and an overall enhancement in the stability and performance of the model.

In conclusion, in order to optimise the performance of the algorithm for the Autoencoder and Canonical Correlation Analysis approach, it is necessary either to prioritise the optimisation of one of the two performance metrics, or to identify a balance between the two metrics. As previously stated, the RsMSE and Cramer-Von Mises metrics evaluate disparate aspects of a prediction model's performance.

Therefore, if the primary objective is to optimise the RsMSE, the values of the hyperparameters that optimise the algorithm are as follows: $d = 3$; $p_2h = 0.55$; $(p_2lat_x/p_2lat_y) = 14$. Should the primary objective be the optimisation of the Cramer-von Mises statistic whilst maintaining the data structure, the optimal values are as follows: $d = 1$; $p_2h = 0.65$; $(p_2lat_x/p_2lat_y) = 6$.

Ultimately, in an effort to enhance the algorithm's performance by considering both metrics concurrently, it is essential to strike a balance between the precision of predictions (minimising errors)

and the preservation of inter-variable dependencies (minimising the discrepancy between empirical and predicted distributions).

A final observation regarding the two methods reveals that the KCCA approach is able to produce lower values for the Cramer-Von Mises statistic than the A-CCA method, while the A-CCA technique produces a slightly lower RsMSE than the KCCA approach. This phenomenon can be explained by the different objectives of these two approaches. As previously stated, the KCCA approach is more efficacious in identifying the dependencies between variables due to its non-linear nature and capacity for high-dimensional analysis. In contrast, the A-CCA method is primarily concerned with the individual reproduction of the values of $X^B \; and \; Y^B$, whereby the reproduction error is minimised.

6 Conclusion

The objective of this chapter is to present a synthesis of the principal findings of this study, to delineate the constraints encountered, and to propose avenues for future research. Firstly, we will present a summary of the research carried out and the main findings. Subsequently, we will examine the constraints of our research endeavour, offering a critical analysis and identifying avenues for enhancement. In conclusion, we will propose avenues for future research, indicating areas and questions that could benefit from further investigation.

6.1 Summary of the study

The objective of this thesis was to optimise the bandwidths, and the dimension of latent spaces present in machine learning algorithms for the Kernel Canonical Correlation Analysis and Autoencoder and Canonical Correlation Analysis methods. Both approaches are statistical matching techniques, a crucial area in modern data analysis. In light of the growing volume of data from a multitude of sources, the necessity for statistical matching is paramount to enable the coherent and meaningful study of disparate subjects. This technique is particularly useful for combining and comparing heterogeneous data sets, thereby facilitating the drawing of more robust and relevant conclusions.

In this thesis, we initially conducted a review of the existing literature on the subject matter, after which we proceeded to propose a methodology for addressing the research question. This outlined the calculations of the various methods and identified the hyperparameters to be optimised. Furthermore, the database was constructed in R software and divided into a training set comprising $4{,}000$ rows and a validation set consisting of $1{,}000$ rows. Both sets contained 30 variables (columns), including 15 common and 15 non-common. The hyperparameters of the two methods were studied using the Grid Search strategy, and the performance of the different models was evaluated using two performance metrics: the Cramer-Von Mises statistic and the Root standardised Mean Squared Error (RsMSE).

On the one hand, the Kernel Canonical Correlation Analysis (KCCA) approach entails the utilisation of kernel techniques for the transformation of data into higher-dimensional Hilbert spaces. Subsequently, the CCA technique is employed in these spaces to identify the data projections where their canonical correlation is maximised. On the other hand, the Autoencoder and Canonical Correlation Analysis (A-CCA) method seeks to utilise a distinct encoder for each dataset $X^B \; and \; Y^B$, prior to the application of the CCA technique. The primary distinction between the two methods lies in their respective objectives. The KCCA approach is more effective at identifying dependencies between data, whereas the A-CCA method is primarily concerned with enhancing the individual reproduction of $X^B \; and \; Y^B$. It can be seen that the KCCA loss function is designed to maximise the relations between $X^B \; and \; Y^B$, whereas the A-CCA loss function is intended to minimise the reproduction error.

In this study, two performance metrics were employed: the RsMSE and the Cramer-Von Mises statistic. The Root standardised Mean Squared Error (RsMSE) is a metric that gauges the precision of prediction errors. Accordingly, optimising performance in accordance with the RsMSE entails reducing the deviations between the predicted values ($\widehat{Y^A}$) and the actual values (Y^A). Conversely, the Cramer-Von Mises statistic (CVM_C_NC) compares the cumulative distribution of the predicted and actual values, with the objective of minimising the distance between the two distributions. This approach preserves

the dependencies between variables and ensures that the overall distributions of predicted and actual values are similar. The primary distinction between these two metrics is that the RsMSE emphasises the precision of individual predictions, whereas the Cramer-Von Mises statistic evaluates the overall concordance of the distributions.

The findings of the KCCA approach demonstrate that the value of the hyperparameters has a significant influence on both performance metrics, although the manner in which this occurs differs. Indeed, for this approach, a low value for the bandwidth p_2h minimises the Cramer-Von Mises statistic, whereas a higher value of this hyperparameter optimises the RsMSE. With respect to the dimension of the latent space d, a higher dimension is associated with a favourable outcome for the Cramer-Von Mises statistic, while the RsMSE initially exhibits a decline as d increases, followed by an ascent. The discrepancy in the evolution of the two metrics can be attributed to a divergence in their underlying objectives. The RsMSE prioritises the accuracy of predictions, whereas the Cramer-Von Mises statistic emphasises the conservation of dependencies.

Furthermore, a higher latent space d enables more effective capture of the dependencies between the data, as the neighbourhood is more accurately selected, thereby enhancing the Cramer-Von Mises statistic. Conversely, a lower dimension d facilitates the model's generalisation, which in turn reduces the RsMSE. Finally, a lower value of p_2h minimises the Cramer-Von Mises statistic due to the increased sensitivity of the kernel function, which facilitates the capture of dependencies and thus enables the identification of similarities between the actual and predicted distributions of data. Conversely, a higher value results in a smoothing out of the variations, increasing the Cramer-Von Mises statistic but decreasing the RsMSE due to the enabling of generalisation and avoidance of overfitting.

In consideration of the aforementioned behaviours and relationships, it becomes evident that achieving optimal performance in both metrics simultaneously represents a challenging undertaking. In order to optimise the algorithm's performance, it is necessary to prioritise either the RsMSE or the Cramer-Von Mises statistic, or alternatively, to identify a compromise. If the objective is to minimise the RsMSE, a lower dimension of d and a higher p_2h are required. The optimal values are as follows: $d = 6$; $p_2h = 0.71$; $p_2h_x = 0.003$; $p_2h_y = 0.007$. In order to minimise the Cramer-Von Mises statistic, it is necessary to employ a higher dimension d and a lower p_2h. Thus, the optimal values are as follows: $= 7$; $p_2h = 0.01$; $p_2h_x = 0.001$; $p_2h_y = 0.05$. Ultimately, in order to optimise the algorithm in a manner that considers both metrics, it is essential to identify a balance between the precision of predictions and the maintenance of data dependencies.

In relation to the A-CCA approach, the findings also demonstrate that the hyperparameters have an impact on the performance of the models. It can be observed that larger dimensions of the latent spaces ($p_2lat_x, p_2lat_y \; and \; d$) tend to favour the RsMSE, whereas smaller dimensions favour the Cramer-Von Mises statistic. Consequently, a reduction in the value of p_2h is favourable for the RsMSE and has an adverse effect on the Cramer-Von Mises statistic. Once more, the two metrics do not exhibit a similar trend, which can be attributed to the distinct objectives of each.

It can be observed that larger dimensions of latent spaces (p_2lat_x/p_2lat_y) result in a reduction of the RsMSE, due to the fact that the loss function of the autoencoder is defined in terms of the reproduction error. Thus, it becomes evident that the larger the dimension of p_2lat_x/p_2lat_y, the smaller the error

and the lower the RsMSE. In contrast, the A-CCA approach is ineffective in capturing the dependencies between variables, which eliminates the necessity for a large dimension of p_2lat_x/p_2lat_y. Moreover, there is a relationship between the dimensions of p_2lat_x/p_2lat_y and the dimension of d. Indeed, if p_2lat_x/p_2lat_y are large, d must also be large in order to capture all the variations included in these different latent spaces, which will therefore minimise the RsMSE. Ultimately, a low value of p_2h is beneficial for the RsMSE, as the A-CCA approach prioritises the independent reproduction of $X^B\ and\ Y^B$. Therefore, this hyperparameter should not be excessively large. Conversely, to minimise the statistic, p_2h should be increased to better reproduce the dependencies.

In light of these considerations, it becomes evident that there is no straightforward approach to optimising the A-CCA algorithm by simultaneously minimising both metrics. However, our analysis revealed that the values of RsMSE and the Cramer-Von Mises statistic were strikingly similar when attempting to minimise one metric or the other. Thus, it would seem that there is merit in favouring either the RsMSE or the Cramer-von Mises statistic, without unduly compromising performance in the other metric. If the objective is to minimise the RsMSE, the latent space dimensions are higher, and the bandwidth is lower $d = 3$; $p_2h = 0.55$; $(p_2lat_x/p_2lat_y) = 14$. Conversely, if the objective is to optimise the statistics, the dimensions are lower and the p_2h is higher $d = 1$; $p_2h = 0.65$; $(p_2lat_x/p_2lat_y) = 6$. A final solution is to identify a compromise between prediction accuracy (RsMSE) and dependency conservation (Cramer-Von Mises statistic).

From a practical standpoint, this thesis proposes a hyperparameter optimisation method that can be transposed to different databases and analytical contexts. The flexibility of this method allows practitioners to apply it to a variety of datasets and algorithms, thereby providing a tool for enhancing the performance of the two methods under investigation in the context of statistical matching.

From a managerial standpoint, this research facilitates more nuanced examination of intricate interconnections between disparate data sources. This can assist managers in making more informed decisions based on more precise insights and significant correlations between key variables. A more nuanced comprehension of the interconnections between data can enhance business strategy, resource planning and market analysis, thereby facilitating more strategic and informed decision-making.

Finally, this work will facilitate more comprehensive data collection for the team developing these two algorithms, enabling them to gain deeper insights into their work, its quality, and consistency. The thesis will facilitate a more nuanced comprehension of the relationships between the diverse hyperparameters and the two-performance metrics under investigation.

6.2 Limitations et future research

This section will address the challenges and limitations encountered throughout the course of this research, examining their potential impact on the results. Additionally, suggestions will be put forth regarding avenues for future research.

Firstly, the optimisation of hyperparameters for the Kernel Canonical Correlation Analysis and Autoencoder and Canonical Correlation Analysis approaches is a computationally expensive process, necessitating significant execution times. To accelerate the process, which spanned approximately two

months, the codes were executed on six computers. The mean execution time for the KCCA method is between 16 and 24 hours, while the mean execution time for the A-CCA approach is between 10 and 18 hours (depending on the computers and their processing power). Furthermore, the aforementioned costs are amplified when larger databases are under study.

Secondly, the size of the database had to be reduced from the $30{,}000$ rows initially planned to $5{,}000$ rows for time reasons. In fact, execution took more than three days. With a reduced database, the predictions and results can be less accurate and less stable because they depend on the size of the sample, among other things.

