8-2023

An Adaptive Deep Learning for Causal Inference Based on Support Points With High-Dimensional Data

Lynda Aouar

UNIVERSITY OF NORTHERN COLORADO

Greeley, Colorado

The Graduate School

AN ADAPTIVE DEEP LEARNING FOR CAUSAL INFERENCE BASED ON SUPPORT POINTS WITH HIGH-DIMENSIONAL DATA

A Dissertation Submitted in Partial Fulfillment
of the Requirements of the Degree of
Doctor of Philosophy

Lynda Aouar

College of Education and Behavioral Science
Department of Applied Statistics and Research Methods

August 2023

ABSTRACT

Aouar, Lynda. *An adaptive deep learning for causal inference based on support points with high-dimensional data*. Published Doctor of Philosophy dissertation, University of Northern Colorado, 2023.

The Sample splitting method in semiparametric statistics could introduce inconsistency in inference and estimation. Thus, to make adaptive learning based on observational data and establish valid learning that helps in the estimation and inference of the parameters and hyperparameters using double machine learning, this study introduces an efficient sample splitting technique for causal inference in the semiparametric framework, in other words, the support points sample splitting(SPSS), a subsampling method based on the energy distance concept is employed for causal inference under double machine learning paradigm.

This work is based on the idea that the support points sample splitting (SPSS) is an optimal representative point of the data in a random sample versus the counterpart of random splitting, which implies that the support points sample splitting is an optimal sub-representation of the underlying data generating distribution. To my best knowledge, the conceptual foundation of the support points-based sample splitting is a cutting-edge method of subsampling and the best representation of a full big data set in the sense that the unit structural information of the underlying distribution via the traditional random data splitting is most likely not preserved. Three estimators were applied for double/debiased machine learning causal inference a paradigm that estimates the causal treatment effect from observational data based on machine learning algorithms with the support points sample splitting (SPSS). This study is

considering Support Vector Machine (SVM) and Deep Learning (DL) as the predictive estimators. A comparative study is conducted between the SVM and DL with the support points technique to the benchmark results of Chernozhukov et al. (2018) that used instead, the random forest, the neural network, and the regression trees with random k-fold cross-fitting technique.

An ensemble machine learning algorithm is proposed that is a hybrid of the super learner and the deep learning with the support points splitting to compare it to the results of Chernozhukov et al. (2018). Finally, a socio-economic real-world dataset, for the 401(k)-pension plan, is used to investigate and evaluate the proposed methods to those in Chernozhukov et al. (2018).

The result of this study was under 162 simulations, shows that the three proposed models converge, support vector machine (SVM) with support points sample splitting (SPSS) under double machine learning (DML), the deep learning (DL) with support points sample splitting under double machine learning (DML), and the hybrid of super learning (SL) and deep learning with support points sample splitting under double machine learning. However, the performance of the three models differs.

The first model, support vector machine (SVM) with support points sample splitting (SPSS) under double machine learning (DML) has the lowest performance compared to the other two models. In terms of the quality of the causal estimators, it has a higher *MSE* and inconsistency of the simulation results on all three data dimension levels, low-high-dimensional (p = 20,50,80), moderate-high-dimensional (p = 100, 200, 500), and big-high-dimensional p = (1000, 2000, 5000). The two other models, deep learning (DL) with support points sample splitting under double machine learning (DML), and the hybrid of super learning (SL) and deep learning with support points sample splitting under double machine learning have produced a

competing performance and results in terms of the best estimation compared to the two other methods. The first model was time efficient to estimate the causal inference compared to the third one. But the third model was better performing in terms of the estimation quality by producing the lowest *MSE* compared to the other two models.

The results of this research are consistent with the recent development of machine learning. The support vector machine learning has been introduced in the previous century, and it looks like it is no longer showing efficiency and quality estimation with the recent emerging double machine learning. However, cutting-edge methods such as deep learning and super learner have shown superior performance in the estimation of the causal double machine learning target estimator, and efficiency in the time of computation.

DEDICATION

To my son, my heart, who I have not seen for more than three years, and who always tells me: “I am proud of you, mama.” To my parents, and my late grandfather, who was a dearest friend. He exemplified wisdom. Once he said to me, Lynda, you will go far, very far.

Here I am, ten thousand miles away from home, and far into my career path and far from my history. Thank you, grandfather, for believing in me.

ACKNOWLEDGEMENTS

To my advisor, Prof. Han Yu, his patience, guidance, and encouragement towards submitting this work were marked by his gracious time and knowledge, even on the weekends, when needed.

I wish to recognize my committee members for their valued time and support, Prof. Daniel Mundfrom, Prof. Chia-Lin Tsai, Prof. Jodie Rommel, and the Department Chair, Prof. Randy Larkins.

Dr. V. Roshan Joseph, the author of the Support Point paper (Joseph & Vakayil, 2021; Mak & Joseph, 2018) on which my dissertation is built, generously replied to communications, and answered my questions on the various clarifications needed and shared the codes I asked for.

To Dr. Christian Hansen for his generosity to answer my questions and for sharing the codes of the paper he was an author in Chernozhukov et al. (2018).

To Colorado University in Boulder for their unlimited help with the high-performance computing

Associate Dean of the Graduate School, Dr. Cindy Wesley, thank you for your kindness, support, and your confidence in me. Also, gratitude is extended to Keighley Gurney in the Office of the Dean.

Noted kindness is appreciated by Kara LaSota, Ms. Jane Borisova, and Ms. Olga Baron in the International Students Office.

To Ms. Carol Steward in the Graduate School office for her kindness and patience in providing all the information needed and answering questions generously.

To my colleagues, David Agboola, and Sami Saad Alanazi for sharing their dissertation experience, their codes, and the time to discuss my developed codes.

Prof. Aminu Mamman, Manchester University Business School, UK, who was my teacher in the MBA program in the United Arab Emirates. His diligent support and encouragement since my graduation in 2010, and still today assisted in higher potential.

Psychologist Dr. Raymond. H. Hamden, known to me for more than 12 years while residing in the United Arab Emirates, opened my eyes to the opportunity of earning a Ph.D. in Statistics. I remember him saying, "Lynda, you have the potential to achieve a doctoral-level education in Statistics or Predictive Analysis or both. I recommend that you continue such academic pursuits in the United States. The world needs people like you who can reach their highest potential."

To my colleague, Prof. Makhtar Sarr, who taught statistics at Penn State University and North Carolina State University, thank you for your unwavering encouragement.

To my family, specifically my youngest sister, Kenza, I owe her a model of inspiring diligence, dedicated work ethic, and ambition as a physician. As she prepares for the USA examination for Medical School Residency, our reunion is my vision.

Special thanks to my father, an educator, who taught me to love education, and modeled the spirit of teaching and truly caring about his students. I do remember, even after more than 30 years of a career in education, he would be around his office desk fully dedicated to preparing for the next day of classes. He gave his students the best effort each day without tiring. The image of

him fully engaged in the preparation task at his office desk will always live in my mind and inspires me in the educational realm.

To my late friend and colleague Becky DeOliveira,

My classmates at the University of Northern Colorado: Hanadi Alomari, Michael Safo, Aziz AlQahtani, Kofi Wagya, thank you for being such great friends and colleagues. It is so fortunate to have the privilege of our UNC journey together. You made life easier and tolerable during times of frustration, as well as, supportive and jovial during the many Celebrations of Success.

You are all, everyone, my family home away from home … Thank God for You.

TABLE OF CONTENTS

CHAPTER

II. continued

CHAPTER

III. continued

LIST OF TABLES

Table

LIST OF FIGURES

Figure

LIST OF ACRONYMS

AL	Adaptive Learning
ANN	Artificial Neural Networks
ATA	Average Treatment Effect
BHD	Big-High-Dimensional Data
CIA	Conditional Independence Assumption
CML	Causal Machine Learning
CML	Causal Machine Learning
CV	Cross Validation
DL	Deep Learning
DML	Double Machine Learning
HD	High-Dimensional Data
LHD	Low-High-Dimensional Data
MHD	Moderate-High-Dimensional Data
ML	Machine Learning
RKHS	Reproducible Kernels Hilbert Space
SDL	Super Deep Learning
SL	Super Learner
SPSS	Support Points Sample Splitting
SVM	Support Vector Machine

CHAPTER I

INTRODUCTION

In the context of estimating the treatment effect from observational data, economists, statisticians, and social scientists have been developing models to estimate the effect of the target policy parameter. Firpo (2007) has introduced a doubly staged method to estimate the quantile treatment effect, which is based on estimating initially the nuisance parameter and then the estimation of the quantile variable of interest. However, this method has shown a limitation in the cases where there is a high dimensional confounder, and in the case in which the sample size is much smaller than the number of the nuisance variables ($p >> N$).

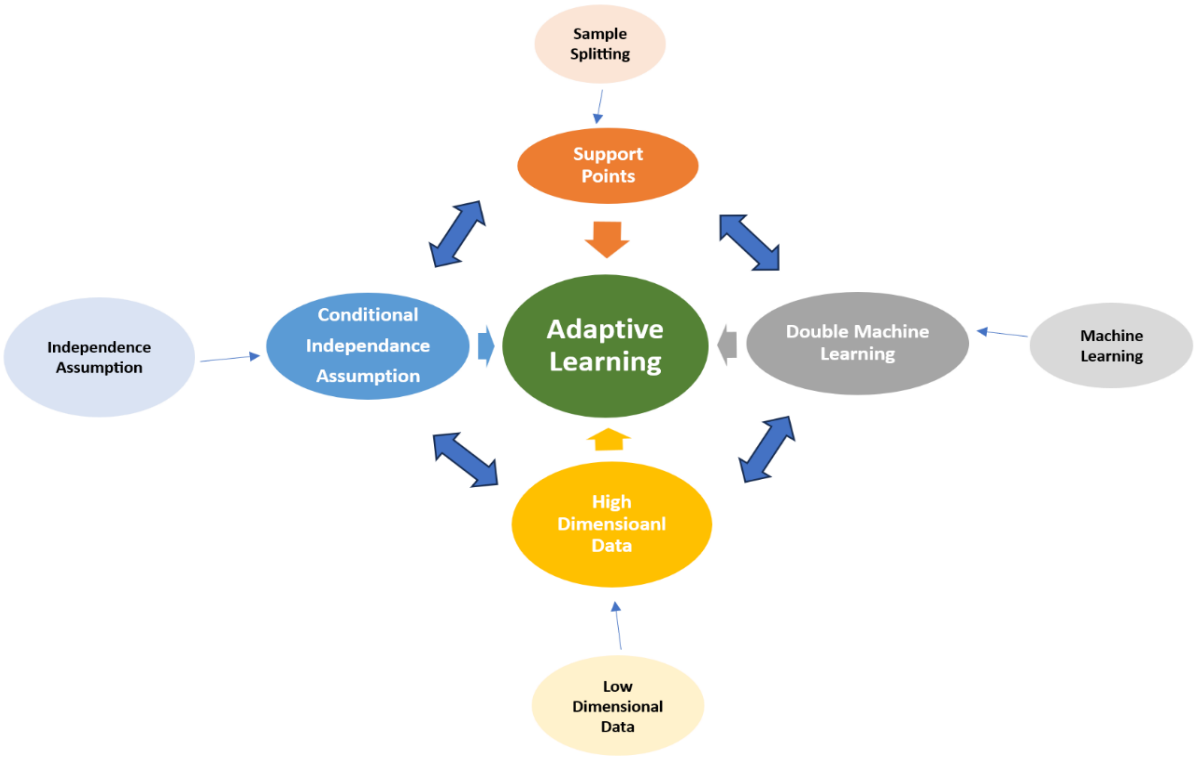
Chernozhukov et al. (2018) have been developing a Double/Debiased Machine Learning method as an extension to Firpo's (2007) work and built upon the work of Belloni et al. (2012), Belloni et al. (2014), Chernozhukov et al. (2015), and Belloni et al. (2017). Double/debiased machine learning is a two-step causal inference with observational data for estimating the average treatment effect. A two-staged bias correction is adopted in this method (Chernozhukov et al., 2022) by using the Neyman orthogonalization and moment score function to undertake the regularization bias of the target estimator (Klosin, 2021). Sample splitting as a cross-fitting technique overcomes the bias introduced by the model overfitting dilemma (Bach et al., 2022). The Neyman orthogonality delivers an estimation of $\sqrt{}$ n-rates to the target parameter and allows an asymptotic normal distribution convergence (Lewis & Syrgkanis, 2021).