In order to overcome these limitations, it may be beneficial to explore approaches to reduce the computational cost associated with hyperparameter optimisation for Kernel Canonical Correlation Analysis (KCCA) and Autoencoder and Canonical Correlation Analysis (A-CCA) methods. This would result in a reduction in the time required for performance optimisation and would facilitate the utilisation of larger databases, thereby enhancing the stability and robustness of the results.

Moreover, the hyperparameters optimised in this thesis are specific to the database under consideration and cannot be directly transferred to other contexts or datasets without undergoing the same optimisation process. Consequently, the optimisation process must be repeated for each new application, which is a time-consuming process.

Furthermore, this thesis has concentrated on specific hyperparameters of the second phase of the algorithms, leaving the remaining hyperparameters at their default values and neglecting phase one. It would therefore be beneficial for future research to explore the optimisation of unstudied hyperparameters in order to further improve model performance. It would also be beneficial to consider the integration of phase one, which deals with categorical variables, in order to ascertain its effect on the overall results. These steps would permit a more profound analysis, and the optimisation of the methodologies employed, as well as the discovery of the impact of these hyperparameters on the results.

As previously stated in the discussion section, a limitation of the A-CCA approach has been identified, specifically in relation to the utilisation of the random search method. The method in question employs a random selection process for hyperparameters, which may impede the ability to reach optimal values and, consequently, impact the quality of the resulting outcomes. One avenue for future research would be to increase the number of hyperparameter combinations tested by random search. By investigating a more expansive search space, it would be feasible to ascertain whether a notable expansion in the number of trials would result in enhanced performance and more resilient outcomes, through the more expedient identification of the optimal values of the hyperparameters.

7 Appendices

***Appendix A**: Code for building the database*

```r
rm(list=ls())
set.seed(123)

dt <- matrix(NA, ncol = 30, nrow = 5000)

dt[,1] <- rbinom(5000, size = 1, prob = 0.5)

dt[,2] <- rnorm(5000, mean = 0, sd = 1)

for (colonne in 3:5) {
  dt[, colonne] <- rnorm(5000, mean = 0.5 * dt[, colonne - 1], sd = 1)
}

for (i in 1:5000) {
  if(dt[i,4]>0){
    dt[i,5] <- 1
  } else{
    dt[i,5] <- 0
  }
}

dt[,6] <- rnorm(5000, mean = dt[,4]+ 0.8*dt[,3]-dt[,2], sd = 1)

for (i in 7:9){
  for(j in 1:5000){
    if(dt[j,i-1]>0){
      dt[j,i] <- rbinom(1, size = 1, prob = 0.8)
    } else{
      dt[j,i] <- rbinom(1, size = 1, prob = 0.2)
    }
  }
}

for (colonne in 10:12) {
  dt[, colonne] <- rnorm(5000, mean = 0.5 * dt[, colonne - 1]+0.5*dt[, colonne - 2], sd = 1)
}

for (i in 13:15){
  for(j in 1:5000){
    if(dt[j,i-3]>0){
      dt[j,i] <- 0
    }else{
      dt[j,i] <- 1
    }
  }
}

for (i in 16:19){
  for(j in 1:5000){
    if(dt[j,i-1]>0){
      dt[j,i] <- rbinom(1, size = 1, prob = 0.45)
    } else{
      dt[j,i] <- rbinom(1, size = 1, prob = 0.55)
    }
  }}

for(i in 20:24){
  dt[,i] <- rnorm(5000, mean = 0.2*(dt[,i-7]+dt[,i-6]+dt[,i-5]+dt[,i-4]+dt[,i-3]))
}

for (i in 25:28){
  for(j in 1:5000){
    if(dt[j,i-9]+dt[j,i-13]+dt[j,i-2]>1.1){
      dt[j,i] <- 1
    }else{
      dt[j,i] <- 0
    }
  }
}

for(i in 29:30){
  dt[,i] <- rnorm(5000, mean = 0.25*(dt[,i-1]+dt[,i-3]+dt[,i-5]+dt[,i-7]))
}

dt <- data.frame(dt)
colnames(dt)<-c("X1", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9", "X10",  "X11",  "X12",  "X13",  "X14",
                "Y1", "Y2", "Y3", "Y4", "Y5", "Y6", "Y7", "Y8", "Y9", "Y10",  "Y11",  "Y12",  "Y13",  "Y14",

XA <- dt[1:1000,1:15]
XB <- dt[1001:5000,1:15]
YA <- dt[1:1000,16:30]
YB <- dt[1001:5000,16:30]
```

P_2h / d	0.0051								0.0076								0.01							
	Obj_fun	RsMSE	CVM_C_NC	CVM_NCNC	CVM_all	CVM_all2	p_2hx	p_2hy	Obj_fun	RsMSE	CVM_C_NC	CVM_NCNC	CVM_all	CVM_all2	p_2hx	p_2hy	Obj_fun	RsMSE	CVM_C_NC	CVM_NCNC	CVM_all	CVM_all2	p_2hx	p_2hy
1	0.97256	0.97905	0.04336	0.08114	0.0000011	0.00090	0.01	0.71	0.97445	0.97729	0.04399	0.08241	0.0000011	0.00093	0.01	0.71	0.97347	0.97644	0.04434	0.08313	0.0000012	0.00096	0.01	0.71
2	0.99047	0.99323	0.04021	0.07469	0.0000011	0.00077	0.01	0.01	0.98333	0.98575	0.04147	0.07726	0.0000011	0.00085	0.01	0.01	0.97903	0.98232	0.04213	0.07861	0.0000011	0.00087	0.01	0.01
3	1.02472	1.03296	0.04205	0.07811	0.0000011	0.00074	1.06	0.01	1.02217	1.03290	0.04205	0.07812	0.0000011	0.00074	1.06	0.01	1.02214	1.03267	0.04206	0.07813	0.0000011	0.00124	1.06	0.01
4	1.02769	1.02946	0.04188	0.07791	0.0000011	0.00124	1.06	0.01	1.02085	1.02946	0.04188	0.07791	0.0000011	0.00124	1.06	0.01	1.02724	1.02946	0.04188	0.07791	0.0000011	0.00124	1.06	0.01
5	1.01754	1.01459	0.03745	0.06917	0.0000011	0.00063	0.01	1.06	1.01673	1.01459	0.03745	0.06917	0.0000011	0.00063	0.01	1.06	1.01804	1.01459	0.03745	0.06917	0.0000011	0.00063	0.01	1.06
6	1.00266	1.00208	0.03862	0.07159	0.0000012	0.00069	0.01	0.01	1.00280	1.00208	0.03862	0.07159	0.0000012	0.00069	0.01	0.01	1.00267	1.00208	0.03862	0.07159	0.0000012	0.00069	0.01	0.01
7	0.99185	0.99488	0.03970	0.07378	0.0000012	0.00075	0.01	0.01	0.99346	0.99488	0.03970	0.07378	0.0000012	0.00075	0.01	0.01	0.99285	0.99488	0.03970	0.07378	0.0000012	0.00075	0.01	0.01

0.11									0.21								0.31							
Obj_fun	RsMSE	CVM_C_NC	CVM_NCNC	CVM_all	CVM_all2	p_2hx	p_2hy	Obj_fun	RsMSE	CVM_C_NC	CVM_NCNC	CVM_all	CVM_all2	p_2hx	p_2hy	Obj_fun	RsMSE	CVM_C_NC	CVM_NCNC	CVM_all	CVM_all2	p_2hx	p_2hy
0.97372	0.97738	0.04595	0.08642	0.0000011	0.00099	0.01	1.06	0.97633	0.97960	0.04652	0.08761	0.0000011	0.00104	0.01	1.06	0.98189	0.98405	0.04727	0.08924	0.0000011	0.00119	0.01	1.06
0.96832	0.97324	0.04535	0.08522	0.0000011	0.00096	0.01	0.01	0.96992	0.97493	0.04574	0.08603	0.0000011	0.00097	0.01	0.71	0.97195	0.97730	0.04602	0.08659	0.0000011	0.00099	0.01	0.36
0.96780	0.97245	0.04475	0.08396	0.0000011	0.00095	0.01	0.71	0.96668	0.97242	0.04523	0.08495	0.0000011	0.00097	0.01	1.06	0.96730	0.97342	0.04546	0.08543	0.0000011	0.00097	0.01	1.06
0.97256	0.97475	0.04297	0.08035	0.0000011	0.00093	0.01	0.71	0.96690	0.97152	0.04434	0.08317	0.0000011	0.00097	0.01	0.71	0.96600	0.97156	0.04469	0.08384	0.0000011	0.00096	0.01	0.71
0.97762	0.97952	0.04157	0.07751	0.0000014	0.00091	0.01	0.01	0.96752	0.97121	0.04377	0.08203	0.0000011	0.00094	0.01	0.01	0.96545	0.97035	0.04441	0.08334	0.0000011	0.00092	0.01	0.01
0.98464	0.98645	0.04043	0.07524	0.0000012	0.00082	0.01	0.01	0.97080	0.97498	0.04271	0.07988	0.0000012	0.00093	0.01	0.01	0.96684	0.97112	0.04368	0.08185	0.0000011	0.00096	0.01	0.01
0.98683	0.98862	0.04029	0.07498	0.0000012	0.00080	0.01	0.01	0.97600	0.98000	0.04184	0.07812	0.0000014	0.00090	0.01	0.01	0.96958	0.97450	0.04299	0.08045	0.0000012	0.00096	0.01	0.01

0.41							
Obj_fun	RsMSE	CVM_C_NC	CVM_NCNC	CVM_all	CVM_all2	p_2hx	p_2hy
0.98740	0.98877	0.04780	0.09031	0.0000011	0.00120	0.01	1.06
0.97287	0.97792	0.04620	0.08697	0.0000011	0.00102	0.01	0.36
0.96774	0.97377	0.04561	0.08577	0.0000011	0.00100	0.01	1.06
0.96620	0.97168	0.04486	0.08420	0.0000011	0.00095	0.01	0.71
0.96532	0.97059	0.04468	0.08387	0.0000011	0.00092	0.01	0.01
0.96492	0.96987	0.04416	0.08281	0.0000011	0.00094	0.01	0.01
0.96629	0.97169	0.04365	0.08179	0.0000011	0.00096	0.01	0.01