This study is focused on statistical adaptive learning (AL), a concept used to adapt the statistical model to the data distribution in semiparametric framework (Bickel et al., 2000;

Chambaz et al., 2016, Van der Laan et al., 2004). In high dimensional settings, we need the adaptive learning tools to help for data reduction by retaining the original data features (Vakayil & Joseph, 2022). The support point sample splitting (SPSS) method is implemented as it is an optimal adaptation to the data distribution versus the random splitting (Mak & Joseph, 2018). This research takes into consideration the conditional independence (CIA) instead of the independence assumption (Knaus, 2021). The learning approach considered is the double machine learning (DML) for causal inference (Chernozhukov et al., 2018). Figure 1 summarizes the work map of this study as follows,

Figure 1

Study Work Map



Identification of Double Machine Learning for Semiparametric Causal Inference

This study specializes in the double machine learning model, the partial linear regression, and considers it as the identified model for the causal effect of the treatment variable T. The model could be described as follows, (Chernozhukov et al., 2018),

$$Y = T\beta_0 + g_0(\mathbf{X}) + U, \mathbb{E}[\, U \mid \mathbf{X}, T\,] = 0,$$

$$T = m_0(\mathbf{X}) + V, \mathbb{E}[\, V \mid \mathbf{X}\,] = 0,$$

where Y is the response variable, $\boldsymbol{X}$ = (X_1,..., X_p) is the covariate vector, and T is the target treatment variable. The confounder $\boldsymbol{X}$ affects the outcome Y as well as the treatment effect variable T through the functional parameters $g_0(.)$ and $m_0(.)$respectively. β_0 is the parameter about the causal effect of *T*. U and V are the disturbances.

Consider $\eta_0 = (g_0(.), m_0(.))$ the nuisance parameter, whose dimension is higher than the sample size, N. Under these conditions the traditional assumption that the sample size, N, is larger than the P fails to be met.

Double Machine Learning Estimator

Definition and Construction

Chernozhukov et al. (2018) have defined two methods to construct the estimators of double machine learning. The following are assumptions for the model construction,

1. η_0 is the true value of the nuisance parameter $\eta \in \mathcal{N}$.
2. The true causal parameter β_0 of the target parameter $\beta \in \Theta \subseteq \mathbb{R}^{\mathrm{d}_\beta}$ satisfies the moment criteria:

$$E_P[\psi(Z; \beta_0, \eta_0)] = 0.$$

3. $(Z_i)_{i=1}^{N}$ is the iid random sample from the distribution of Z.

4. Z is a random element in the measurable space $(\mathcal{Z}, \mathcal{A}_{\mathcal{Z}})$ that has probability measure $P \in \mathcal{P}_N$.

5. The vector of the known Neyman orthogonal score functions $\psi = \left(\psi_1, \ldots, \psi_{\mathrm{d}_\beta}\right)'$, such that ψ_j , j={1,…, d_β} is a function defined on $\mathcal{Z} \times \Theta \times \mathcal{N}$ and map on $\mathbb{R}$, and are measurable if assigning Θ and $\mathcal{N}$ with their Borel σ-fields.

6. Θ is a non-empty subset from $\mathbb{R}^{\mathrm{d}_\beta}$.

Definition 1

Consider $(I_k)_{k=1}^K$, a K-fold random partition of a sample of N cases. Let the complement set I_k^c for each I_k(with a size n= N/K) where k∈{1,…, K} be: $I_k^c = \{1, \ldots, N\} \setminus I_k$. Consider the fold I_k has a size n = N/K. For each k∈{1, …, K},

1. Construct the machine learning estimator $\grave{\eta}_{0,k}$,

$$\grave{\eta}_{0,k} = \grave{\eta}_0\left((Z_i)_{i \in I_k^c}\right).$$

2. Taking $E_{n,k}[\psi(W)] = n^{-1}\sum_{i \in I_k}\psi(W_i)$ as the expectation of the k^{th} fold, calculate the k^{th} target parameter estimator $\grave{\beta}_{0,k}$ that satisfies,

$$\mathbb{E}_{n,k}\left[\psi\left(Z; \grave{\beta}_{0,k}, \grave{\eta}_{0,k}\right] = 0.\right.$$

3. Construct the final target estimator that is a combination of the k estimators, called the DML1 estimator as follows,

$$\grave{\beta}_0 = \frac{1}{K}\sum_{k=1}^K \grave{\beta}_{0,k}.$$

4. Alternatively, in the second step, directly calculate the target estimator without any further steps. In this case, it is the DML2 estimator, which is defined as follows:

$$\frac{1}{K}\sum_{k=1}^K E_{n,k}\left[\psi\left(Z; \grave{\beta}_0, \hat{\eta}_{0,k}\right)\right] = 0.$$

Motivation

Many works after the established work of Chernozhukov et al. (2018) have emerged. Such as in Guo et al. (2022) have developed what is called doubly debiased Lasso, a causal estimation model from observational data considering the existence of high dimensional unobserved covariates. They have handled the correction of the bias arising from both the high dimensionality and hidden nuisance variables.

To my best knowledge, there is no study from the literature that has employed SVM, DL, and Super Learner for debiased machine learning with support points sample splitting. SVM and DL have been chosen in this study because they are known for their effectiveness in tuning the hyperparameter. The hybrid methods of double machine learning and causal inference are nowadays a cutting-edge area in the practice and methodological studies (Knaus, 2021)

This research aims to develop a support points-based DML, that makes an intelligent learning of observational data. The second purpose of this study is to compare the proposed frameworks of the estimation of the average treatment effect (ATE) in structural causal models using the DML paradigm to the original work of DML introduced in the literature (Chernozhukov et al., 2018). An investigation of the performance of the three methods, support vector machine (SVM), deep learning (DL), and super learner (SL) with the support points sample splitting (SPSS) method compared to the k-fold sample splitting utilized in Chernozhukov et al. (2018). To my best knowledge, this method suggests models that consist of hybrid methods that are different from the literature and from what has been studied in the past.

Ju et al. (2018) have applied various kinds of deep neural networks (DNN) assigned with different layers of depth for each learner along with the super learner (SL) method. A research paper has created a combined algorithm developed from deep neural networks and super learner

methods (Young et al., 2018). J. Yang et al. (2020) have studied double machine learning with support vector machine (SVM) and k-fold cross-validation. A research study has been conducted by Kebonye (2021) using a combination of the two methods, support points sample splitting (SPSS) and support vector machine (SVM). Varaku (2021) has applied the mixture of double machine learning (DML) with deep neural networks (DL). A causal effect framework (Heiler & Knaus, 2022) has applied double machine learning, deep learning, and k-fold cross-validation. A research paper has combined double machine learning (DML) with support points sample splitting (Agboola & Yu, 2023). A dissertation study that has used double machine learning (DML) joined with the super learner (Alanazi, 2022).

None of those works has handled the double machine learning (DML) framework using support vector machine (SVM), deep learning (DL), and super learner (SL) with the support points sample splitting (SPSS) all together in one study. Double machine learning (DML) with support points sample splitting.

Research Questions

The following are the research questions and sub-questions:

Q1 How does double machine learning (DML) using support vector machine (SVM) and support points sample splitting (SPSS) perform compared to the DML used in the work of Chernozhukov, et al. (2018)?

- Q1a How does DML using support vector machine (SVM) and support points splitting (SPSS) perform in the simulation?
- Q1b How does DML using support vector machine (SVM) and support points sample splitting (SPSS) perform compared to the DML used in the work of Chernozhukov, et al. (2018) in the real-world data?

Q2 How does DML using deep learning (DL) and support points splitting sample (SPSS) perform compared to the DML used with the work of Chernozhukov, et al. (2018)?

Q2a How does DML using deep learning (DL) and support points sample splitting (SPSS) perform with simulated data?

Q2b How does DML using deep learning (DL) and support points sample splitting (SPSS) perform compared to the DML used in the work of Chernozhukov, et al. (2018) with the real-world data?

Q3 How does DML using Super Learner (SL) based on deep learning (DL) and support points sample splitting (SPSS) perform compared to the ensemble used in the work of Chernozhukov, et al. (2018)?

Q3a How does DML using Super Learner (SL) based on the deep learning (DL) and support points sample splitting (SPSS) perform with simulated data?

Q3b How does DML using Super Learner (SL) based on deep learning (DL) and support points sample splitting (SPSS) perform compared to the ensemble used in the work of Chernozhukov, et al. (2018) with real-world data?

Organization of the Dissertation

The study is divided into five chapters. In Chapter 1, a general introduction of the support point splitting idea applied for the semiparametric causal inference with observational data is presented. The motivation behind the study and the research questions are stated. Chapter II introduces the semiparametric models, the sample splitting for cross-validation, and the causal inference for observational data. In Chapter III, the methodology of this dissertation is described, which consists of the heuristic theoretical framework of the support points splitting. An introduction of the dissertation models, which is a hybrid of the support points splitting, and the following methods: the support vector machine (SVM), deep learning (DL), and super learner (SL). A detailed simulation plan is presented along with the intended real-world data. Chapter IV presents the results of the simulation and investigation of real-world data 401(k) plans and pension accounts. Chapter V contains the conclusion of the dissertation, the research limitations, and suggested future work as an extension to this research.

Concepts Definitions

A hyperparameter is a parameter that could be learned from data or set beforehand. The machine learning process tunes the hyperparameter by searching for the optimal value that could deliver the best performance of the model.

Adaptive Learning (AL) is a statistical technique that can adapt to the data distribution by adjusting the parameters of the algorithm based on the data information and data patterns. Specifically, this is a very useful technique in high-dimensional data reduction.

Cross Validation (CV) is a resampling technique that uses various parts of the sample to train a model and test its performance.

Double/debiased machine learning (DML) is a two-step causal inference with observational data for estimating the average treatment effect. A two-staged bias correction by using the Neyman orthogonalization to undertake the regularization bias of the target estimator. Sample splitting as a cross-fitting technique overcomes the bias introduced by the model overfitting.

High-Dimensional Data (HD) is data that could have the number of variables (p) much higher than the sample size, N, that is $p >> N$.

Nonparametric Model is a statistical model that has infinite-dimensional parameters, or a distribution free model.

Parametric Models is a statistical model that has finite-dimensional parameters.

Sample splitting is a method used to partition the original data into different portions to use them for cross-validation.

Semiparametric model is a model that is a combination of two components, parametric and nonparametric models. They could tolerate less rigorous assumptions on the nuisance parameters and keep the parametric rigorous assumptions of the target parameter.

Support Points Sample Splitting (SPSS) is a sample splitting for cross-validation method that delivers the best representation of the original distribution.

CHAPTER II

LITERATURE REVIEW

This section introduces the concept of sample splitting and cross-validation, the causal inference for observational data, double/debiased machine learning (DML), the semiparametric models, and the Reproducing Kernels Hilbert Space (RKHS).

Sample Splitting and Cross-Validation

Sample splitting is a technique that is used to divide the sample data into three subgroups: the training, the testing, and the validation subsamples (Picard & Berk, 1990; Snee, 1977) to build the model and assess its prediction accuracy.

Data splitting for cross-validation is a very crucial stage in the machine learning paradigm as it facilitates parameter estimation, model building, model performance evaluation, and the hyperparameters tuning for model selection. It overcomes the underfitting and overfitting challenges, the issues that the machine learning techniques could suffer from if the researcher does not take into consideration the sample splitting. For instance, in double machine learning, if the estimation of the target causal parameter and the nuisance parameter is run without considering the sample splitting, it could most likely introduce a bias in the estimators induced by either underfitting or overfitting (Chernozhukov et al., 2018).

A further sample splitting use is for solving the difficulties of the significance tests. The data have been subsampled into two subgroups, one applied for the hypothesis test and the other for the significance assessment (Cox, 1975). However, the target of this study is sample splitting for cross-validation. However, cross-validation operates a comparison between suggested models

to decide the optimal one that will be adapted to the data to predict the performance of this chosen model for future data (Yadav & Shukla, 2016). An important component that the researcher should take into consideration is that the quality of the estimation and the inference results will be impacted by the choice of sample-splitting techniques (Meng et al., 2020).

Another challenge that could arise from the practice of sample splitting is holding small data for developing the model, which leads to a poor prediction for future observations. Or, in contrast, reserving small data for the validation stage could deteriorate the model evaluation performance. So, a trade-off between the training set size and the validation set size is crucial to ensure an effective subsampling framework (Picard & Berk, 1990). In most cases, the subsample splitting methods are applied under the randomization concept, which will suffer from the consistency of the data analysis results, and different findings from the same model are caused by the randomization of the sample splitting (Cox, 1975; Shao, 1997; Y, Yang, 2007).

Other alternatives could be used for sample splitting such as PRESS and bootstrapping methods, which train all the available data for model development and create a simulated dataset for the assessment stage. Even these methods could have a good advantage in the context when the true distribution is difficult to discern, they suffer from time inefficiency, and when model selection needs the researcher's decision. Kennard and Stone (1969) have suggested the DUPLEX algorithm, for splitting the sample into prediction and estimation subgroups when time is not a variable in the data.

However, sample splitting-based cross-validation is not a new concept; it has taken the attention of the scholar from earlier decades. Stone (1974) stated that cross-validation consists of splitting the data, either in a “controlled “or “uncontrolled” method, into two subsamples, the first one for the statistical estimation and the choice of the predictors; the second will be used for

the prediction's comparison to the first subsample. The very detailed definition of cross-validation using sample splitting could be owed to Mosteller and Tukey (1968).

Herzberg (1969) stated theoretical foundations and empirical examples of applying cross-validation for the measurement of the model accuracy. From a more practical perspective, a study of criminology by F. H. Simon (1971) used cross-validation in the model creation. In an educational research setting, Larson (1931) deployed random sample splitting for cross-validation to study correlation and relationship.