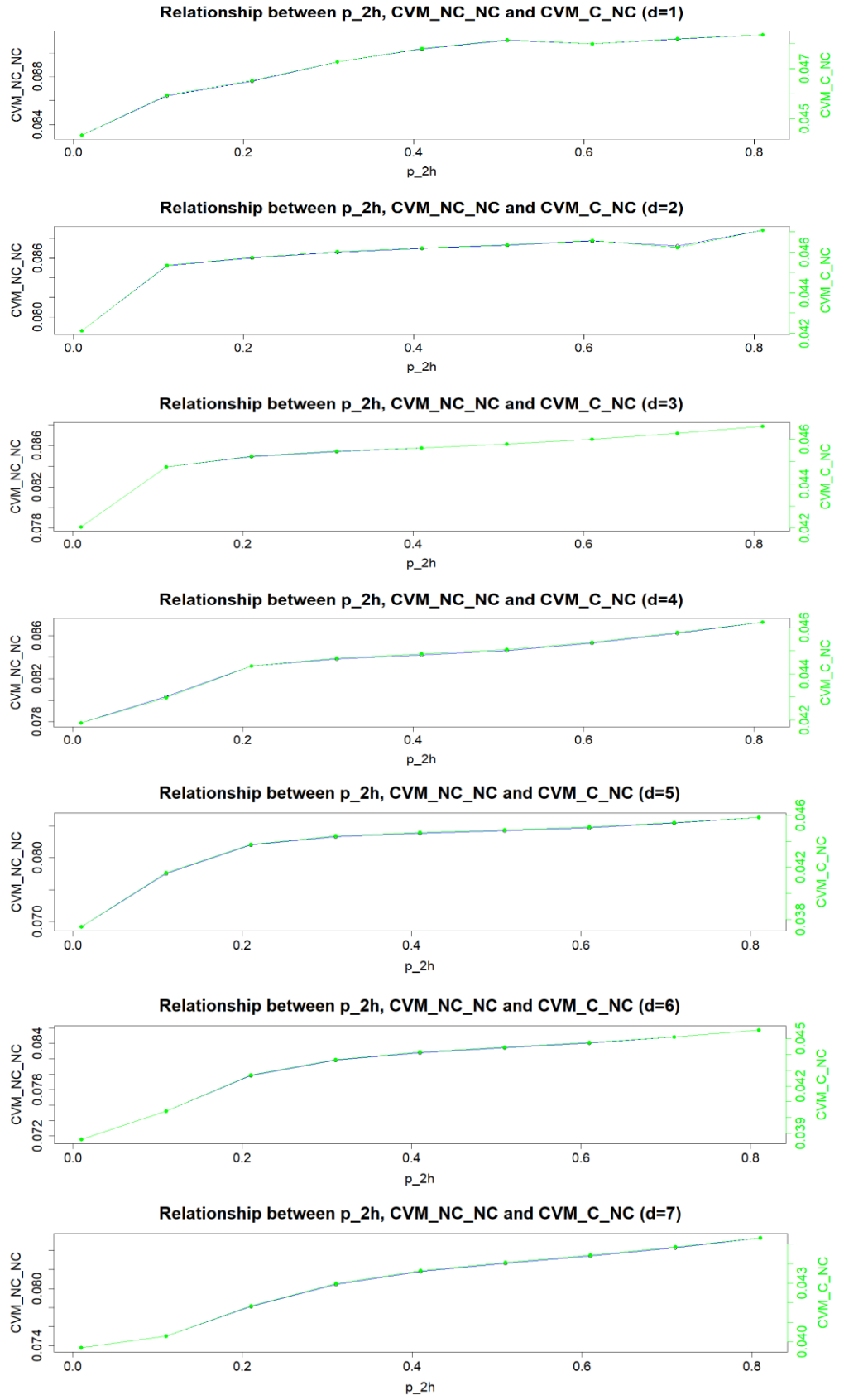
0.51							
Obj_fun	RsMSE	CVM_C_NC	CVM_NCNC	CVM_all	CVM_all2	p_2hx	p_2hy
0.99089	0.99213	0.04815	0.09105	0.0000011	0.00117	0.01	1.06
0.97396	0.97834	0.04636	0.08732	0.0000011	0.00104	0.01	0.36
0.96822	0.97368	0.04578	0.08615	0.0000011	0.00101	0.01	1.06
0.96575	0.97159	0.04505	0.08459	0.0000011	0.00096	0.01	0.71
0.96527	0.97086	0.04487	0.08425	0.0000011	0.00094	0.01	0.01
0.96428	0.96952	0.04447	0.08344	0.0000011	0.00094	0.01	0.01
0.96503	0.97052	0.04406	0.08264	0.0000011	0.00095	0.01	0.01

0.61							
Obj_fun	RsMSE	CVM_C_NC	CVM_NCNC	CVM_all	CVM_all2	p_2hx	p_2hy
0.99258	0.99450	0.04800	0.09072	0.0000011	0.00115	0.36	1.06
0.97508	0.97898	0.04656	0.08774	0.0000011	0.00107	0.01	0.36
0.96900	0.97413	0.04600	0.86607	0.0000011	0.00104	0.01	0.01
0.96638	0.97154	0.04537	0.08530	0.0000011	0.00099	0.01	0.71
0.96541	0.97102	0.04510	0.08475	0.0000011	0.00097	0.01	0.01
0.96447	0.96949	0.04477	0.08407	0.0000011	0.00096	0.01	0.01
0.96503	0.97002	0.04444	0.08341	0.0000011	0.00097	0.01	0.01

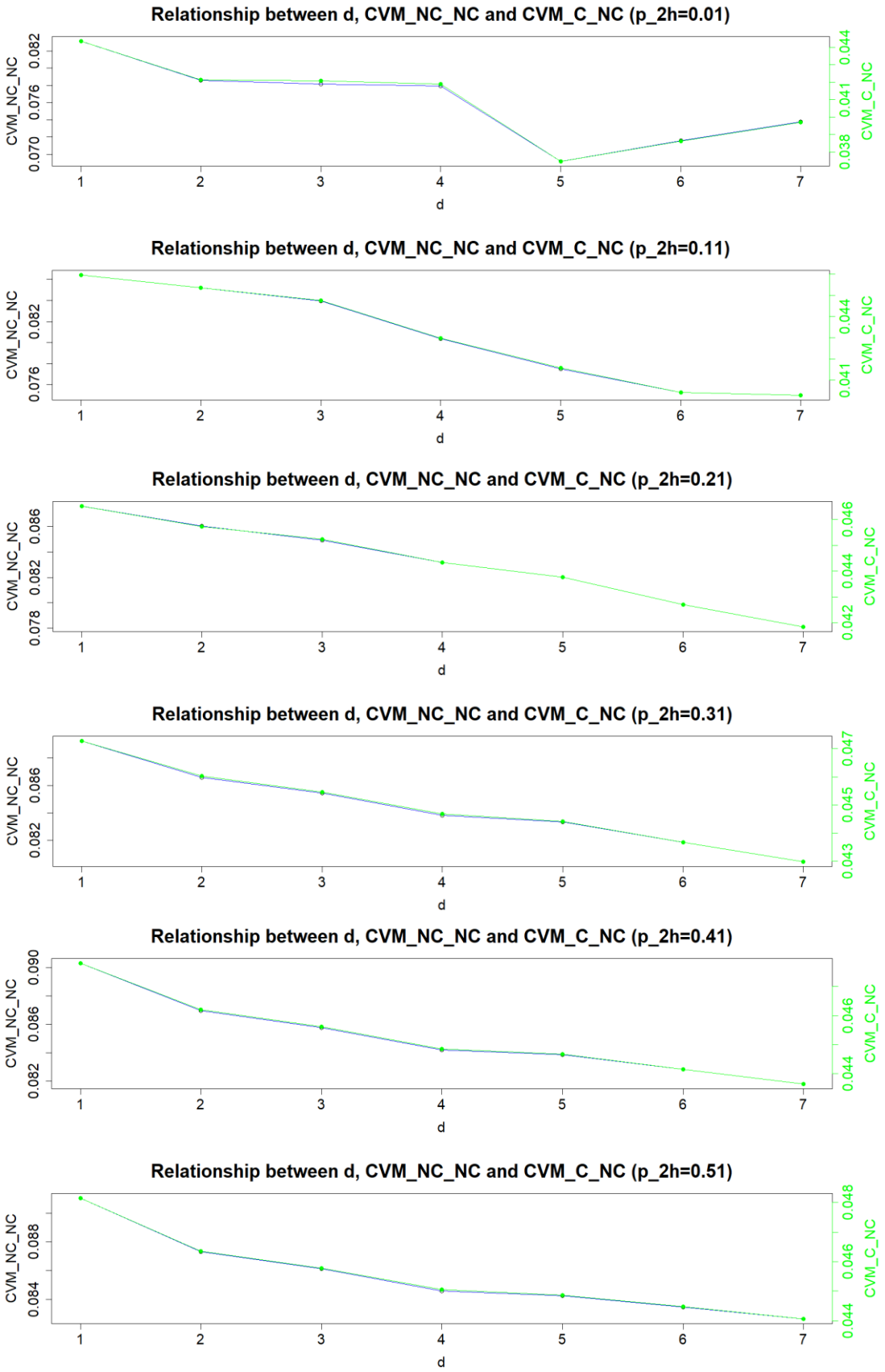
0.71							
Obj_fun	RsMSE	CVM_C_NC	CVM_NCNC	CVM_all	CVM_all2	p_2hx	p_2hy
0.99401	0.99558	0.04820	0.09114	0.0000011	0.00114	0.71	0.36
0.97011	0.97415	0.04623	0.08722	0.0000011	0.00113	0.01	0.36
0.97032	0.97490	0.04626	0.08718	0.0000011	0.00105	0.01	0.01
0.96793	0.97229	0.04578	0.08622	0.0000011	0.00104	0.01	0.71
0.96609	0.97144	0.04542	0.08547	0.0000011	0.00101	0.01	0.01
0.96475	0.96974	0.04513	0.08483	0.0000011	0.00099	0.01	0.01
0.96509	0.97003	0.04485	0.08430	0.0000011	0.00100	0.01	0.01

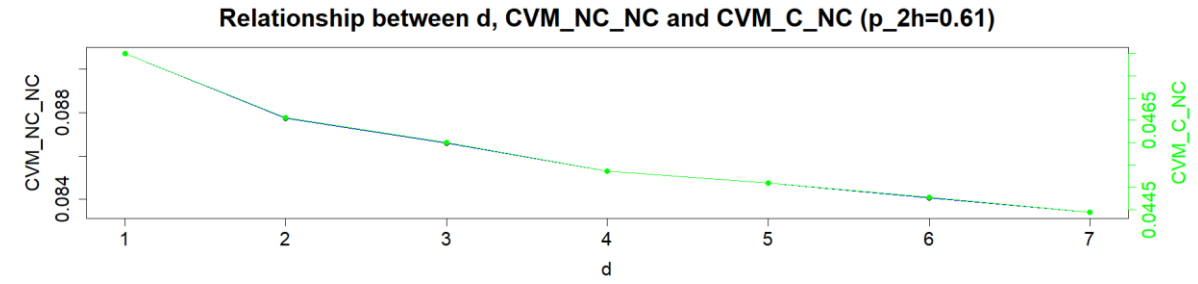
0.81							
Obj_fun	RsMSE	CVM_C_NC	CVM_NCNC	CVM_all	CVM_all2	p_2hx	p_2hy
0.99517	0.99645	0.04836	0.09149	0.0000011	0.00114	0.36	1.06
0.97863	0.98166	0.04708	0.08883	0.0000011	0.00116	0.01	0.36
0.97273	0.97662	0.04658	0.08788	0.0000011	0.00108	0.01	0.01
0.97011	0.97415	0.04623	0.08722	0.0000011	0.00113	0.01	0.71
0.96779	0.97263	0.04582	0.08637	0.0000011	0.00106	0.01	0.01
0.96609	0.97064	0.04555	0.08574	0.0000011	0.00104	0.01	0.01
0.96632	0.97078	0.04532	0.08532	0.0000011	0.00105	0.01	0.01

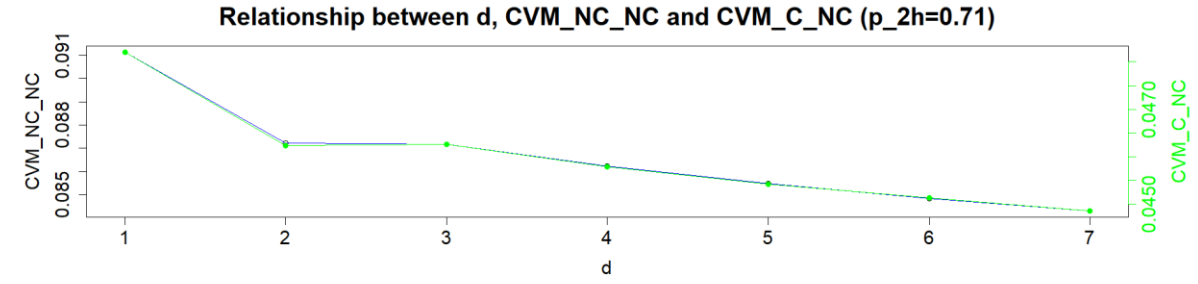
***Appendix C**: Relationship between CVM_C_NC and CVM_NC_NC as a function of the value of p_2h for each value of d for the KCCA approach*

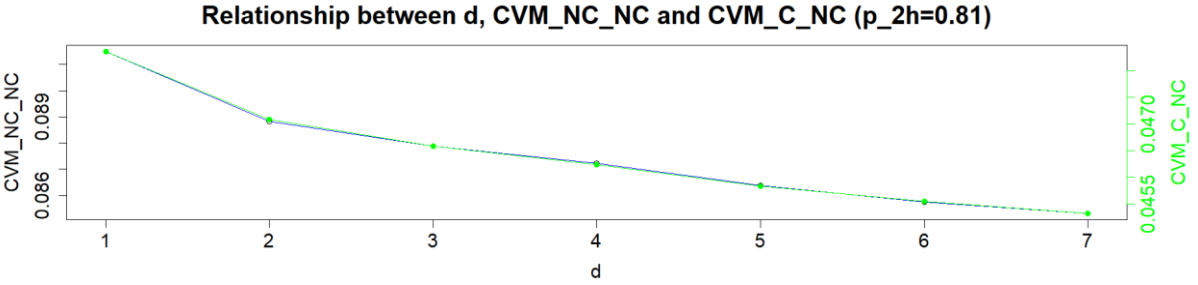


***Appendix D**: Relationship between CVM_C_NC and CVM_NC_NC as a function of the dimension d for each value of p_2h for the KCCA approach*

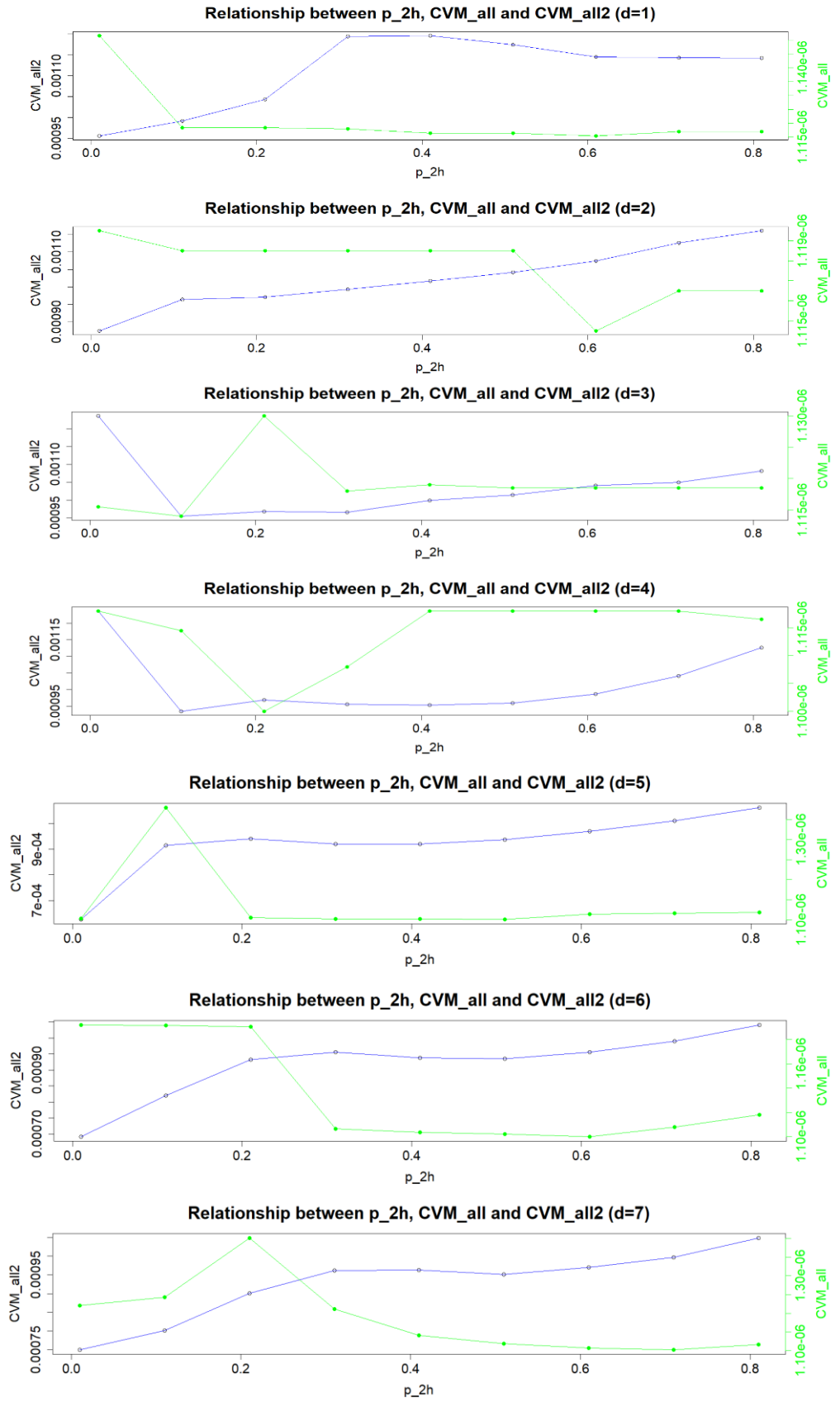



Relationship between d, CVM_NC_NC and CVM_C_NC (p_2h=0.61)
CVM_NC_NC
0.084
0.088
CVM_C_NC
0.0445
0.0465
1
2
3
4
5
6
7
d

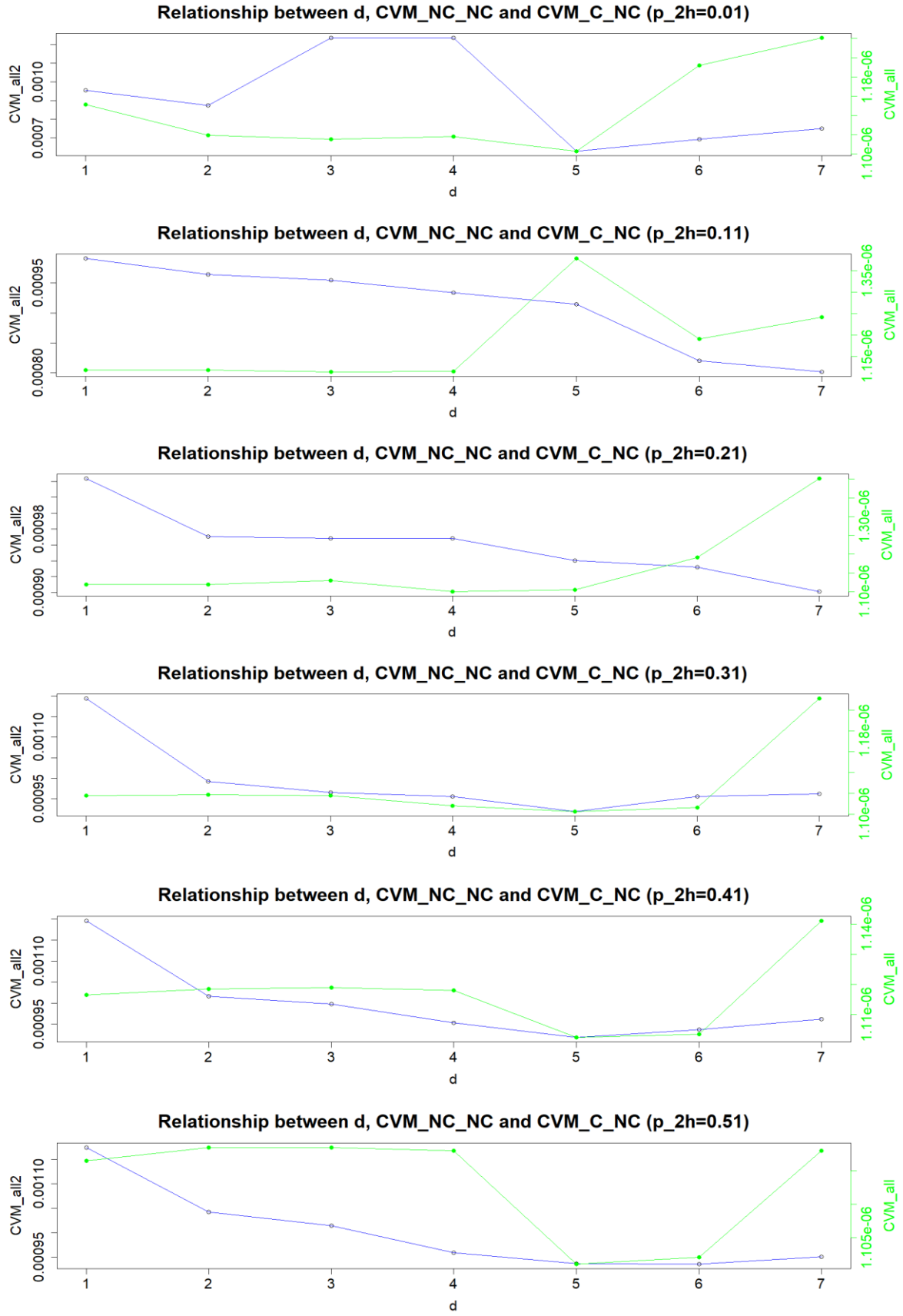

Relationship between d, CVM_NC_NC and CVM_C_NC (p_2h=0.71)
CVM_NC_NC
0.085
0.088
0.091
CVM_C_NC
0.0450
0.0470
1
2
3
4
5
6
7
d

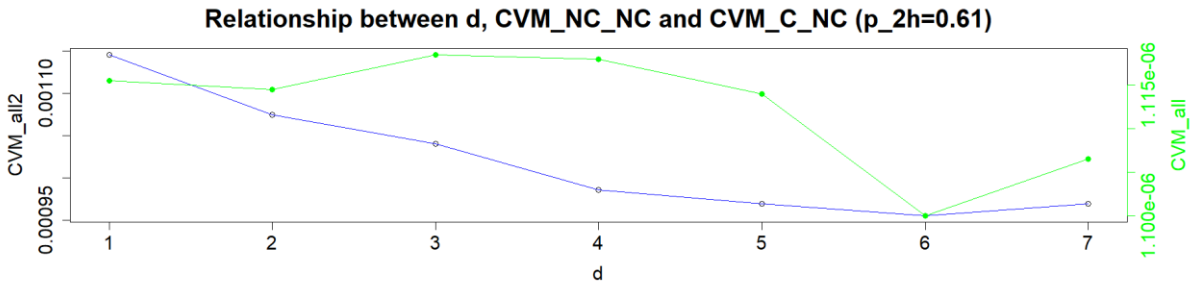

Relationship between d, CVM_NC_NC and CVM_C_NC (p_2h=0.81)
CVM_NC_NC
0.086
0.089
CVM_C_NC
0.0455
0.0470
1
2
3
4
5
6
7
d