From Observational Data to Causation Inference

The following work is based on Pearl (1995, 2009) and Yao et al. (2021). Pearl's (1995) claims that the idea of using causal explanation, based on the graphical models of the nonparametric structural equations, stems from the econometrics studies of Frisch (1938) and H. A. Simon (1953). In many cases, the causation is mistakenly considered a correlational relationship. However, the correlation describes an association relationship between variables when there is a trend to either increase or decrease. The causation goes a step ahead; it defines a cause-effect between variables, when not only the trend is considered but the change in the conditions of the cause will lead to a change in the effect.

The causation relationship could be deduced effectively based on the experimental designs or the randomized controlled trials. However, in most cases, these studies are not feasible, for instance, due to an unethical issue, financial cost considerations, or being too time-consuming.

For these reasons, considerable attention is shifting to observational data. However, this shift also has challenges in terms of defining the causal effect when the cases have not been randomly assigned to the treatment or because there are no control and treatment groups. To

solve these problems of deducing a causal effect from observational data, models have been developed for these concerns, the Structural Causal Model (SCM), and the Neyman-Rubin Potential outcome framework. This study is concentrating on the SCM.

Identification of the Causal Model

Any causal model will be identifiable with the following fundamental components, the unit, the treatment, the observed output, the potential output, and the counterfactual. Also, three types of variables are defined to help identify the causal effect, the pre-treatment variables, the post-treatment variables, and the treatment effect variables.

The unit is the primary individual case considered in the research. The treatment is the target action of the research deployed on the unit in the study. The observed output is the outcome variable that has been depicted on the unit after it received the treatment. The potential output is the possible outcomes that could have happened if the unit would undertake the treatment. The counterfactual output is the other outcomes seen on the unit taking another treatment that is different from the actual treatment (or not taking any treatment), where the pre-treatment variables and the post-treatment variables are the variables that are not impacted by the treatment and those impacted by the treatment, respectively.

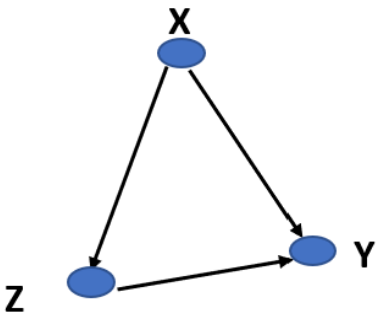
The treatment effect is the measurement that quantifies the causal effect of the treatment on the unit of the study. For instance, an example is defined when the treatment is binary, and it could be extended to different cases easily for more than two potential outcomes. The treatment effect for the whole research population is called the Average Treatment Effect (ATE), which is calculated based on the $\mathrm{Y}(W = 1)$ and $\mathrm{Y}(W = 0)$, the potential outcomes of the causal model identification. For this specific example, they are the treated and control groups' outputs. So, the Average Treatment Effect (ATE) is defined as follows,

$$\text{ATE} = \mathbb{E}[\text{Y}(W = 1) - \text{Y}(W = 0)]$$

The following Figure 2 describes the Directed Acyclic Graph (DAG), proposed by Pearl (2009) to identify the causal effect of the variable Z on the outcome Y given a confounder X that affects both Z and Y.

Figure 2

Directed Acyclic Graph (DAG) of the Causal Relationship.



Machine Learning Methods in the Causality Framework

Recently, causal inference has received wide attention from a variety of applications, such as, in econometrics, social sciences, and medical sciences when they are shifting the inference about the casual effect from the costly experimental design in terms of time and ethical issues to observational data. Furthermore, the use of machine learning methods has helped develop causal models from observational data.

The machine learning tools, such as artificial neural networks (ANN), ensembles, and super learner (Van Der Laan et al., 2007) could deliver a more adaptive estimation and superior prediction behavior than the classical methods due to their high performance compared to the traditional methods. The machine learning techniques can account for the confounder covariates

effects separately, which deploy the estimation of treatment effect in the causal inference model more effectively.

Moreover, machine learning tools could handle high dimensional data where the numbers of the covariates are way larger than the sample size, a characteristic that the statistician couldn't enjoy and have been able to perform a decade ago.

Double Machine Learning Characteristics

This section provides the characteristics of double machine learning founded on the theoretical background and proofs from the work of Chernozhukov et al. (2018), Bach et al. (2021), and Belloni et al. (2017). The construction of the confidence regions of the DML estimators, the variance estimator of the DML causal target parameter, the semiparametric efficiency, the uniformly valid confidence interval of the scaler parameter of DML, and the inference in the partially linear regression model with DML are introduced. All under the identification model of the partial linear regression of the semiparametric causal framework,

$$Y = T\beta_0 + g_0(\mathbf{X}) + U,\ \mathbb{E}[\,U \mid \mathbf{X}, D\,] = 0,$$

$$T = m_0(\mathbf{X}) + V,\ \mathbb{E}[\,V \mid \mathbf{X}\,] = 0.$$

The Construction of the Confidence Regions of the Double Machine Learning Estimators

Two theorems are introduced to help build the confidence regions of the Double Machine Learning estimator (DML1 or DML2). The first theorem emphasizes the asymptotic normality property of the estimator. The second theorem identifies the variance estimator. Before that, the following assumptions for this theorem are set up. Assumption 3 is required to make sure that the score functions are Neyman orthogonal or approximately orthogonal, and mild smoothness. Assumption 3 is also about the quality of the nuisance parameter estimator and the score function regularity condition.

Assumption 1

Approximate Neyman Orthogonality and Linear Scores: **S**uppose the scores functions are linear as follows,

$$\psi(z;\beta,\eta) = \psi^a(z;\eta)\beta + \psi^b(z;\eta), \text{ for all } z \in \mathcal{Z},\ \beta \in \Theta, \eta \in T\ .$$

Let $\{\mathcal{P}_N\}_{N\geq 1}$ a sequence of probability sets distributions P of Z on $\mathcal{Z}$. Consider $\{\Delta_N\}_{N\geq 1}$ and $\{\delta_N\}_{N\geq 1}$ be two convergent sequences that tend to zero and are both sequences of positive constant. The constants c_0, c_1, s , K (fold size), q, are positive, $where\ c_0 \leq c_1,\ K \geq 2\ ,\ q > 2$. Then

$\forall\ N \geq 3\ , \forall\ P \in \mathcal{P}_N$, the true parameter β_0 satisfies,

$$E_P\psi(W;\beta_0,\eta_0)[\eta - \eta_0] = 0.$$

And

the matrix $J_0\ \ has$ singular values $\in [c_0\ , c_1]$,

$$J_0 := E_P[\psi^a(Z;\eta_0)].$$

Also, the score function ψ holds the Neyman orthogonality. Or the score function ψ obeys at (β_0, η_0) the Neyman near-orthogonality condition λ_N with respect to η such that,

$$\lambda_N := \sup_{\eta \in \mathcal{T}_N} \|\partial_\eta E_P\psi(Z;\beta_0,\eta_0)[\eta - \eta_0]\| \leq \delta_N N^{-1/2},$$

where the function $E_P[\psi(W;\theta,\eta)]$ with respect to η is twice Gateaux-differentiable on T.

Assumption 2

The Quality of the Nuisance Parameter Estimator and the Score Regularity: the nuisance parameter estimator convergence rates $\forall\ N \geq 3\ , \forall\ P \in \mathcal{P}_N$, thus, the moment conditions are satisfied,

$$m_N \quad := \sup_{\eta \in \mathcal{T}_N} \left(E_P \left[\|\psi(Z; \beta_0, \eta)\|^q \right] \right)^{1/q} \leq \ c_1 \ ,$$

$$m'_N \quad := \sup_{\eta \in \mathcal{T}_N} \left(E_P \left[\|\psi^a(Z; \eta)\|^q \right] \right)^{1/q} \leq c_1 .$$

Suppose a random fold I $\subset\ [N] = \{1, \ldots, N\}$ of size $n = N/K$, and $\mathcal{T}_N$ is the realization set of the nuisance parameter, thus,

$$\mathrm{P}\ [\ \grave{\eta}_0 = \grave{\eta}_0((W_i)_{i \in I^c}) \in \mathcal{T}_N] \leq 1 - \Delta_N .$$

And the eigenvalues of the following matrix are bounded by c_0, In other words, the score function ψ has a non-degenerate variance,

$$E_P[\psi(\mathrm{Z}; \beta_0, \eta_0)\psi(\mathrm{Z}; \beta_0, \eta_0)'].$$

Also, the next inequalities are satisfied at rates λ'_N , r_N, r'_N, respectively,

$$\lambda'_N := \sup_{r \in (0,1), \eta \in \mathcal{T}_N} \left\| \partial_r^2 E_P \left[\psi\left(Z; \beta_0, \eta_0 + r(\eta - \eta_0)\right) \right] \right\| \leq \delta_N / \sqrt{N}$$

$$r_N \quad := \sup_{\eta \in \mathcal{T}_N} \| E_P[\psi^a(Z; \eta)] - E_P[\psi^a(Z; \eta_0)] \| \leq \delta_N ,$$

$$r'_N \quad := \sup_{\eta \in \mathcal{T}_N} \left(E_P \left[\|\psi(Z; \beta_0, \eta) - \psi(Z; \beta_0, \eta_0)\|^2 \right] \right)^{1/2} \leq \delta_N ,$$

which means that under the assumption 4, and for a chosen value of ε_N such that, $\|\grave{\eta}_0 - \eta\|_T \lesssim \varepsilon_N$ in the realization set $\mathcal{T}_N$, and take $\lambda'_N \lesssim \varepsilon_N^2$, $r_N \lesssim \varepsilon_N$, $r'_N \lesssim \varepsilon_N$, then when considering a special case where $\lambda'_N = o\left(N^{-1/2}\right)$ it will follow that $\varepsilon_N = o\left(N^{-1/4}\right)$. Thus, the nuisance parameter estimator $\grave{\eta}_0$ has $N^{-1/4}$ rate of convergence.

The Asymptotic Normality of the Double Machine Learning Causal Target Estimator

The following theorem shows that the estimator, $\grave{\beta}_0$, based on the orthogonal scores, will reach a convergence of $\sqrt{N}$ rate and will have normal distribution approximately. This distributional approximation and concentration rate are both maintained uniformly in $\mathcal{P}_N$,

where, $\mathcal{P}_N$ is an expanding class of probability measures, $(P_N)_{N\geq 1}$ is a sequence of probability distributions such that for each N, $P_N \in \mathcal{P}_N$, and P is varying over $\mathcal{P}_N$.

Theorem 1

Under assumption 1 and assumption 2, $\forall N$, let $\delta_N \geq 1/\sqrt{N}$. The DML1 estimator $\grave{\beta}_0$ (and the DML2) has the asymptotic normality distribution property with a root- N convergence,

$$\sqrt{N}\sigma^{-1}\left(\grave{\beta}_0 - \beta_0\right) = \frac{1}{\sqrt{N}}\sum_{i=1}^{N} \psi(Z_i) + O_P(\rho_N) \xrightarrow{d} N(0, 1_d),$$

where the approximate variance is

$$\sigma^2 := J_0^{-1} E_P[\psi(Z; \beta_0, \eta_0)\psi(W; \beta_0, \eta_0)'](J_0^{-1})'.$$

And the influence function in this case will be defined by

$$\grave{\psi}(\cdot) := -\sigma^{-1} J_0^{-1} \psi(\cdot, \beta_0, \eta_0).$$

The remainder ρ_N satisfies

$$\rho_N := N^{-1/2} + r_N + r_N' + N^{1/2}\lambda_N + N^{1/2}\lambda_N' \lesssim \delta_N.$$

The Variance of the Double Machine Learning Causal Target Parameter Estimator

Theorem 2

Under the criteria of assumption 1 and assumption 2, $\forall N$, let $\delta_N \geq N^{-[(1-2/q)\wedge 1/2]}$. Then the asymptotic variance matrix of the $\sqrt{N}\left(\grave{\beta}_0 - \beta_0\right)$ is

$$\grave{\sigma}^2 = \grave{J}_0^{-1} \frac{1}{K}\sum_{k=1}^{K} E_{n,k}\left[\psi\left(Z; \grave{\beta}_0, \grave{\eta}_{0,k}\right)\psi\left(Z; \grave{\beta}_0, \grave{\eta}_{0,k}\right)''\right]\left(\grave{J}_0^{-1}\right)',$$

where

$$\grave{J}_0 = \frac{1}{K}\sum_{k=1}^{K} E_{n,k}\left[\psi^a\left(Z; \grave{\eta}_{0,k}\right)\right].$$

And

$$\grave{\sigma}^2 = \sigma^2 + O_P(\varrho_N),$$

$$\varrho_N := N^{-[(1-2/q)\wedge 1/2]} + r_N + r'_N \lesssim \delta_N,$$

which allows to substitute σ^2 by $\grave{\sigma}^2$ with a remainder,

$$\rho_N = N^{-[(1-2/q)\wedge 1/2]} + r_N + r'_N + N^{1/2}\lambda_N + N^{1/2}\lambda'_N.$$

Semiparametric Efficiency

Corollary 1

In general conditions, the semiparametric efficiency of the target estimator is not met, however, special cases exist. Under a semiparametric paradigm (Van der Laan & McKeague, 1998), and if theorem 1 is met, and if the estimator, $\grave{\beta}_0$, is efficient based on the score function ψ at specific $P \in \mathcal{P} \subset \mathcal{P}_N$, where, $\mathcal{P}_N$ is an expanding class of probability measures, $(P_N)_{N\geq 1}$ is a sequence of probability distributions such that $P_N \in \mathcal{P}_N$, P is varying over $\mathcal{P}_N$, and $\mathcal{P}$ is the model, then the variance σ_0^2 of $\grave{\beta}_0$ attains the bounds of the semiparametric efficiency at P relative to $\mathcal{P}$.

Confidence Interval of Scaler Parameter Estimator of Double Machine Learning

Corollary 2

Uniformly Valid Confidence Interval of Scaler Parameter estimator of DML: If the theorem 2 holds, then for some vectors $\ell_{d_\beta \times 1}$, the constructed confidence interval for the scaler parameter $\ell'\beta_0$ will be as follows:

$$\mathrm{CI} := \left(\ell'\grave{\beta}_0 \pm \Phi^{-1}(1-\alpha/2)\sqrt{\ell'\hat{\sigma}^2\ell/N}\right),$$

that satisfies:

$$\sup_{P\in\mathcal{P}_N} |\Pr_P\left(\ell'\beta_0 \in \mathrm{CI}\right) - (1-\alpha)| \to 0,$$

which means that $\forall \{P_N\} \in \mathcal{P}_N$, then the confidence interval obeys also,

$$\Pr_{P_N}(\ell'\beta_0 \in \mathrm{CI}) \to (1-\alpha)\,.$$

Thus, the confidence interval is uniformly valid. So, for instance, if $\epsilon_N \to 0$, then,

$$\sup_{P\in\mathcal{P}_N} \left|\Pr_P(\ell'\beta_0 \in \mathrm{CI}) - (1-\alpha)\right| \le \left|\Pr_{P_N}(\ell'\beta_0 \in \mathrm{CI}) - (1-\alpha)\right| + \epsilon_N \to 0\,.$$

Semiparametric Methods

Semiparametric methods are the methods developed for a class of statistical models that have the parametric and the nonparametric components by adopting assumptions that fully define the distribution (Kosorok, 2009). However, semiparametric models still require minimum structure (Max & Zang, 2019). Specifically, a statistical model is a class of probability measures $\{\mathrm{P},\ \mathrm{P} \in \mathcal{P}\}$ on a sample space χ (Kosorok, 2006). Assume that P is indexed by a parameter space Θ, for each $\theta \in \Theta$, P_θ is specified such that $\mathrm{P} = \{P_\theta,\ \theta \in \Theta\}$. Thus, the statistical model P which is indexed by $\theta \in \Theta$ is considered parametric if $\Theta \subseteq \mathbb{R}^k$, the Euclidean space of k-dimensional for a positive integer k (Bickel et al., 2006). And it is a nonparametric model if the space of the parameters $\Theta \subseteq$ H, where H is an infinite-dimensional space. The statistical models are defined as semiparametric models $\{P_{\theta,\eta} : \theta \in \Theta,\ \eta \in \mathrm{H}\}$ if they have one or more finite-dimensional parameter constituents $\theta \in \Theta$, and one or more infinite-dimensional parameter elements $\eta \in$ H , where H is a space of functions $\theta \in \Theta \subseteq \mathbb{R}^k$ is the parameter of interest, and $\eta \in$ H is the infinite-dimensional nuisance parameter (Bickel et al., 2000; Kosorok, 2006).

For instance, assume the semiparametric regression model $Y = \beta\mathrm{Z} + \epsilon$, where β is the k-dimensional Euclidean space parameter defining the parametric statistical components in the model (Kosorok, 2006). With infinite-dimensional space of all joint functions of (Z, ϵ) with $E[\epsilon/Z] = 0$, and $\mathrm{E}[\epsilon^2/\mathrm{Z}] = 0 < \mathrm{k} < \infty$, almost surely (Kosorok, 2006).

The parametric methods are completely defined in the parameter space (Müller et al., 2004). The fitted model is estimated and explained under restricted parametric assumptions such

as normality, homogeneity of variance, and independent errors. If those assumptions are not met, where in the real-world data they are most likely going to collapse (Hollander et al., 2015), relying in this case on the parametric inference, results will be misrepresentative and inconsistent (Müller et al., 2004). An alternative such as the nonparametric models could be considered, as they provide fewer constrictive assumptions and they tolerate the studied data to adapt the shape of the function accordingly (Rodriguez-Poo & Soberón, 2017). However, in those models, some challenges could be encountered even with this appealing flexibility (Rodriguez-Poo & Soberón, 2017). First, the problem of the models' interpretability complexity could appear due to the curse of the dimensionality that comes from the high-dimensional regressors in the nonparametric settings (Wolfgang et al., 2004). Second, the estimator from the unknown function could have a higher variance (Rodriguez-Poo & Soberón, 2017). These problems were a motivation to adopt a dimension reduction with a parametric model along with keeping the flexible characteristic of the nonparametric components (Wolfgang et al., 2004). The developed framework is a semiparametric model, a combination of the two former models. By that, the advantage of the uncomplicated model's interpretability from the parametric part and the advantage of lessened assumptions from the nonparametric counterpart (Wolfgang et al., 2004). Moreover, even when there is a nonparametric component that tends to have a slow rate of convergence, the acquired estimators from the parametric components display the $\sqrt{\mathrm{N}}$ consistency, the same as if the model is entirely parametric (Robinson, 1988; Speckman, 1988).

The semiparametric models could be described in terms of the tangent space concept (Pfanzagl & Wefelmeyer, 1982). That could produce infinite score functions, where the root of those densities is in Hilbert space.

Reproducing Kernels Hilbert Space

This research study's working space is mapped on the Reproducing Kernels Hilbert Space (RKHS), a space that is based on the kernel methods that map the data into high dimensional feature space using the inner products. The reason for the wide use of the kernel's methods in machine learning is for their computational efficiency, dealing with the high dimensionality of data, and the integration of prior information. Also, the kernel framework provides an appealing result due to their interpretability and simplicity (Marron, 1994). Practically, the positive definite kernels are the basis for the learning in the feature space and for the machine learning estimation framework by finding functional solutions in the reproducing kernel Hilbert space (RKHS). Those functions are defined on the domain of the empirical data and mapped using kernels to high dimensional space that is more representative of the data features (Hofmann et al., 2008; W. Zhang et al., 2010).

Suppose we have empirical data $\{ (x_1, y_1), \dots, (x_n, y_n)\} \in \mathcal{X} \times \mathcal{Y}$. Let us define the following functions k and ϕ as follows (Hofmann et al., 2008; Schölkopf, 2000):

$$k: \mathcal{X} \times \mathcal{X} \to \mathbb{R},$$

$$(x, x') \mapsto k(x, x'),$$

and

$$\phi: \mathcal{X} \to \mathcal{H},$$

and

$$x \mapsto \phi(x),$$

such that

$$k(x, x') = \langle \phi(x), \phi(x') \rangle,$$

Where K is called the kernel function, Φ is the features map function, F is some high-dimensional feature space.

Based on the work of Hofmann et al. (2008) and W. Zhang et al. (2010), the following is the definitions of the reproducible kernel (R.K) and the reproducing kernel Hilbert space (RKHS), followed by a display of the kernel matrix definition and its positive definite matrix characteristics (PDK). Those definitions are descriptions and are a necessary background to introduce the theorems of the Mercer's Kernels and their relationship with the kernels being positive definite and the reproducing kernel Hilbert space (RKHS).

Definition 2

Reproducible Kernels: The function $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducible kernel (R.K) of a Hilbert space $\mathcal{H}$, if the following conditions are met

1. $\forall x,\ k_x(y) = k(y, x)$
2. $\forall x \in \Omega$, and $\forall f \in \mathcal{H},\ f(x) =< f, K_x >$
3. Given a Hilbert space $\mathcal{H}$, then

$$\mathcal{H} = \overline{\text{span}\,\{K_x(\cdot) \mid x \in \Omega\}}\,,$$

which means that the Hilbert space $\mathcal{H}$ is spanned by $K: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

Noting that a reproducible kernel (R.K) holds the characteristics of building new complex reproducible kernels (R.K) based on products and sums of simpler reproducible kernels.

Definition 3

Reproducing Kernel Hilbert Space: Let $\mathcal{H}$ be a Hilbert space of function $f: \mathcal{X} \to \mathbb{R}$, where $\mathcal{X} \subseteq \mathbb{R}^d$. Then $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) if the function $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfies the conditions stated above.

Definition 4

Gram Matrix: Let a function $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function with input $(x_1, x_2, \dots, x_n)$. The following n × n matrix is called a Gram matrix or a kernel matrix of k with respect to $(x_1, x_2, \dots, x_n)$,

$$K := (k(x_i, x_j))_{ij}$$

Definition 5

Positive Definite Kernel**:** Let the function $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with input $(x_1, x_2, \dots, x_n)$ be a kernel function, where the corresponding kernel matrix (Gram matrix) is positive definite matrix as follows as,

$$\forall c_i \in \mathbb{R}, \ \sum_{i,j} c_i c_j K_{ij} \geq 0,$$

Then we call the function $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ as positive definite kernel (PDK).

Theorem 3

Let $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function. K is called a Mercer kernel iff K is a positive definite kernel (PDK).

Theorem 4

Let $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function. K is a Mercer kernel iff there exists a reproducible kernel Hilbert space (RKHS) $\mathcal{H}$ with reproducible kernel K.

According to Schölkopf and Smola (2018), the kernels set K enjoy both properties of a convex cone set and a closed set under pointwise convergence as follows.

Proposition 1

Sums of Kernels: Let K be the set of the kernels. For each k_1 and k_2 and $\alpha_1, \alpha_2 \geq 0$, then $\alpha_1 k_1 + \alpha_2 k_2$ is a kernel. Which means that K is a convex set.

Proposition 2

Limits of Kernels: Let K be the set of kernels. Consider k_1, k_2,…, k_n, be kernels from the set *K*. If

$$k(x, x') := \lim_{n\to\infty} k_n(x, x') \text{ exists for all } x, x',$$

then k is a kernel, which means that K is closed set under pointwise convergence.

The examination of the literature review has indicated the importance of causal inference in observational studies and how the current research direction in causal inference is collaborating with machine learning methods.

Because using machine learning procedures will help address the curse of the dimensionality that our era is witnessing as the big data era, which will help deliver a quality causal estimator compared to the counterpart of the traditional methods where they collapse in front of the high dimensionality of the covariates. The previous studies also denoted the importance of the choice of sample splitting in the process of statistical estimation or statistical inference.

Based on this literature review, this research proposes a new causal inference model that has not been studied before, using double machine (DML) learning tools such as support vector machines (SVM), deep learning (DL), and super learner (SL), along with the support points sample splitting (SPSS).

CHAPTER III

METHODOLOGY

This chapter is dedicated to investigating the performance of the methods, super vector machine (SVM) with double machine learning (DML) using support points sample splitting (SPSS), deep learning (DL) with double machine learning (DML) using support points sample spitting, and hybrid of super learner and deep learning with the double machine learning using support points ample splitting. First, the optimal data splitting method, the support points subsampling, is introduced. The theoretical foundation is presented to show that this technique is an optimal splitting for the sample. To the best knowledge of the researcher, this could be considered as a new addition to the body of knowledge and a state-of-art sample splitting method that could best represent the sample. Second, the models under study are presented, to which the machine learning estimators are applied: Support vector machine and deep learning. Third, a new ensemble method is introduced that is a hybrid of the super learner, deep learning, and support points splitting. Fourth, the double machine learning for causal inference is described (Chernozhukov et al., 2018) with the sample splitting technique to construct the target estimator after the estimation of the nuisances. Finally, the chapter is concluded with the simulation scheme, and the empirical socio-economics example is used as a demonstration, of the 401(k) plan, a dataset used to estimate the effect of the eligibility of this plan on the financial assets.

Support Points Sample Splitting

This section starts by introducing the sample splitting with the support points. The following definition and assumptions are customized to this study (Joseph &Vakayil, 2021; Székely & Rizzo, 2013).