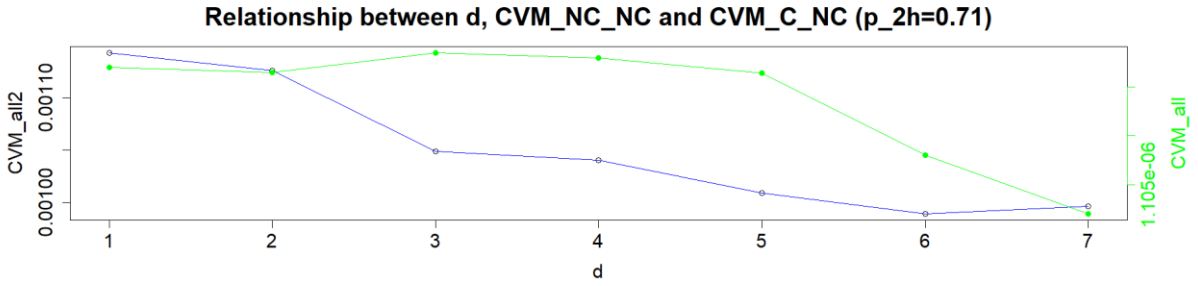
***Appendix E**: Relationship between CVM_{all}, CVM_{all2}, p_2h for each dimension d for the KCCA approach*

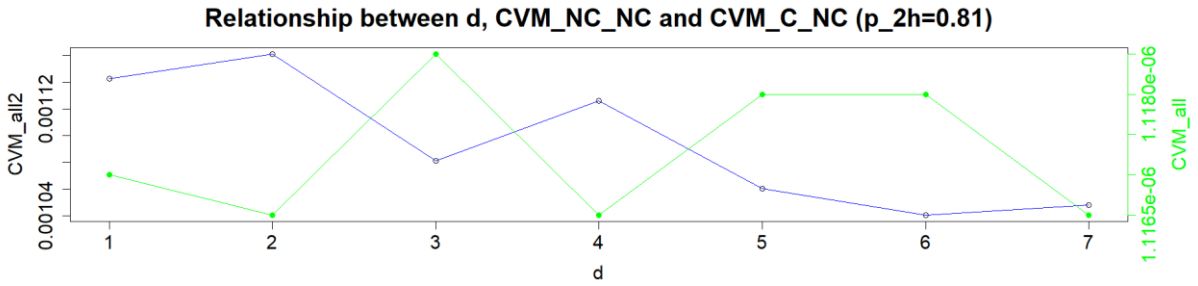


***Appendix F**: Relationship between CVM_{all}, CVM_{all2}, d for each value of p_2h for the KCCA approach*




Relationship between d, CVM_NC_NC and CVM_C_NC (p_2h=0.61)
CVM_all2
0.00095
0.00110
CVM_all
1.100e-06
1.115e-06
1
2
3
4
5
6
7
d


Relationship between d, CVM_NC_NC and CVM_C_NC (p_2h=0.71)
CVM_all2
0.00100
0.00110
CVM_all
1.105e-06
1
2
3
4
5
6
7
d


Relationship between d, CVM_NC_NC and CVM_C_NC (p_2h=0.81)
CVM_all2
0.00104
0.00112
CVM_all
1.1165e-06
1.1180e-06
1
2
3
4
5
6
7
d

***Appendix G**: Results of the evaluation of the cross-validation technique performance for the KCCA approach*

The initial values taken from the table in Appendix B are highlighted in blue, while yellow shows the optimum values after minimisation of the RsMSE by modifying the hyperparameters p_2h_x and p_2h_y.

d = 1 & p_2h = 0,21		
p_2hx / p_2hy	RsMSE	CVM_C_NC
0,01/1,06	0.97959853	0.04652169
0,01/1,2	0.97939601	0.04647650
0,01/1,25	0.97936528	0.04646764
0,01/1,3	0.97934960	0.04646267
0,01/1,35	0.97934638	0.04645988
0,01/1,4	0.97935373	0.04645758
0,015/1,35	0.97945957	0.04640783
0,0095/1,35	0.97934995	0.04646613

d = 2 & p_2h = 0,21		
p_2hx / p_2hy	RsMSE	CVM_C_NC
0,01/0,71	0.97492802	0.04573680
0,01/0,5	0.97564616	0.04573617
0,01/0,8	0.97476290	0.04574746
0,01/0,85	0.97472058	0.04575631
0,01/0,90	0.97471250	0.04576805
0,01/0,95	0.97473012	0.04577443
0,01/1	0.97685425	0.04573831
0,009/0,90	0.97792948	0.04582932
0,0095/0,9	0.97468741	0.04576587
0,015/0,9	0.97497639	0.04578141
0,02/0,90	0.97523786	0.04580894
0,025/0,9	0.97547736	0.04583957
0,03/0,9	0.97570289	0.04585215

d = 2 & p_2h = 0,51		
p_2hx / p_2hy	RsMSE	CVM_C_NC
0,01/0,36	0.9783367	0.0463578
0,01/0,3	0.9783831	0.0463594
0,01/0,4	0.9783031	0.0463594
0,01/0,45	0.9782734	0.0463638
0,01/0,5	0.9782580	0.0463646
0,01/0,55	0.9782576	0.0463652
0,01/0,6	0.9782721	0.0463711
0,01/0,65	0.9783004	0.0463715
0,0011/0,55	0.9783835	0.0463663
0,009/0,55	0.9782310	0.0463645
0,006/0,55	0.9781495	0.0463657
0,005/0,55	0.9781235	0.0463663
0,004/0,55	0.9781002	0.0463662
0,003/0,55	0.9780837	0.0463655
0,002/0,55	0.9780896	0.0463687
0,001/0,55	0.9788219	0.0463685
0,0005/0,55	0.9785962	0.0463997

***Appendix H**: Results of the optimisation of p_2h_x and p_2h_y by minimising the RsMSE*

The initial values taken from the table in Appendix B are highlighted in blue, while yellow shows the optimum values after minimisation of the RsMSE by modifying the hyperparameters p_2h_x and p_2h_y.

d = 6 & p_2h = 0.51		
p_2hx / p_2hy	RsMSE	CVM_C_NC
0.01/0.01	0.96952296	0.04447152
0.01/0.02	0.96953468	0.044491655
0.01/0.0095	0.96952521	0.044466678
0.01/0.009	0.96952883	0.04445931
0.011/0.01	0.96954639	0.044469715
0.0095/0.01	0.96952830	0.044470447
0.009/0.01	0.96953570	0.044468374
0.0085/0.01	0.96953693	0.044467759

d = 6 & p_2h = 0.61		
p_2hx / p_2hy	RsMSE	CVM_C_NC
0.01/0.01	0.9694925	0.04477321
0.01/0.02	0.9696207	0.04477645
0.01/0.0095	0.9694829	0.04477215
0.01/0.009	0.9694733	0.04476961
0.01/0.0085	0.9694664	0.04476595
0.01/0.008	0.9694561	0.04476149
0.01/0.0075	0.9694495	0.04475377
0.01/0.007	0.9694450	0.04476145
0.01/0.0065	0.9694432	0.04473399
0.01/0.006	0.9694447	0.0447228
0.01/0.0055	0.9694491	0.04471146
0.011/0.0065	0.9694553	0.04473687
0.0095/0.0065	0.96944154	0.0447336
0.009/0.0065	0.96944223	0.04473209
0.0085/0.0065	0.96944447	0.04473111

d = 6 & p_2h = 0.71		
p_2hx / p_2hy	RsMSE	CVM_C_NC
0.01/0.01	0.96974415	0.04512671
0.01/0.02	0.96987994	0.04512116
0.01/0.009	0.96972562	0.04512726
0.01/0.008	0.96970893	0.04512486
0.01/0.0075	0.96970306	0.04512345
0.01/0.007	0.96969981	0.04512168
0.01/0.0065	0.96970041	0.04511782
0.01/0.006	0.96970471	0.04511346
0.011/0.007	0.96970845	0.04511265
0.009/0.007	0.96967826	0.04511261
0.008/0.007	0.96965959	0.04510712
0.007/0.007	0.96964287	0.04510056
0.006/0.007	0.96962263	0.04509583
0.005/0.007	0.96959261	0.04508842
0.004/0.007	0.96954609	0.04507371
0.0035/0.007	0.96951627	0.04506352
0.003/0.007	0.96933382	0.04463136
0.0025/0.007	0.96951175	0.04504120
0.002/0.007	0.96955602	0.04507126

***Appendix I**: Results of the optimisation of p_2h_x and p_2h_y by minimising the CVM_C_NC*

The initial values taken from the table in Appendix B are highlighted in blue, while yellow shows the optimum values after minimisation of the RsMSE by modifying the hyperparameters p_2h_x *and* p_2h_y.

d = 5 & p_2h = 0.01		
p_2hx / p_2hy	RsMSE	CVM_C_NC
0.01/1.06	1.01458959	0.03745322
0.01/1	1.013464	0.0376273
0.01/1.1	1.015984	0.03737201
0.01/1.2	1.017373	0.03730212
0.01/1.3	1.015592	0.03733792
0.01/1.4	1.016199	0.03744223
0.015/1.2	1.015001	0.0376052
0.005/1.2	1.0197658	0.03713663
0.001/1.2	1.02665592	0.03684649
0.0005/1.2	1.01500103	0.03636047
0.0001/1.2	1.04417036	0.0362753
0.00005/1.2	1.04287332	0.03635654
0.00001/1.2	1.04212864	0.03641817
0.000005/1.2	1.0426509	0.03641845

d = 6 & p_2h = 0.01		
p_2hx / p_2hy	RsMSE	CVM
0.01/0.01	1.002077	0.03862195
0.01/0.005	1.005693	0.03842162
0.01/0.015	1.004633	0.03848353
0.01/0.001	1.014210	0.03790229
0.01/0.0005	1.010243	0.03793978
0.01/0.0001	1.014210	0.03817364
0.015/0.001	1.013287	0.03792488
0.005/0.001	1.01237558	0.03766417
0.001/0.001	1.03230632	0.0362205
0.0005/0.001	1.02902628	0.03640745
0.0001/0.001	1.02851476	0.03660472

d = 7 & p_2h = 0.01		
p_2hx / p_2hy	RsMSE	CVM
0.01/0.01	0.994880	0.03969673
0.01/0.015	0.994595	0.03950836
0.01/0.005	0.993971	0.0395663
0.01/0.02	0.994827	0.03926257
0.01/0.03	0.997585	0.03935461
0.01/0.05	0.997995	0.03918215
0.01/0.04	0.997748	0.0392403
0.01/0.06	0.99624944	0.0392494
0.01/0.07	0.99685197	0.03937746
0.015/0.05	0.99320916	0.039775
0.005/0.05	1.00427506	0.03862049
0.001/0.05	1.03139704	0.03599899
0.0005/0.05	0.99320916	0.03604268
0.0001/0.05	1.01028588	0.03800668
0.00005/0.05	1.0065414	0.03834638
0.00001/0.05	0.99635132	0.03920846

P_2latx/yd	1							2							3						
	Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h	Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h	Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h
1	0.97824	0.98356	0.04497	0.08452	0.0000011	0.00101	0.10000	0.97499	0.97996	0.04519	0.08493	0.0000011	0.00098	0.10000	0.97347	0.97762	0.04480	0.08409	0.0000011	0.00098	0.40000
2								0.97677	0.98160	0.04578	0.08618	0.0000011	0.00112	0.40000	0.97348	0.97800	0.04534	0.08523	0.0000011	0.00102	0.40000
3															0.97234	0.97648	0.04425	0.08299	0.0000011	0.00099	0.40000
4																					
5																					