Definition 6

Support-Points Sample Splitting: Suppose that a sample unit data structure $S = \{(\boldsymbol{U}_i\ , Y_i)\ \}_{i=1}^{N}$ that consists of the predictor $\boldsymbol{U} = (\mathrm{T}, \boldsymbol{X})$ of dimension p, where T is the causal target (treatment) variable, and response *Y*. The aim is to perform the sample splitting with the support points method and divide the data into two mutually exclusive and disjoint sets of $\mathcal{S}$, a training set $\mathcal{S}_1$ and test set $\mathcal{S}_2$ such that $N = \mathrm{n}_{train} + \mathrm{n}_{test} = \mathrm{card}\,(\mathcal{S}_1) + \mathrm{card}(\mathcal{S}_2)$,

$$\mathcal{S}\ = \mathcal{S}_1 \cup \mathcal{S}_2\ ,\ (\mathcal{S}_1)^c = \mathcal{S}_2\,.$$

Assumption 3

Assume that the samples come from a distribution $G,$ and they are independent and identically distributed, that is:

$$(\boldsymbol{U}_i\ , Y_i) \overset{iid}{\sim} G, i = 1, \dots, N.$$

Assumption 4

Let $H(\mathbf{U};\ \boldsymbol{\theta})$ be the adaptive predictor from the dataset, $\boldsymbol{\theta}$ is the parameter vector to be estimated from the loss function $L\ (Y, H\ (\mathbf{U};\ \boldsymbol{\theta}))$. Take the loss function as the squared or absolute error loss, or the negative predictor log-likelihood. The wish is that the adaptive predictor $H\ (\mathbf{U};\ \boldsymbol{\theta})$ is near to the true predictor E (Y | **U**) under some specific $\boldsymbol{\theta}$. So, take the training sample to train multiple predictive models and then test their performance. The unknown vector parameter could be estimated by,

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}}\ L\left(Y_i^{\text{train}}, H\left(\mathbf{U}_{\text{i}}^{\text{train}};\ \boldsymbol{\theta}\right)\right).$$

Given the training dataset,

$$\left(\mathbf{U}_{\mathrm{i}}^{\text{train}}, Y_i^{\text{train}}\right) \sim G, i = 1, \ldots, n_{\text{train}}.$$

The performance of the models could be evaluated by calculating the generalization error (Hastie et al., 2009),

$$\mathcal{E} = E_{\mathbf{U},Y}\left\{L(Y, H(\mathbf{U}; \widehat{\boldsymbol{\theta}})) \mid \mathcal{S}^{\text{train}}\right\}.$$

And given that the testing dataset is from,

$$(\mathbf{U}_i^{\text{test}}, Y_i^{\text{test}}) \sim G, i = 1, \ldots, n_{\text{test}},$$

estimate this error from the testing set $\mathcal{S}^{\text{test}}$,

$$\hat{\mathcal{E}} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} L\left(Y_i^{\text{test}}, H\left(\mathrm{U}_i^{\text{test}}; \widehat{\boldsymbol{\theta}}\right)\right).$$

Thus, the estimation $\hat{\mathcal{E}}$ will be a Monte Carlo (MC) estimator which decreases at a rate of $\mathcal{O}\left(1/\sqrt{N_{test}}\right)$. However, Mak and Joseph (2018) introduced the support points method for sample splitting with a Quasi-Monte Carlo (QMC) sample. This method could improve the estimation of $\mathcal{E}$ with a faster convergence rate of $\mathcal{O}(1/N_{\text{test}})$. Furthermore, it could be applied on a sample from a general distribution not only limited to the uniform distribution (Niederreiter, 1992).

The Energy Distance

Definition 7

Assume $\mathbf{V} = (\mathbf{U}, Y)$ is a continuous variable. The energy distance between the empirical distribution of points $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}$ and the distribution $G(\mathbf{V})$ is described as follows,

$$ED = \frac{2}{n}\sum_{i=1}^{n} \mathbb{E}\|\mathbf{v}_i - \mathbf{V}\|_2 - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \|\mathbf{v}_i - \mathbf{v}_j\|_2 - \mathbb{E}\|\mathbf{V} - \mathbf{V}'\|_2 ,$$

where $\|.\|_2$ is the Euclidean distance. $\mathbf{V}, \mathbf{V}'$are both distributed as the distribution *G.* And all the expectation has been taken with respect to *G* taking into consideration that all variables should be standardized with mean zero and the unit standard deviation to calculate the Euclidean distance. Mak and Joseph (2018) have noted that the energy distance will be small in the case $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}$ are close to *G.* So, they have expressed the minimizer of the energy distance to be the support points definition as follows,

$$\{\mathbf{v}_i^*\}_{i=1}^n \in \underset{\mathbf{v_1},\mathbf{v_2},\ldots,\mathbf{v_n}}{\operatorname{argmin}} ED = \underset{\mathbf{v_1},\mathbf{v_2},\ldots,\mathbf{v_n}}{\operatorname{argmin}} \left\{ \frac{2}{n} \sum_{i=1} \mathbb{E}\|\mathbf{v}_i - \mathbf{V}\|_2 - \frac{1}{n^2} \sum_{i=1} \sum_{j=1} \|\mathbf{v}_i - \mathbf{v}_j\|_2 \right\}$$

The Support Points Sample Splitting:
An Optimal Adaptive Learning

First, one of the characteristics of the support points is that the expectation in the support points equation could be substituted with the Monte Carlo average that is computed from $S = \{(\boldsymbol{U}_i\ , Y_i)\ \}_{i=1}^N$, the sample set of interest (Joseph & Vakayil, 2021). This substitution is designed to solve the difficulty of not having the exact distribution of *G*, which makes the support points a flexible data adaptive technique. Thus, the updated formula of the support points will be,

$$\{\mathbf{v}_i^*\}_{i=1}^n \in \underset{\mathbf{v_1},\mathbf{v_2},\ldots,\mathbf{v_n}}{\operatorname{argmin}} \left\{ \frac{2}{nN} \sum_{i=1}^{n} \sum_{j=1}^{N} \|\mathbf{v}_i - \mathbf{V}_j\|_2 - \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|\mathbf{v}_i - \mathbf{v}_j\|_2 \right\} .$$

Second, the support points method is regarded as the best n points set that could represent the data distribution *G* based on the energy distance criteria (Mak & Joseph, 2018). It outperforms the other points splitting techniques such as the principal points method defined by Flury (1990), and *MSE*-rep method introduced by Fang and Wang (1994). Precisely, the support points converge in distribution to *G*, which makes it as a QCM sample for *G.* where the two other methods do not have this property.

The idea that the support points sample splitting is an optimal adaptive subsample to a dataset is addressed in this section and that is well-representative of the underlying distribution of this dataset. In the following, consider the work in some probability measure space ($\Omega, \mathcal{F}, P$).

Lemma 1

Consider a sequence $\{X_j\}_{j=1}^{n}$ of random variables (R.V.'s) and a subsequence $\{S_{ij}\}$ of support points with the distribution function (DF) G_n such that:

$$\mathbf{X}_n \sim G_n ,$$

$$\mathbf{S}_n \sim G_n ,$$

$$\mathbf{X} \sim G.$$

Suppose $\varphi_n(t)$ and $\varphi(t)$ are the characteristic functions of $\mathbf{S}_n$ and $\mathbf{X}$ respectively. If

$$\lim_{n\to\infty} \varphi_n(t) = \varphi(t) ,$$

then

$$\mathbf{S}_n \xrightarrow{d} \mathbf{X} .$$

Proof

This lemma could be verified using Halley's theorem and the Cramér-Lévy theorem.

Proposition 3

The Existence of a Convergent Subsequence: Consider $\{X_j\}_{j=1}^{n}$ a sequence of R.V.'s of sample where $\mathbf{X}_n \xrightarrow{d} \mathbf{X}$**.** Thus, there exists a subsequence $\{S_{ij}\}$ such that $\mathbf{S}_n \xrightarrow{d} \mathbf{X}$, which means that $\varphi_n(t) \rightarrow \varphi(t)$, and one of these subsequences is the support points subsequence.

Proof

Halley's theorem could be used to certify the existence of such subsequences. The following theorem1, theorem 2, theorem 3, and theorem 4, show that one of these subsequences is the support points subsample that satisfies the conditions of this proposition.

Theorem 5

Assume $\{S_{ij}\}$ is a sequence of independent and identically distributed (iid) support points-based R. V.'s. Allow $\breve{G}_n$, G indicate the empirical distribution function (EDF) and limiting distribution function respectively (DF) with the corresponding characteristic functions $\breve{\varphi}_n(t), \varphi(t)$, then

$$\lim_{n\to\infty} \breve{\varphi}_n(t) \quad = \quad \varphi(t) \, ,$$

and

$$\lim_{n\to\infty} E\left[\, |\breve{\varphi}_n(t) \quad - \ \varphi(t) \,)|^2 \, \right] = 0 \, .$$

Proof

Given the EDF $\breve{G}_n \ (x) = \frac{1}{n} \sum_{k=1}^{n} \ I(X_k \leq x)$, then by the Glinvenko-Cantelli Lemma, the following holds,

$$\sup_{x} \left| \breve{G}_n(x) - \ G(x) \, \right| \to 0$$

$$\Longrightarrow \ \breve{G}_n \ (x) \to \ G(x) \, .$$

By the Cramér- Lévy theorem, it be deduced that,

$$\lim_{n\to\infty} \ \breve{\varphi}_n(t) = \ \varphi(t), \text{ on any finite } |t| \in \ T \quad (*).$$

Also,

$$\left| e^{i \, xt} \right| \ \leq \ 1$$

$$\Longrightarrow |\breve{\varphi}_n(t)| \ \leq 1$$

$$\Longrightarrow \ |\breve{\varphi}_n(t) - \ \varphi(t) \, |^2 \ \leq c \, , \ \ c \ is \ constant \, .$$

As the $|\breve{\varphi}_n(t) \ - \ \varphi(t) \, |^2$ is bounded, then by the Portmanteau theorem, the equation (*) will be as follows,

$$\lim_{n\to\infty} E \ [\, | \, \breve{\varphi}_n(t) \ - \ \varphi(t) \,)|^2] = 0.$$

Theorem 6

Assume $\{S_{ij}\}$ is a sequence of independent and identically distributed (iid) support points-based R. V.'s. Allow $\check{G}_n$, G to indicate the empirical distribution function (EDF) and the limiting distribution function respectively (DF) with the corresponding characteristic functions $\breve{\varphi}_n(t)$, $\varphi(t)$. Let $E_d(\check{G}_n \ , G \ )$ is the energy distance. Thus, the following holds:

$$\lim_{n\to\infty} E\left[\ E_d \ (\check{G}_n \ , G \ )\right] = 0,$$

where the definition of the energy distance is defined by Székely and Rizzo (2013),

$$E_d\big(\check{G}_n, G \ \big) = \frac{1}{K_p}\int \ \frac{|\breve{\varphi}_n(t) \ - \varphi(t) \ )|^2}{\|t\|_2^{p+1}} \ dt,$$

where

$$K_p = \ \frac{\pi^{p+1}}{\Gamma(\frac{p+1}{2})}.$$

Proof

The energy distance definition is as follows:

$$E_d(\check{G}_n \ \ , G \ ) = \frac{1}{K_p}\int \frac{|\breve{\varphi}_n(t) \ - \varphi(t) \ )|^2}{\|t\|_2^{p+1}} \ dt,$$

where $E_d(\check{G}_n \ , G \ ) < \ \infty$ (Székely & Rizzo, 2013)

$$\Longrightarrow E\left\{ \ E_d(\check{G}_n \ , G \ )\right\} = \ E\left\{ \ \ \frac{1}{K_p}\int \frac{|\breve{\varphi}_n(t) - \varphi(t) \ )|^2}{\|t\|_2^{p+1}} \ dt \ \ \right\}.$$

By Fubini theorem

$$E\left\{ \ E_d\big(\check{G}_n \ , G \ \big)\right\} = \ \ \frac{1}{K_p}\int \frac{E[|\breve{\varphi}_n(t) \ - \ \varphi(t) \ )|^2]}{\| \ t \ \|_2^{p+1}} \ dt.$$

That implies the following by the dominated convergence theorem (DCT)

$$\lim_{n\to\infty} E\{\ E_d(\check{G}_n\ , G\ )\} = \lim_{n\to\infty}\ \frac{1}{K_p}\int \frac{E[|\breve{\varphi}_{\mathrm{n}}(\mathrm{t})\ -\ \varphi(t)\ )|^2]}{\|\ t\ \|_2^{p+1}}\ dt$$

$$= \frac{1}{K_p}\int\ \lim_{n\to\infty} \frac{E[|\breve{\varphi}_{\mathrm{n}}(\mathrm{t})\ -\ \varphi(t)\ )|^2]}{\|\ t\ \|_2^{p+1}}\ dt\ .$$

By theorem 5,

$$\lim_{n\to\infty} \mathrm{E}[|\breve{\varphi}_{\mathrm{n}}(\mathrm{t})\ -\ \varphi(\mathrm{t})\ )|^2] = 0.$$