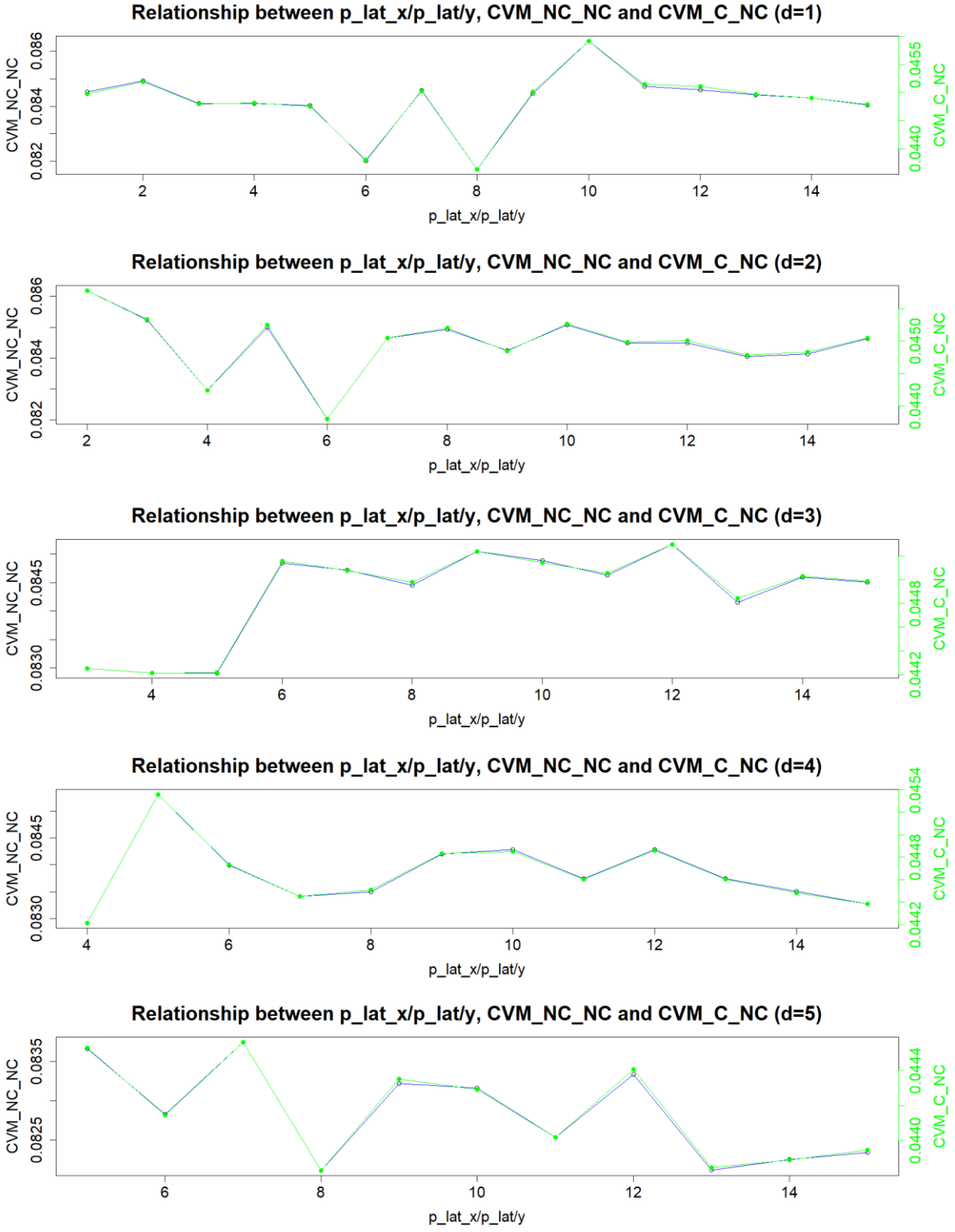
4							5							6						
Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h	Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h	Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h
0.97183	0.98192	0.04481	0.08410	0.0000011	0.00097	0.10000	0.97169	0.97599	0.04475	0.08402	0.0000011	0.00102	0.40000	0.96890	0.97545	0.04380	0.08203	0.0000011	0.00096	0.70000
0.97232	0.97772	0.04424	0.08295	0.0000012	0.00099	0.40000	0.96928	0.97365	0.04525	0.08500	0.0000011	0.00099	0.40000	0.96888	0.97550	0.04380	0.08203	0.0000011	0.00095	0.70000
0.97204	0.97859	0.04421	0.08292	0.0000011	0.00101	0.40000	0.96986	0.97718	0.04422	0.08291	0.0000013	0.00105	0.40000	0.96822	0.97256	0.04516	0.08484	0.0000011	0.00099	0.70000
0.97204	0.97859	0.04421	0.08292	0.0000011	0.00101	0.40000	0.97131	0.98367	0.04536	0.08531	0.0000012	0.00117	0.70000	0.96869	0.97408	0.04473	0.08399	0.0000011	0.00097	0.70000
							0.97178	0.98259	0.04453	0.08367	0.0000011	0.00106	0.70000	0.96858	0.97459	0.04414	0.08283	0.0000011	0.00096	0.70000

7							8							9						
Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h	Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h	Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h
0.97189	0.98154	0.04503	0.08457	0.0000011	0.00097	0.10000	0.97021	0.97590	0.04363	0.08171	0.0000011	0.00096	0.70000	0.97185	0.97746	0.04501	0.08448	0.0000011	0.00101	0.10000
0.96828	0.97430	0.04505	0.08465	0.0000011	0.00102	0.40000	0.96746	0.97358	0.04520	0.08492	0.0000011	0.00099	0.40000	0.96791	0.97309	0.04485	0.08424	0.0000011	0.00100	0.40000
0.96714	0.97258	0.04508	0.08472	0.0000011	0.00103	0.70000	0.96555	0.97174	0.04498	0.08445	0.0000011	0.00095	0.70000	0.96621	0.97078	0.04524	0.08504	0.0000011	0.00101	0.70000
0.96781	0.97397	0.04445	0.08342	0.0000014	0.00103	0.70000	0.96613	0.97286	0.04451	0.08350	0.0000011	0.00094	0.70000	0.96702	0.97210	0.04483	0.08420	0.0000011	0.00101	0.70000
0.96915	0.98387	0.04456	0.08375	0.0000011	0.00104	0.70000	0.96810	0.97561	0.04383	0.08211	0.0000007	0.00094	0.70000	0.96878	0.97387	0.04435	0.08322	0.0000011	0.00100	0.70000

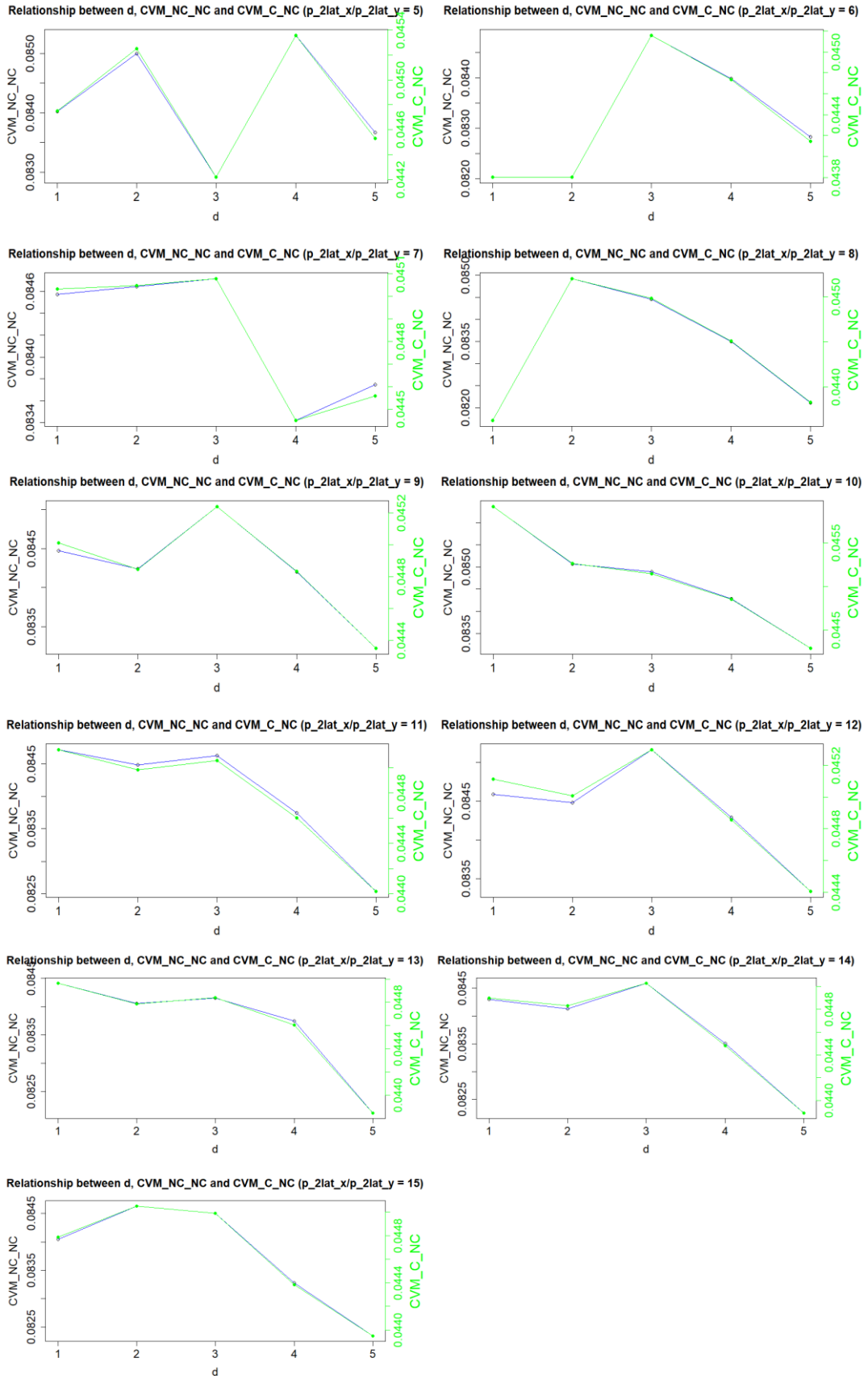
10							11							12						
Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h	Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h	Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h
0.97177	0.97714	0.04592	0.08637	0.0000011	0.00103	0.40000	0.97108	0.97698	0.04515	0.08472	0.0000011	0.00099	0.10000	0.97179	0.97725	0.04511	0.08459	0.0000011	0.00099	0.10000
0.96748	0.97499	0.04526	0.08507	0.0000011	0.00101	0.40000	0.96667	0.97483	0.04499	0.08448	0.0000011	0.00102	0.40000	0.96687	0.97237	0.04501	0.08448	0.0000011	0.00098	0.70000
0.96611	0.97003	0.04515	0.08489	0.0000011	0.00102	0.70000	0.96554	0.96978	0.04506	0.08462	0.0000011	0.00101	0.70000	0.96608	0.96990	0.04530	0.08517	0.0000011	0.00103	0.70000
0.96671	0.96978	0.04485	0.08428	0.0000011	0.00102	0.70000	0.96629	0.97184	0.04460	0.08374	0.0000011	0.00099	0.70000	0.96650	0.96968	0.04486	0.08429	0.0000022	0.00102	0.70000
0.96834	0.97503	0.04429	0.08316	0.0000011	0.00102	0.70000	0.96745	0.97263	0.04402	0.08254	0.0000022	0.00095	0.70000	0.96784	0.97049	0.04441	0.08333	0.0000011	0.00101	0.70000