Thus,

$$\lim_{n\to\infty} E\ [\ E_d(\check{G}_n\ , G\ )]\ =\ 0\ .$$

Theorem 7

Assume $\{S_{ij}\}$ is a sequence of independent and identically distributed (iid) support points-based R.V.'s. Allow $G_n,\ \check{G}_n\ , G$ indicate the cumulative distribution function (CDF), the empirical distribution function (EDF) and the limiting distribution function (DF) respectively. Consider the corresponding characteristic functions $\varphi_n(t),\ \breve{\varphi}_{\mathrm{n}}(\mathrm{t}),\ \varphi(\mathrm{t})\ ,$ and the $E_d(G_n\ , G\ )$ be the energy distance. Thus, the following holds:

$$\lim_{n\to\infty}\ \varphi_n(t) =\ \varphi(\mathrm{t}).$$

Proof

From Mak and Joseph (2018), the energy distance satisfies the following property:

$$0 \le E_d(G_n, G\ ) \le E\ [E_d(\check{G}_n\ , G\ )].$$

And from theorem 6,

$$\lim_{n\to\infty} E\ [\ E_d(\check{G}_n\ , G\ )]\ =\ 0.$$

Thus,

$$0 \leq E_d(G_n, G\ ) \leq \lim_{n\to\infty} E\ [\ E_d(\check{G}_n\ , G\ )]\ =\ 0$$

$$\Longrightarrow E_d(G_n, G\ ) = 0\ .\ \ldots(\text{i})$$

By the definition of the energy distance,

$$E_d(G_n\ , G\ ) = \frac{1}{K_p}\int \frac{|\varphi_n(t) - \varphi(t)\ )|^2}{\|\ t\ \|_2^{p+1}}\ dt$$

$$\Longrightarrow\ \lim_{n\to\infty} E_d(G_n\ , G\ ) = \lim_{n\to\infty} \frac{1}{K_p}\int \frac{|\ \varphi_n(t) - \varphi(t)\ )|^2}{\|t\|_2^{p+1}}\ dt.$$

By the dominated convergence theorem, the following holds,

$$\lim_{n\to\infty} E_d(G_n\ , G\ ) = \frac{1}{K_p}\int \lim_{n\to\infty} \frac{|\ \varphi_n(t) - \varphi(t)\ )|^2}{\|\ t\ \|_2^{p+1}}\ dt. \quad (ii)$$

From (i) and (ii) that implies,

$$\lim_{n\to\infty} [|\varphi_n(t)\ -\ \varphi(\mathrm{t})\ )|^2] = 0.$$

Then

$$\lim_{n\to\infty}\ \varphi_n(t) =\ \varphi(\mathrm{t}).$$

Theorem 8

Let the sequences $\{\mathrm{X_j}\}_{\mathrm{j=1}}^{\mathrm{n}}$, $\{\mathrm{S_{ij}}\}$ be the sample and the support points-based subsample of random variables (R.V.'s) respectively, such that,

$$\mathbf{X}_{\mathrm{n}} \sim G_n\ ,$$

$$\mathbf{X}_{\mathrm{n}} \xrightarrow{d} \mathbf{X} \sim G\ ,$$

$$\mathbf{S}_{\mathrm{n}} \sim G_n.$$

Thus,

$$\mathbf{S}_{\mathrm{n}} \xrightarrow{d} \mathbf{X} \sim G.$$

Proof

From proposition 1, theorem 1, theorem 2, and theorem 3, previously introduced, it could be concluded that the $\{S_{ij}\}$, the sequence of random variables of the support points satisfies,

$$\lim_{n\to\infty} \varphi_n(t) = \varphi(t).$$

Thus, by lemma 1 that implies:

$$\mathbf{S}_n \xrightarrow{d} \mathbf{X} \sim G.$$

Corollary 3

Let $\{X_j\}_{j=1}^{n}$, $\{S_{ij}\}$ Sequences of the sample and the support points subsample respectively of random variables (R.V.'s) such that,

$$\mathbf{X}_n \sim G_n \ ,$$

$$\mathbf{X}_n \xrightarrow{d} \mathbf{X} \sim G \ ,$$

and

$$\mathbf{S}_n \sim G_n.$$

Suppose f is continuous functions such that f: $(\Omega, \mathcal{F}, P) \to \mathbb{R}$, thus

$$f\ (\mathbf{S}_n) \xrightarrow{d} f(\mathbf{X}).$$

Proof

This corollary could be proved using theorem 4 and the continuous mapping theorem.

Corollary 4

Suppose the sequence $\{X_j\}_{j=1}^{n}$, $\{S_{ij}\}$ of random variables (R.V.'s) of the sample and the support points subsample, respectively, such that

$$\mathbf{X}_n \sim G_n \ ,$$

$$\mathbf{X}_n \xrightarrow{d} \mathbf{X} \sim G \ ,$$

and

$$\mathbf{S}_{\mathrm{n}} \sim G_n.$$

Suppose f is continuous and bounded function such that f: ($\Omega, \mathcal{F}, P$) $\rightarrow \mathbb{R}$, thus

$$\lim_{n\to\infty} E\,[\,\mathbf{f}\,(\mathbf{S}_{\mathrm{n}})\,] \;=\; \mathbf{E}\,[\,\mathbf{f}\,(\mathbf{X})\,].$$

Proof

This corollary could be proved using the portmanteau theorem.

Comparison Between the Support Points Sample Splitting and Random Splitting

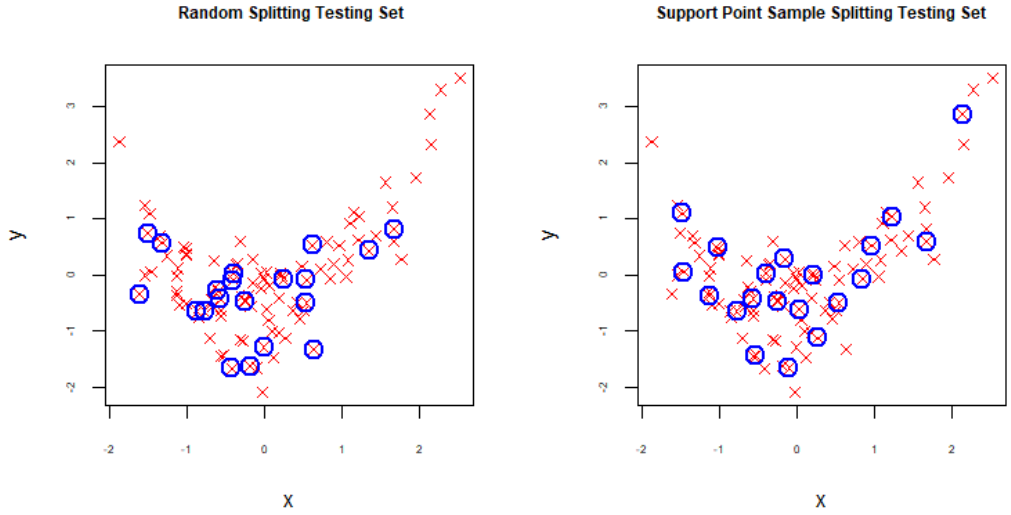
Joseph and Vakayil (2021) have stated that the application of the support points for splitting the dataset into training and testing subsets has shown an optimal result versus the counterpart method of the random splitting. Empirically, consider taking the training set larger than the testing set, so, it will be more computationally efficient to create the testing set first. By implementing the equation stated earlier and taking n = N_{test}, thus,

$$\{\mathbf{v}_i^*\}_{i=1}^n \in \underset{\mathbf{v_1},\mathbf{v_2},\ldots,\mathbf{v_n}}{\mathrm{Argmin}} \left\{ \frac{2}{nN} \sum_{i=1}^n \sum_{j=1}^N \left\| \mathbf{v}_i - \mathbf{V}_j \right\|_2 - \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \left\| \mathbf{v}_i - \mathbf{v}_j \right\|_2 \right\}.$$

From the following Figure 3, observe the visualization of the testing set of both support points sample splitting and random splitting, where the support points splitting set is noticeably more representative of the original dataset than the random sample splitting set, which will deliver a much better estimation and inference accuracy (Székely & Rizzo, 2013).

Figure 3

Empirical Comparison Between the Random and Support Points-Based Splitting



The Validation Sets are Optimal with Support Points Sample Splitting

Definition 8

The Optimal Validation Set: Let $\{\mathbf{v}_i^*\}_{i=\mathbf{N}_{test}+1}^{n}$ be the validation set and let $\{\mathbf{v}_i^*\}_{i=1}^{\mathbf{N}_{test}}$ be the testing set such that $\mathbf{N}_{vali} + \mathbf{N}_{test} = n$. Thus, the identification of an optimal validation points set that are away from the testing set, by using the support points technique based on the energy distance (Joseph & Vakayil, 2021) will be as follows,

$$\{\mathbf{v}_i^*\}_{i=\boldsymbol{N}_{test}+1}^{n} \in \underset{\mathbf{v}_{\mathbf{N}_{test}},\ldots,\mathbf{v}_{\boldsymbol{n}} \in \boldsymbol{D}}{\operatorname{argmin}} \left\{ \frac{2}{nN} \sum_{\mathrm{i}=\mathbf{N}_{\mathrm{test}}+1}^{\mathrm{n}} \sum_{j=1}^{N} \left\| \mathbf{v}_i - \mathbf{V}_j \right\|_2 - \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left\| \mathbf{v}_i - \mathbf{v}_j \right\|_2 \right\}.$$

This support points splitting could be used as an optimal method for data splitting in a specific situation (Vakayil & Joseph, 2022), such as where the hyper parameter estimation is needed (Joseph & Vakayil, 2021). The semiparametric causal inference with double machine learning models requires the estimation of the nuisance parameters where they are counted as hyperparameters, and in applications they are typically high dimensional. Thus, applying the

support points splitting will be more useful for the double machine learning estimation of the nuisance parameters than the random technique.

Taking into consideration that machine learning techniques such as, SVM, DL, Lasso, regression trees, and random forests are more efficient machine learning techniques in estimating the hyperparameter, this study adopt the SVM and DL as the methods for the causal inference in the double machine learning settings.

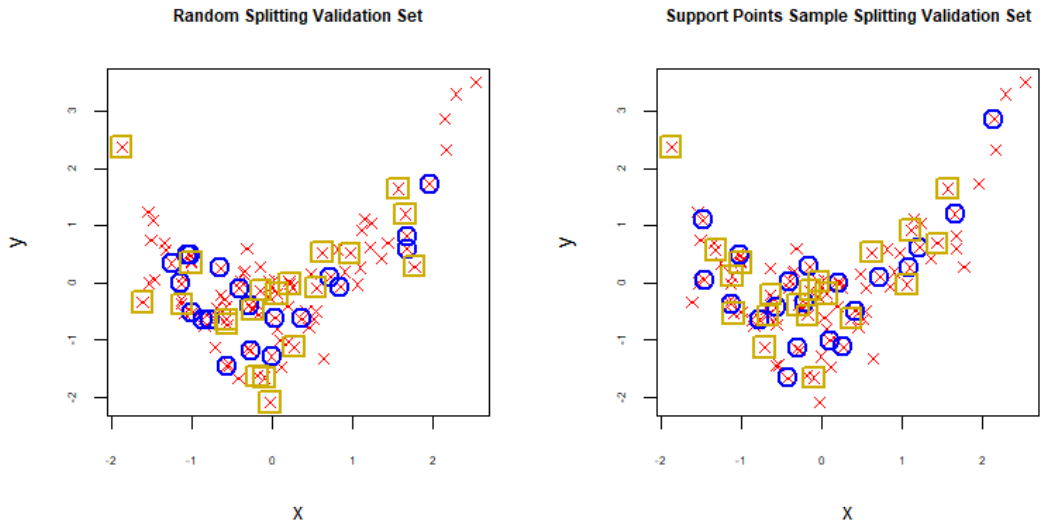
Two optimal characteristics of the validation's points set could be depicted using support points splitting. First, note that the importance of the validation set comes from their role to be used in estimating the hyperparameters. As the empirical examples shows, the validation sets using the support splitting do not intercept with the testing sets as much as in the random splitting because of the second term in the energy distance optimization $\frac{1}{n^2}\sum_{i=1}^{n}\ \sum_{j=1}^{n}\ \left\|\mathbf{v}_i - \mathbf{v}_j\right\|_2$ that attest that the validation set points are distant from the testing points set. Thus, the bias that could arise from testing the model from a subset that is near to the training/validation sets is handled better with the support points splitting than the random splitting.

Second, the validation sets, shown in squares, obtained from the support points splitting, are optimal in the representation of the original data than the validation sets generated from random splitting validation, which will ensure a better estimation of the hyperparameters of the nuisance parameters in this framework of the semiparametric causal double machine learning. That means that the support points splitting method is more adaptive learning than the counterpart of random splitting.

Figure 4 shows the optimal characteristics of the validation points set using support points versus the random sample. The validation and testing sets are in squares and in circles respectively, from Joseph and Vakayil (2021).

Figure 4

The Validation Points Set Using Support Points Versus the Random Sample



Support Vector Machine

Smola and Schölkopf (2004) developed the support vector machine (SVM) for regression based on VC theory (Vapnik, 1982, 1995; Vapnik & Chervonenki, 1974), which is a theoretical foundation that initially introduced the support vector machine, a method that is considered as kernel-based (Che & Wang, 2014).

In this framework, consider the data defined by $\mathcal{D} = \{(x_i, y_i)\ (\mathbf{x_i}, y_i) \in \mathbb{R}^d \times \mathbb{R}\}$, and Let F (x, $\boldsymbol{w}$) be the estimated distribution function that is parameterized by $\boldsymbol{w}$, where $\widehat{\boldsymbol{w}}$ quantify the error between F (x, $\widehat{\boldsymbol{w}}$), and G(x) the true distribution function.

Definition 9

The Support Vector Machine: The support vector machine (SVM) has a good application in economics in terms of restricting the tolerable amount in investments. However, The SVM is a model with ε-support vector, a ε – radius cylinder that allows the utmost value between the fitted line and the data. In other words, ε −support vector identifies a loss function that equals to

zero if the fitted value is within the ε-radius cylinder (Drucker et al., 1997; Smola & Schölkopf, 2004; Vapnik, 1995) as follows,

$$\mathcal{L} = \begin{cases} |y_i - F_2(\boldsymbol{x}_i, \grave{\boldsymbol{w}})| - \varepsilon, & if \quad |y_i - F_2(\boldsymbol{x}_i, \grave{\boldsymbol{w}})| - \varepsilon > 0 \\ & \\ 0, & or \quad elsewhere \end{cases}$$

The Lagrange method is used to find the minimization solution value to the loss function

$$\mathcal{L} = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{k} (\xi_i + \xi_i^*) - \sum_{i=1}^{k} (\eta_i \xi_i + \eta_i^* \xi_i^*) - \alpha_i(\varepsilon + \xi_i - y_i + \langle w, x_i \rangle + b)$$

$- \sum_{i=1}^{\ell} \alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle w, x_i \rangle - b),$

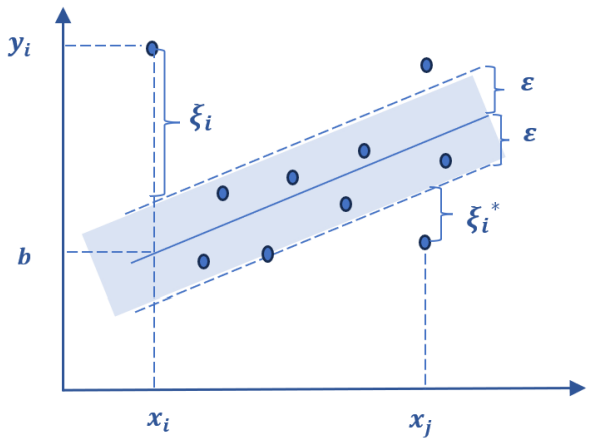
where: $\eta_i^*, \eta_i,\ \alpha_i^*, \alpha_i$ are Lagrange multipliers, ξ_i^*, ξ_i are slack variables (Cortes & Vapnik, 1995). After the Lagrange problem is solved (Smola & Schölkopf, 2004), then,

$$w = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) x_i,$$

which is the expansion equation of the support vector machine. To summarize the concept of the support vector machine (SVM) for regression based on Smola and Schölkopf (2004), Figure 5 illustrate this work geometrically as follows,

Figure 5

Illustration of Support Vector Machine



Deep Learning

Definition 10

Deep Learning: The following description and definitions are based on the work of A. Zhang et al. (2023) and Goodfellow et al. (2016).

Deep learning is an artificial intelligence framework based on a multilayer perceptron (MLP) model; an advanced supervised learning paradigm composed of neurons organized in a couple of layers. Each neuron of a layer is linked to the previous and the next layer neurons to form a multiple layer's neural network architecture. This is different from the simple neural network that is composed of only one single layer. Figure 6 illustrates an example of a deep learning diagram, which has more than one hidden layer (Two layers), an input layer, and an output layer. In construct, Figure 7 illustrates a shallow deep learning or a neural network that has only one hidden layer, an input layer, and an output layer.

Figure 6

Deep Learning Diagram

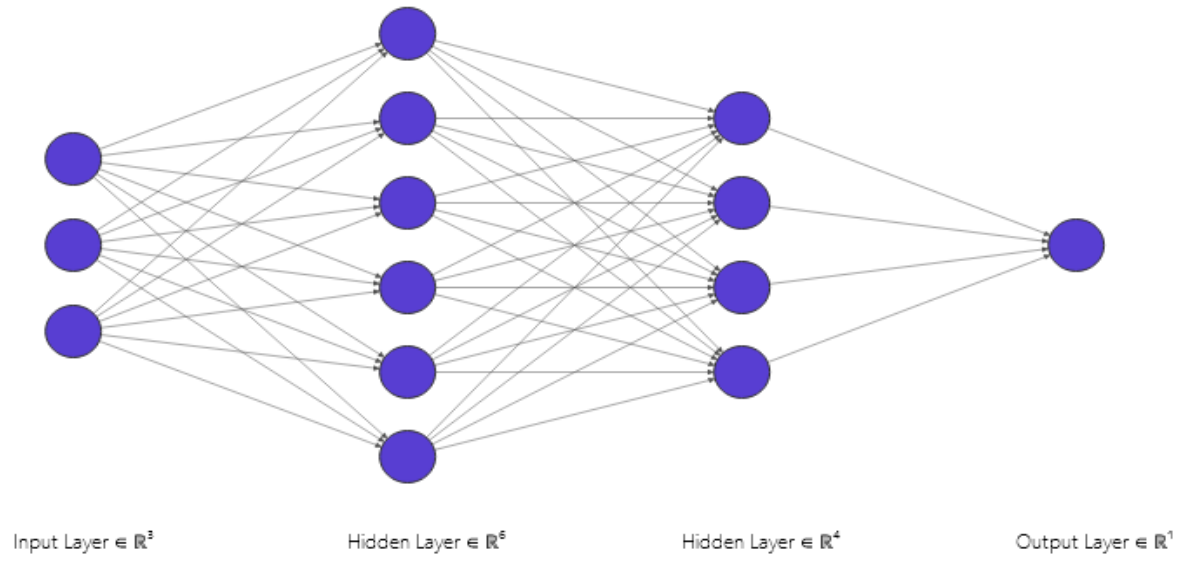
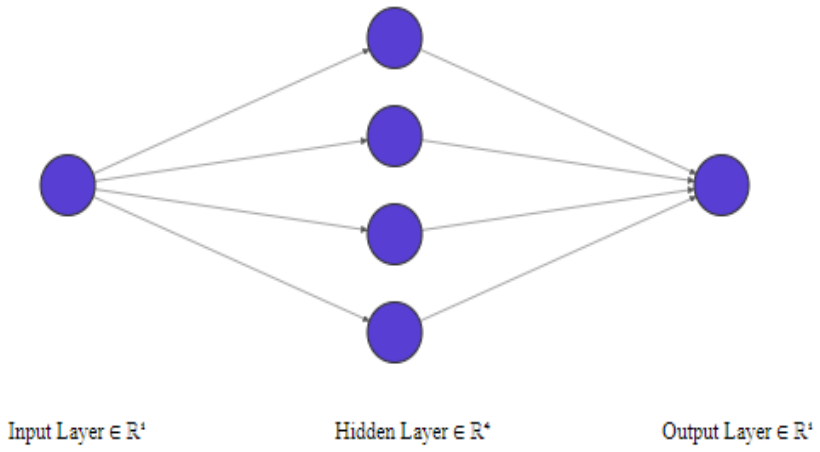


Figure 7

Neural Networks diagram,



Deep learning has offered a good result to many problems in observational data compared to the counterpart of simple neural network learning such as in the speech recognition and the image processing fields. It is based on the feedforward learning flow, a learning chain that starts from an input $\mathbf{X} \in \mathbb{R}^{n\times d}$ to an output $\mathbf{Y} \in \mathbb{R}^{n\times y}$ through multiple hidden layers, conceptually, through a combination of functions. For instance, suppose there are three hidden layers of depth in the following deep learning model, thus, three functions would be defined in this structure, $g^{(1)}$, $g^{(2)}$, and $g^{(3)}$ such that they represent layer 1, layer 2, and layer 3 respectively. This chain structure could be defined as the functional composition,

$$g^{(3)} \circ g^{(2)} \circ g^{(1)} = g^{(3)}((g^{(2)}(g^{(1)}))).$$

In the deep learning model introduced earlier, the output equations of each hidden layers are defined as follows, where $\mathbf{W}^{(i)} \in \mathbb{R}^{d\times \ell i}$, $\mathbf{b}^{(i)} \in \mathbb{R}^{d\times qi}$ are each layer weights, and the bias of each layer respectively,

$$\mathbf{L}^{(1)} = \boldsymbol{X}$$

$$\mathbf{L}^{(2)} = \mathbf{L}^{(1)}\mathbf{W}^{(2)} + \mathbf{b}^{(2)},$$

$$\mathbf{L}^{(3)} = \mathbf{L}^{(2)}\mathbf{W}^{(3)} + \mathbf{b}^{(3)},$$

$$\mathbf{L}^{(4)} = \mathbf{L}^{(3)}\mathbf{W}^{(4)} + \mathbf{b}^{(4)},$$

$$\boldsymbol{Y} = \mathbf{L}^{(5)} = \mathbf{L}^{(4)}\mathbf{W}^{(5)} + \mathbf{b}^{(5)}.$$

Thus, by substituting and performing the functional combination definition,

$$\Longrightarrow \boldsymbol{Y} = (\mathbf{L}^{(3)}\mathbf{W}^{(4)} + \mathbf{b}^{(4)})\mathbf{W}^{(5)} + \mathbf{b}^{(5)}$$

$$\Longrightarrow \boldsymbol{Y} = [(\mathbf{L}^{(2)}\mathbf{W}^{(3)} + \mathbf{b}^{(3)})\mathbf{W}^{(4)} + \mathbf{b}^{(4)}]\mathbf{W}^{(5)} + \mathbf{b}^{(5)}$$

$$\Longrightarrow \boldsymbol{Y} = \{[(\mathbf{L}^{(1)}\mathbf{W}^{(2)} + \mathbf{b}^{(2)})\mathbf{W}^{(3)} + \mathbf{b}^{(3)}]\mathbf{W}^{(4)} + \mathbf{b}^{(4)}\}\mathbf{W}^{(5)} + \mathbf{b}^{(5)}$$

$$\Longrightarrow \boldsymbol{Y} = \mathbf{L}^{(1)}\mathbf{W} + \mathbf{b}$$

$$\Longrightarrow \boldsymbol{Y} = \mathbf{X}\mathbf{W} + \mathbf{b},$$

where the total weight of the output is:

$$\mathbf{W} = \mathbf{W}^{(1)}\,\mathbf{W}^{(2)}\mathbf{W}^{(3)}\,\mathbf{W}^{(4)}\mathbf{W}^{(5)}\,.$$

And the total bias of the output is:

$$\mathbf{b} = \mathbf{b}^{(2)}\,\mathbf{W}^{(3)}\mathbf{W}^{(5)} + \mathbf{b}^{(3)}\mathbf{W}^{(4)}\mathbf{W}^{(5)} + \mathbf{b}^{(4)}\mathbf{W}^{(5)} + \mathbf{b}^{(5)}\,.$$

The relationship between the layers was linear in the previous example. In general, a key ingredient is needed in the deep learning model structure, a nonlinear function that is called an activation function $\mathcal{A}^{(\ell i)}, \ell i \in \{1, \ldots, \ell i\},$ which link each two sequential layers as follows,

$$\mathbf{L}^{(1)} = \mathcal{A}^{(1)}\boldsymbol{X} = \boldsymbol{X},$$

$$\mathbf{L}^{(2)} = \mathcal{A}^{(2)}\,(\,\mathbf{L}^{(1)}\mathbf{W}^{(2)} + \mathbf{b}^{(2)}\,),$$

$$\mathbf{L}^{(3)} = \mathcal{A}^{(3)}\,(\,\mathbf{L}^{(2)}\mathbf{W}^{(3)} + \mathbf{b}^{(3)}),$$

$$\mathbf{L}^{(4)} = \mathcal{A}^{(4)}\,(\,\mathbf{L}^{(3)}\mathbf{W}^{(4)} + \mathbf{b}^{(4)}),$$

$$\boldsymbol{Y} = \mathcal{A}^{(5)}\,(\,\mathbf{L}^{(4)}\mathbf{W}^{(5)} + \mathbf{b}^{(5)}).$$

Super Learner

Super Learner is considered an ensemble method. Ensemble learning methods are a combination of multiple-based learners to train them and make predictions using specific procedures (Ju et al., 2018). Researchers in different fields have demonstrated increased interest in the ensemble's methods due to their high performance in the prediction of empirical data. Such as applying the ensemble method in an online learning study (Benkeser et al., 2018), mortality prediction study (Chambaz et al., 2016), and precision medicine study (Wyss et al., 2018).

For instance, boosting, bagging, and stacking are examples of ensemble learning techniques. The boosting ensemble method takes care of the weak learner and boosts its performance (Freund & Schapire, 1996).

Conversely, bagging ensemble methods take care of the strongest algorithm to minimize its variance by applying the bootstrap aggregation (Breiman, 1996). Stacking is the linear combination of all learners (Wolpert, 1992).

Van der Laan et al. (2007) has extended the work of stacking from Wolpert (1992) to introduce what is called a super learner, which implements cross-validation and minimizes the validation risk to produce an optimal prediction based on the collection of an ensemble of learners which also has superior performance than to those learners individually.

It is an ensemble that estimates the performance of multiple algorithms through the cross-validation method, which has a result that is as good as the best-performing algorithm in the combination.

Super learner generates weights for each learner in the ensemble that is an optimal average based on their performance (Van Der Laan & Dudoit 2003; Van Der Laan et al., 2007).