13							14							15						
Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h	Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h	Obj_fun	RsMSE	CVM_C_NC	CVM_NC_NC	CVM_all	CVM_all2	p_2h
0.97124	0.97581	0.04496	0.08441	0.0000011	0.00097	0.10000	0.97007	0.98535	0.04490	0.08430	0.0000011	0.00102	0.10000	0.97117	0.98021	0.04479	0.08404	0.0000011	0.00093	0.10000
0.96670	0.97248	0.04478	0.08405	0.0000011	0.00099	0.40000	0.96607	0.97208	0.04483	0.08412	0.0000011	0.00099	0.40000	0.96650	0.97491	0.04505	0.08463	0.0000011	0.00099	0.40000
0.96527	0.97008	0.04484	0.08415	0.0000011	0.00096	0.70000	0.96486	0.96902	0.04503	0.08459	0.0000011	0.00100	0.70000	0.96478	0.96938	0.04499	0.08450	0.0000011	0.00101	0.70000
0.96629	0.97184	0.04460	0.08374	0.0000011	0.00099	0.70000	0.96456	0.96907	0.04448	0.08351	0.0000011	0.00095	0.70000	0.96513	0.97012	0.04438	0.08328	0.0000011	0.00097	0.70000
0.96724	0.97211	0.04385	0.08211	0.0000011	0.00095	0.70000	0.96692	0.97013	0.04389	0.08225	0.0000011	0.00093	0.70000	0.96653	0.97241	0.04395	0.08234	0.0000011	0.00091	0.70000

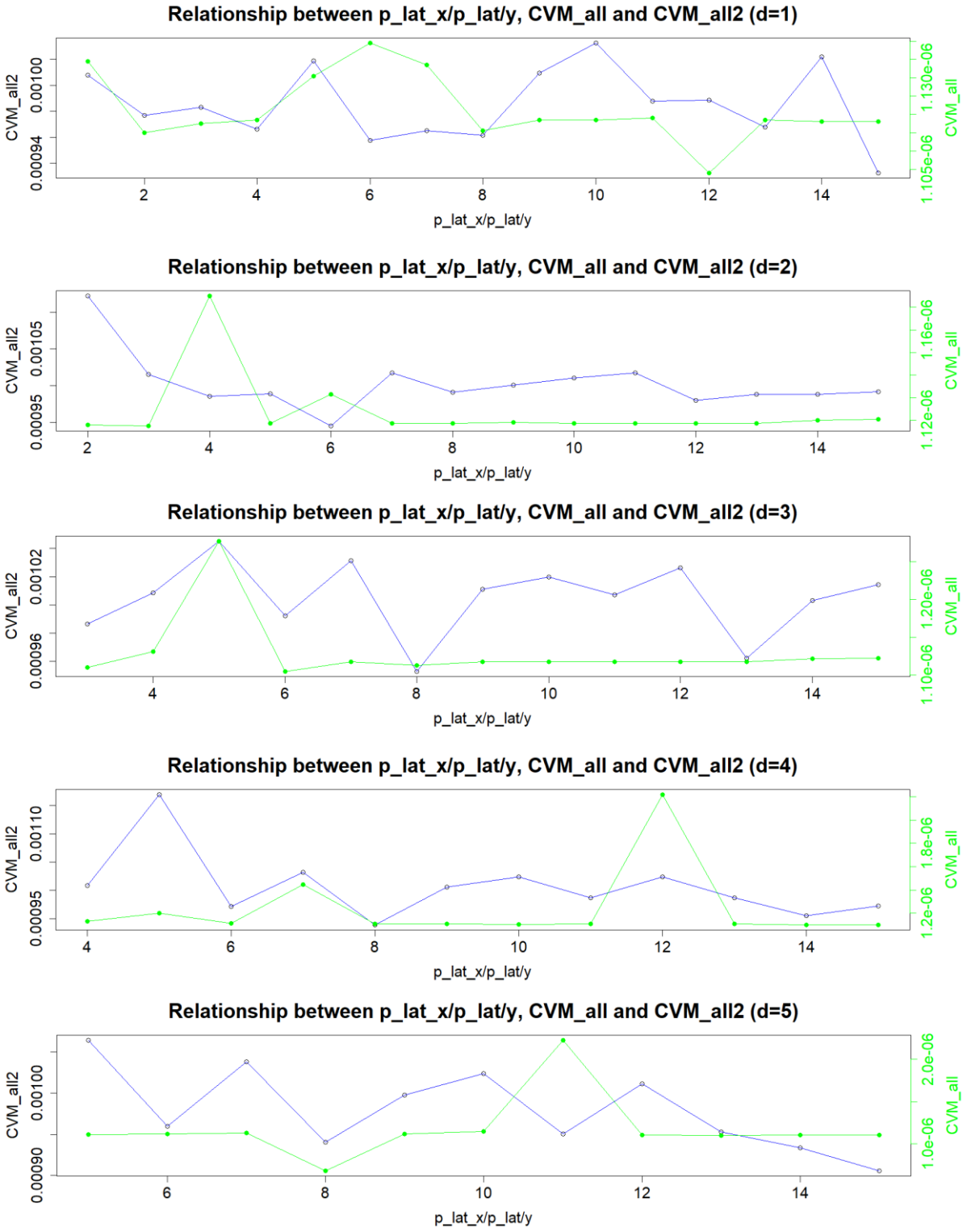
***Appendix K**: Relationship between CVM_C_NC and CVM_NC_NC as a function of the value of p_2h for each value of d for the A-CCA approach*



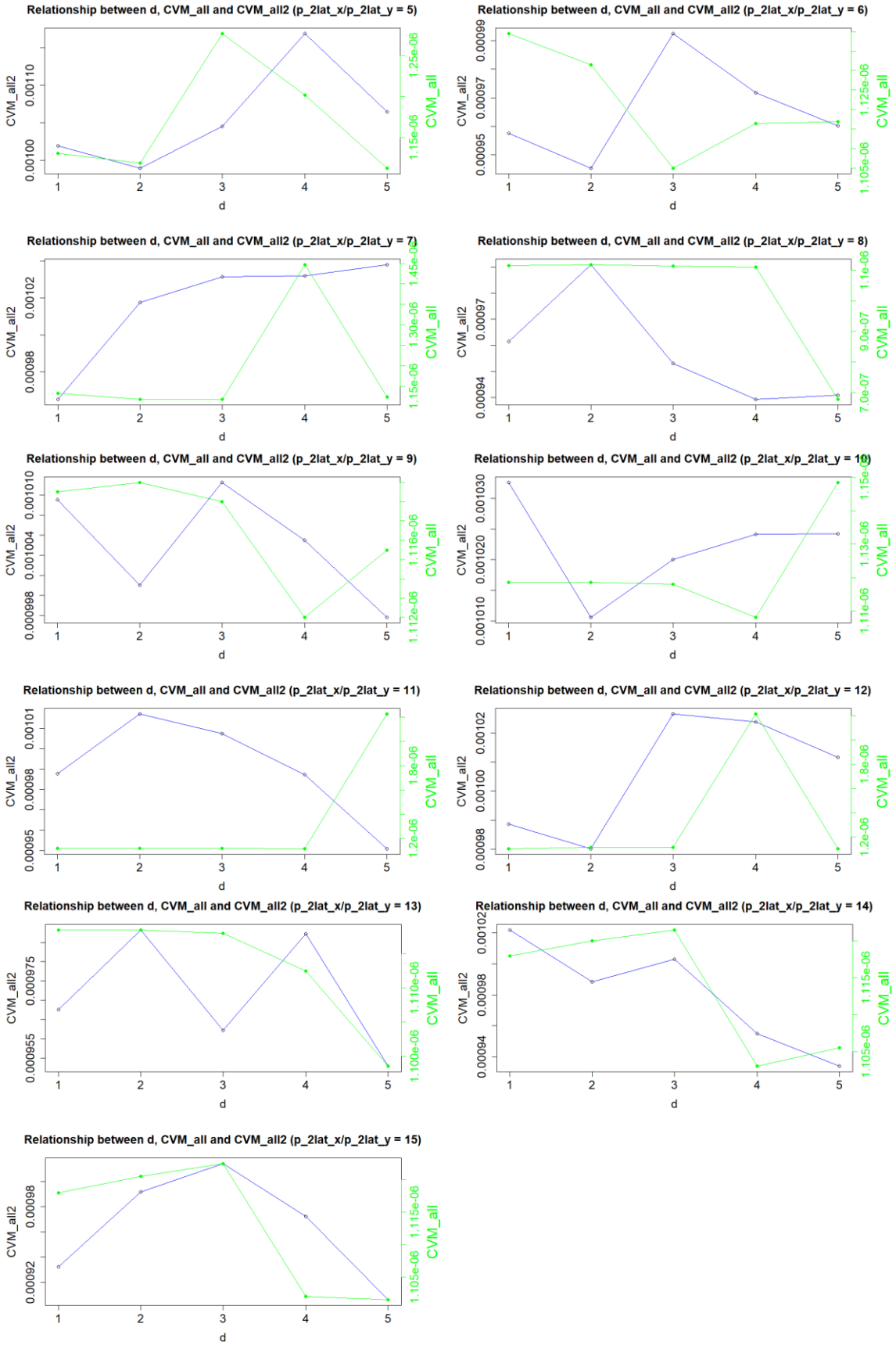
***Appendix L**: Relationship between CVM_C_NC and CVM_NC_NC as a function of the dimension d for each value of p_2h for the A-CCA approach*



***Appendix M**: Relationship between CVM_{all}, CVM_{all2}, p_2h for each dimension d for the KCCA approach*



***Appendix N**: Relationship between CVM_{all}, CVM_{all2}, d for each value of p_2h for the A-CCA approach*



***Appendix O**: Results of the evaluation of the cross-validation technique performance for the A-CCA approach*

The initial values taken from the table in Appendix J are highlighted in blue, while yellow shows the optimum values after minimisation of the RsMSE by modifying the hyperparameters $p_2 h$.

d = 1 & P_2latx/y = 4		
p2_h	RsMSE	CVM_C_NC
0.04	0.98299779	0.04381125
0.06	0.9842191	0.04423100
0.08	0.98194388	0.04459942
0.1	0.98192264	0.04481107
0.15	0.98255344	0.04519911

d = 2 & P_2latx/y = 9		
p2_h	RsMSE	CVM_C_NC
0.45	0.97340719	0.045001
0.4	0.97309284	0.04484595
0.35	0.97214214	0.04502965
0.3	0.97212167	0.04425717
0.25	0.97205778	0.04452327
0.2	0.97239129	0.04406406
0.15	0.97367864	0.04333538
0.1	0.97949489	0.04163615

d = 1 & P_2latx/y = 6		
p2_h	RsMSE	CVM_C_NC
0.55	0.98074157	0.04392865
0.6	0.98006529	0.04435778
0.65	0.97644962	0.04347295
0.7	0.97545452	0.04380102
0.75	0.97473398	0.04410539
0.8	0.97421409	0.04439618
0.85	0.97389000	0.04466684
0.9	0.97374570	0.04491710
0.95	0.97376482	0.04515006
1	0.97392359	0.04537136
1.05	0.97419682	0.04557825

***Appendix P**: Results of the optimisation of p_2h_x and p_2h_y by minimising the RsMSE*

The initial values taken from the table in Appendix J are highlighted in blue, while yellow shows the optimum values after minimisation of the RsMSE by modifying the hyperparameters p_2h.

d = 3 & P_2latx/y = 14		
p2_h	RsMSE	CVM_C_NC
0.4	0.97215184	0.04408377
0.45	0.96936866	0.04414127
0.5	0.96917261	0.04435862
0.55	0.96855524	0.04453943
0.6	0.96904765	0.04470924
0.65	0.97011591	0.04484708
0.7	0.96901617	0.04503044
0.75	0.96901992	0.04518795
0.8	0.96928819	0.04533609
0.85	0.9695609	0.04549422

d = 3 & P_2latx/y = 15		
p2_h	RsMSE	CVM_C_NC
0.45	0.969743458	0.04417913
0.5	0.969068414	0.044349961
0.55	0.968780590	0.044521692
0.6	0.968676083	0.044660197
0.65	0.968967595	0.044772532
0.7	0.969376012	0.044988510
0.75	0.969413757	0.045124365
0.8	0.969679015	0.045296969

d = 4 & P_2latx/y = 14		
p2_h	RsMSE	CVM_C_NC
0.55	0.96898534	0.04389996
0.6	0.96871128	0.04413113
0.65	0.96865352	0.04433593
0.7	0.96906588	0.04448076
0.75	0.96870197	0.04472202
0.8	0.96884357	0.04490496
0.85	0.96906854	0.04509798

***Appendix Q**: Results of the optimisation of p_2h_x and p_2h_y by minimising the CVM_C_NC*

The initial values taken from the table in Appendix J are highlighted in blue, while yellow shows the optimum values after minimisation of the RsMSE by modifying the hyperparameters p_2h.

d = 1 & P_2latx/y = 6		
p2_h	RsMSE	CVM_C_NC
0.55	0.980741566	0.0449286538
0.6	0.980065294	0.0443577785
0.65	0.976449616	0.0434729501
0.7	0.975454521	0.0438010183
0.75	0.974733981	0.0441053918
0.8	0.974214091	0.0443961813

d = 2 & P_2latx/y = 6		
p2_h	RsMSE	CVM_C_NC
0.55	0.975837924	0.0453630772
0.6	0.980065294	0.0443577785
0.65	0.976096518	0.0455310122
0.7	0.97550459	0.0438025079
0.75	0.974722033	0.0444104759
0.85	0.973986074	0.0446720796

d = 1 & P_2latx/y = 8		
p2_h	RsMSE	CVM_C_NC
0.5	0.976991742	0.0461439251
0.55	0.977658382	0.0439437598
0.6	0.97550459	0.0438025079
0.65	0.978265787	0.0445739955
0.7	0.975895028	0.0436290944
0.75	0.979099901	0.0450570292
0.8	0.978429068	0.0447684347
0.85	0.976754376	0.0439322150
0.9	0.976029069	0.0442761608
0.95	0.975536366	0.0445815392
1	0.974989049	0.0449124789
1.05	0.974993698	0.0450950166

8 Bibliography

Akaho, S. (2006). *A kernel method for canonical correlation analysis*. https://doi.org/10.48550/ARXIV.CS/0609071

Aluja-Banet, T., Daunis-i-Estadella, J., & Pellicer, D. (2007). GRAFT, a complete system for data fusion. *Computational Statistics & Data Analysis*, *52*(2), 635-649. https://doi.org/10.1016/j.csda.2006.11.029

Anderson, T. W. (1957). Maximum likelihood estimates for a multivariate normal distribution when some observations are missing. *Journal of the American Statistical Association*, *52*(278), 200-203. https://doi.org/10.1080/01621459.1957.10501379

Annoye, H., Beretta, A., & Heuchenne, C. (2024). Statistical matching using kernel canonical correlation analysis and super-organizing map. *Expert Systems with Applications*, *246*, 123134. https://doi.org/10.1016/j.eswa.2023.123134

Asendorf, N. A. (2015). *Informative data fusion : Beyond canonical correlation analysis* [Thesis]. http://deepblue.lib.umich.edu/handle/2027.42/113419

Bajaj, R. H., & Ramteke, P. L. (2014). Big data—The new era of data. *International Journal of Computer Science and Information Technologies*, *5*(2), 1875-1885.

Bhande, A. (2018, 11 June). What is underfitting and overfitting in machine learning and how to deal with it. *Medium*. https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76

Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). What is machine learning? A primer for the epidemiologist. *American Journal of Epidemiology*, *188*(12), 2222-2239. https://doi.org/10.1093/aje/kwz189

Chen, M., Mao, S., & Liu, Y. (2014). Big data : A survey. *Mobile Networks and Applications*, *19*(2), 171-209. https://doi.org/10.1007/s11036-013-0489-0

Data fusion. (2023, octobre 17). *DevX*. https://www.devx.com/terms/data-fusion/

D'Orazio, M., Di Zio, M., & Scanu, M. (2006). *Statistical matching : Theory and practice* (1re éd.). Wiley. https://doi.org/10.1002/0470023554

Eurostat (European Commission), Leulescu, A., & Agafiţei, M. (2013). *Statistical matching : A model based approach for data integration : 2013 edition*. Publications Office of the European Union. https://data.europa.eu/doi/10.2785/44822

Fan, J., Han, F., & Liu, H. (2014). Challenges of big data analysis. *National Science Review*, *1*(2), 293-314. https://doi.org/10.1093/nsr/nwt032

Gavin, N. I. (1985). An application of statistical matching with the survey of income and education and the 1976 Health Interview Survey. *Health Services Research*, *20*(2), 183-198.

Gessendorfer, J., Beste, J., Drechsler, J., & Sakshaug, J. W. (2018). Statistical matching as a supplement to record linkage : A valuable method to tackle nonconsent bias? *Journal of Official Statistics*, *34*(4), 909-933. https://doi.org/10.2478/jos-2018-0045

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, *28*(3/4), 321. https://doi.org/10.2307/2333955

Jabbar, H. K., & Khan, R. Z. (2014). Methods to avoid over-fitting and under-fitting in supervised machine learning (Comparative study). *Computer Science, Communication, and Instrumentation Devices*, 163-172. https://doi.org/10.3850/978-981-09-5247-1_017

Kim, J. K., & Shao, J. (2013). *Statistical methods for handling incomplete data* (0 éd.). Chapman and Hall/CRC. https://doi.org/10.1201/b13981

Lai, P. L., & Fyfe, C. (2000). Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, *10*(05), 365-377. https://doi.org/10.1142/S012906570000034X

Luo, L., Xu, J., Lin, J., Zeng, Q., & Sun, X. (2018). An auto-encoder matching model for learning utterance-level semantic dependency in dialogue generation. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 702-707. https://doi.org/10.18653/v1/D18-1075

Michelucci, U. (2022). *An introduction to autoencoders*. https://doi.org/10.48550/ARXIV.2201.03898

Mitsuhiro, M., & Hoshino, T. (2020). Kernel canonical correlation analysis for data combination of multiple-source datasets. *Japanese Journal of Statistics and Data Science*, *3*(2), 651-668. https://doi.org/10.1007/s42081-020-00074-z

O'Brien, S. (1991). The Role of Data Fusion in Actionable Media Marketing in the 1990's. *Marketing and Research Today*, *19*, 15-22.

Radner, D., Jabine, T., & Allen, R. (1980). *Report on Exact and Statistical Matching Techniques* (Statistical Policy Working Papers 5). Office of Federal Statistical Policy and Standards US DoC. https://www.fcsm.gov/assets/files/docs/spwp5.pdf

Rässler, S. (2002). *Statistical matching : A Frequentist Theory, Practical Applications, and Alternative Bayesian Approaches* (Vol. 168). Springer New York. https://doi.org/10.1007/978-1-4613-0053-3

Rässler, S. (2004). Data fusion : Identification problems, validity, and multiple imputation. *Austrian Journal of Statistics*, *33*(1 & 2), 153-171. https://doi.org/https://doi.org/10.17713/ajs.v33i1&2.436

Rebala, G., Ravi, A., & Churiwala, S. (2019). *An introduction to machine learning*. Springer International Publishing. https://doi.org/10.1007/978-3-030-15729-6

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys* (1re éd.). Wiley. https://doi.org/10.1002/9780470316696

Ruggles, N., & Ruggles, R. (1974). *A strategy for merging and matching microdata sets* (p. 353-371) [NBER Chapters]. National Bureau of Economic Research, Inc. https://econpapers.repec.org/bookchap/nbrnberch/10115.htm

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533-536. https://doi.org/10.1038/323533a0

Saxena, S. (2023, 23 November). *Underfitting and Overfitting in Machine Learning*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/02/underfitting-overfitting-best-fitting-machine-learning/

Shimodaira, H. (2014). *A simple coding for cross-domain matching with dimension reduction via spectral graph embedding*. https://doi.org/10.48550/ARXIV.1412.8380

Spaziani, M., Frattarola, D., & D'Orazio, M. (2019). Integration of survey data in r based on machine learning. *Romanian Statistical Review*, *3*. https://doi.org/10.13140/RG.2.2.14022.93762

Valliant, R., & Dever, J. A. (2017). Overview of weighting. In *Survey Weights: A Step-by-Step Guide to Calculation* (1re éd., p. 1-6). Stata Press.

Van Buuren, S. (2012). *Flexible imputation of missing data*. Chapman and Hall/CRC. https://doi.org/10.1201/b11826

Van Der Putten, P., Kok, J. N., & Gupta, A. (2002). Data fusion through statistical matching. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.297501

Executive summary

In the contemporary era, characterised by the proliferation of digital data, a substantial quantity of information is accumulated from a plethora of sources and activities. Nevertheless, the effective utilisation of this data for commercial, scientific, or other purposes remains a challenging and constrained endeavour, due to the fact that the information required is seldom derived from a single source, but rather from a multitude of disparate sources. To address this challenge, two statistical matching techniques, namely kernel canonical correlation analysis (KCCA) and auto-encoder canonical correlation analysis (A-CCA), have been developed. These techniques employ machine learning algorithms with the objective of merging the necessary information from multiple sources into a single database.

In this context, the objective of this thesis is to optimise the dimensions of the various latent spaces and the bandwidths of the KCCA and A-CCA algorithms. To this end, a database was constructed, and the Grid Search strategy was employed to ascertain the optimal values for the aforementioned hyperparameters. The Root Standardised Mean Squared Error (RsMSE) metric and the Cramer-Von Mises statistic were employed for the evaluation of the performance of the various models.

The results demonstrate the influence of hyperparameter values on model performance. Moreover, the optimal values of these hyperparameters differ when minimising the RsMSE or the Cramer-Von Mises statistic. The discussion section elucidates the relationship between the values of the hyperparameters and their impact on the performance of the algorithms for each of them.

The findings indicate that a compromise between the preservation of the dependencies between variables and the structural characteristics of the data (minimising the Cramer-Von Mises statistic) and the prediction accuracy (minimising the RsMSE) may be necessary, or alternatively, that one of these performance metrics may be favoured.

This thesis makes a contribution to a deeper comprehension of the influence exerted by the values of the various hyperparameters on the performance of the algorithms. The findings of this research will assist the project team in evaluating the quality and consistency of their work, and in implementing any necessary modifications.

Word count: 22,959