Accepted the changes Accepted the changes have summarized the super learner algorithm in the following steps,

1. Split data into k blocks.
2. Fit all *M* methods on blocks, leaving out one block.
3. On each block, calculate for each method the mean squared error (*MSE*).
4. Repeat $(k-1)$ times in steps 2 and 3.
5. leave out one block $j = 2,3, \dots, k$ for each repetition.
6. Choose the method with the lowest *MSE* through the blocks.

According to Van der Laan et al. (2007), if each learner $L_k(n)$ (k= 1,..., K(n)) from a collection of learners $\grave{\Psi}_k$ is considered as an algorithm on the empirical distributions, then, a function could be defined mapping the empirical probability distributions P_n to a function of covariates $\grave{\Psi}_k(P_n)$,

$$L_k : \ P_n \rightarrow \grave{\Psi}_k(P_n),$$

where

Ψ is the parameter space,

then, the super learner is defined:

$$\grave{\Psi}(P_n) \equiv \grave{\Psi}_{\grave{K}(P_n)}(P_n)$$

where

$\grave{K}(P_n)$ is the selector that selects the optimal learner to minimize the cross-validation risk, thus,

$$\grave{K}(P_n) \equiv \arg\min_k E_{B_n} \sum_{i, B_n(i)=1} \left(Y_i - \grave{\Psi}_k\left(P^0_{n,B_n}\right)(X_i)\right)^2,$$

where

P^0_{n,B_n} is the empirical probability distribution of the validation set,

P^1_{n,B_n} is the empirical probability distribution of the training set,

$B_n \in \{0,1\}^n$ is a random binary vector to define the split of validation and training learning. $\{i: B_n(i) = 0\}$, and $\{i: B_n(i) = 1\}$, are the validation and training samples.

Van der Laan et al. (2007) define the following theorem to demonstrate that the super learner performs as best as the oracle selector up to the second order. So, the super learner is counted as the optimal learner under the conditions: $L_k(n)$ learners is polynomial in the sample size(n) and under the following assumptions.

Assumption 5

The loss function should be uniformly bounded. $\exists\ S_1 < \infty$, so that

$$\sup_{\psi \in \Psi} \sup_{O} |L(O, \psi) - L(O, \psi_0)| \leq S_1,$$

where

$L(O, \psi) = (Y - \psi(X))^2$ is the loss function,

$\Psi(P_0) = \psi_0$ is the parameter.

Assumption 6

The variance of the ψ_0- centered loss function $L(O, \psi) - L(O, \psi_0)$ can be bounded by its expectation uniformly in ψ.

Theorem 9

Under assumptions 5 and assumption 6, let p be the proportion of observations in the validation sample, specify $\{\grave{\psi}_k = \grave{\Psi}_k(P_n), k = 1, \ldots, K(n)\}$ as the set of $K(n)$ estimators, where the true parameter is defined as follows,

$$\psi_0 = \arg\ \min_{\psi \in \Psi} \int L(o, \psi) dP_0(o)$$

outline the difference of risk between parameter ψ_0 and the candidate estimator ψ as follows,

$$d_0(\psi, \psi_0) \equiv E_{P_0}\{L(O, \psi) - L(O, \psi_0)\},$$

thus, for any λ, the finite sample inequality is:

$$Ed_0\left(\grave{\Psi}_{\grave{K}(P_n)}\left(P^0_{n,B_n}\right), \psi_0\right) \leq (1 + 2\lambda)Ed_0\left(\grave{\Psi}_{\grave{K}(P_n)}\left(P^0_{n,B_n}\right), \psi_0\right) + 2C(\lambda)\frac{1+\log\left(K(n)\right)}{np},$$

where

Ψ is a parameter space,

$$\mathrm{P}\left[\grave{\Psi}_k(P_n) \in \Psi\right] = 1,$$

$\grave{K}(P_n) \equiv \arg\min_k E_{B_n} \int L\left(o, \grave{\Psi}_k\left(P^0_{n,B_n}\right)\right) dP_0(o)$, is the comparable oracle selector, and

$\grave{K}(P_n) \equiv \arg\min_k E_{B_n} \int L\left(o, \grave{\Psi}_k\left(P^0_{n,B_n}\right)\right) dP^1_{n,B_n}(o)$, is the cross-validation selector.

A Hybrid Method of Super Learner and Deep Learning with Support Points

The study proposes a super learner (SL) ensemble that is an ensemble model which is a hybrid of the deep neural network, the super learner ensemble for double debiased machine learning, and Support Points Splitting.

This section started to describe the deep learning paradigm and how its machine learning algorithm performs. Secondly, the super learner ensemble has been defined and its algorithm. Now, the new hybrid ensemble is introduced, a super learner ensemble model which is a hybrid of the deep neural network learning and the super learner ensemble. This method is used as machine learning to estimate the causal target parameter in the double debiased machine learning framework.

Double Machine Learning Inference in the Partially Linear Regression Model

Before stating the inference results of the partially linear regression (PLR) model, the assumptions required to oblige the regularity conditions to be met are formulated as follows,

Assumption 7

Regularity Conditions: Under the partial Linear Regression model, given $\mathcal{P}$ is the collection of probability laws P for the $Z = (Y, D, X)$, suppose a random fold I $\subset [N] = \{1, \ldots, N\}$ of size $n = N/K$, then the nuisance parameter $\grave{\eta}_0 = \grave{\eta}_0((Z_i)_{i \in I^c})$ satisfies,

$$\mathrm{P}\left[\forall\, n \geq 1 \,/\, \|\grave{\eta}_0 - \eta_0\|_{P,q} \leq C \wedge \|\grave{\eta}_0 - \eta_0\|_{P,2} \leq \delta_N\right] \;\geq\; 1 - \Delta_N\,.$$

For the case, the score function is,

$$\psi(Z; \beta, \eta) := \{Y - D\beta - g(X)\}(D - m(X)),\ \eta = (g, m),$$

then the nuisance will obey the following equation,

$$\grave{\eta}_0 = (\grave{g}_0, \grave{m}_0), \|\grave{m}_0 - m_0\|_{P,2} \times \| \grave{g}_0 - g_0 \|_{P,2} \leq \delta_N N^{-1/2}.$$

And for the case of the score function is defined by,

$$\psi(Z; \beta, \eta) := \{Y - \ell(X) - \beta(D - m(X))\}(D - m(X)),\ \eta = (\ell, m),$$

the nuisance will obey the following equation,

$$\grave{\eta}_0 = \left(\grave{\ell}_0, \grave{m}_0\right), \|\grave{m}_0 - m_0\|_{P,2} \times \left(\|\grave{m}_0 - m_0\|_{P,2} + \|\grave{\ell}_0 - \ell_0\|_{P,2}\right) \leq \delta_N N^{-1/2}.$$

And the disturbance terms $U \text{ and } V$, the response variable Y, the covariate X, and the target parameter D satisfies the following,

$$\|E_P[U^2 \mid X]\|_{P,\infty} \leq C,$$

$$\|E_P[V^2 \mid X]\|_{P,\infty} \leq C\,,$$

$$\| Y \|_{P,q} + \| D \|_{P,q} \leq C,$$

and

$$\| UV \|_{P,2} \geq c^2,$$

$$E_P[V^2] \geq \text{c}.$$

Theorem 10

Under assumption 7. The estimator $\grave{\beta}_0$ (either DML1 or DML2) will satisfy,

$$\sigma^{-1}\sqrt{N}\left(\grave{\beta}_0 - \beta_0\right) \xrightarrow{d} N(0,1),$$

which is derived from the following score function,

$$\psi(Z;\beta,\eta) := \{Y - T\beta - g(X)\}\big(T - m(X)\big),\ \eta = (g,m).$$

Or, from

$$\psi(Z;\beta,\eta) := \{Y - \ell(X) - \beta(T - m(X))\}(T - m(X)),\ \eta = (\ell,m),$$

where σ^2 is defined by,

$$\sigma^2 = (E_P[V^2])^{-1}E_P[V^2U^2](E_P[V^2])^{-1}.$$

And the Confidence Region (CR) are valid uniformly asymptotically,

$$\lim_{N\to\infty} \sup_{P\in\mathcal{P}} \left|Pr_P\big(\beta_0 \in \left[\grave{\beta}_0 \pm \Phi^{-1}(1-\alpha/2)\grave{\sigma}/\sqrt{N}\right]\big) - (1-\alpha)\right| = 0.$$

Simulation Scheme

Based on Chernozhukov et al. (2018), the sample size and the number of covariates that have been applied in their study were of *N* = 500, 1000, with numbers of covariates chosen as of *p* = 20. This simulation extends this scheme to more cases, such as *N* = 100 (relatively low sample size), and *p* = (20, 50, 80, 100), *p* = (200, 300, 500), and *p* = (1000, 2000, 3000) for larger covariates size. Under these cases, two scenarios are introduced. Consider the true value of the average treatment effect is to $\beta_0 = 0.5$, under the partial linear regression model.

$$Y = T\beta_0 + g_0(\mathbf{X}) + U, \mathbb{E}[\, U \mid \mathbf{X}, D\,] = 0,$$

$$T = m_0(\mathbf{X}) + V, \mathbb{E}[\, V \mid \mathbf{X}\,] = 0.$$

And the nuisance parameters are (Bach et al., 2021)

$$g_0(x_i) = \frac{\exp(x_i)}{1+\exp(x_i)} + \frac{1}{4}x_i,$$

$$m_0(x_i) = x_i + \frac{1}{4}\frac{\exp(x_i)}{1+\exp(x_i)} \quad .$$

Scenario 1

Consider the simulation studies introduced in Farbmacher et al. (2020), Bach et al. (2021), and Chernozhukov et al. (2018)

$$x_i \sim \mathcal{N}(0, \Sigma),\ \Sigma_{kj} = 0.5^{|j-k|},$$

$$\begin{pmatrix} U \\ V \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}\right).$$

Scenario 2

The following scenario is from Chernozhukov et al. (2018) and Bach et al. (2021)

The error terms are,

$$\zeta_i \sim \mathcal{N}(0,1),$$

$$v_i \sim \mathcal{N}(0,1),$$

with the covariates

$$x_i \sim \mathcal{N}(0, \Sigma),\ \Sigma_{kj} = 0.7^{|j-k|}.$$

The following is Table 1 that summarizes the planned simulation scheme with the scenarios and the cases. It shows 54 cases for each of the three Research Questions which will be a total of 162 simulation cases.

Table 1

Simulation Scheme Plan

Levels of High Dimensional Data	Low-High-Dimensional (LHD)									Moderate-High-Dimensional (MHD)										Big-High-Dimensional (BHD)								
Number of Covariates (*p)*	20			50			80			100			200			500			1000			2000			5000		
Sample Size (*N*)	100	500	1000	100	500	1000	100	500	1000	100	500	1000	100	500	1000	100	500	1000	100	500	1000	100	500	1000	100	500	1000
Scenario 1	Case1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10	Case 11	Case 12	Case 13	Case 14	Case 15	Case 16	Case 17	Case 18	Case 19	Case 20	Case 21	Case 22	Case 23	Case 24	Case 25	Case 26	Case 27	
Scenario 2	Case 28	Case 29	Case 30	Case 31	Case 32	Case 33	Case 34	Case 35	Case 36	Case 37	Case 38	Case 39	Case 40	Case 41	Case 42	Case 43	Case 44	Case 45	Case 46	Case 47	Case 48	Case 49	Case 50	Case 51	Case 52	Case 53	Case 54	

How to Answer the Research Questions

First, to answer the Research Question 1a and Research Question 2a, a simulation study is conducted with detailed scheme cases and scenarios that have been shown previously. Additionally, a comparative study is run between this study methods of each research question to the methods applied in the work of Chernozhukov et al. (2018).

Second, for the Research Question 1b and Research Question 2b, the benchmark study is conducted between the study application of the DML with SVM and support points sample splitting (SPSS), and the application of DML using DL and support points sample splitting compared to the Chernozhukov et al. (2018) methods, where they have used random splitting and the machine learning techniques, random forest, boosting, and neural network.

Third, for Research Question 3, the hybrid method of double machine learning (DML), super learner (SL), and deep learning (DL) deployed with support points sample splitting (SPSS) to estimate the average treatment effect is compared with the Chernozhukov et al. (2018) ensemble.

Finally, as a benchmark, the socio-economics real-world data, the 401(k) eligibility, and selection that has been applied in the paper by Chernozhukov et al. (2018) are used to compare this research methods.

This example has been intentionally chosen, as it touches the economic and social situations of the US participants, and this simulation wants to contribute to this study to detect the causal effect of 401(k) eligibility and selection. Implementing the causal inference from the observational data with the three machine learning techniques will help the decision-makers take informed decision making.

CHAPTER IV

RESULTS

This chapter answers the research questions stated in Chapter I. It presents the results of the simulation study and the real data analysis described in Chapter III. To capture the performance of the causal inference in the double machine framework for the semiparametric approach, the two scenarios described in the simulation scheme are implemented to compare the three models of the partial linear regression. The first scenario is having a data generation with uncorrelated covariance, while the second scenario allows a correlated covariance. For each scenario, three levels of the high dimensional data setting are adopted: low-high-dimensional, moderate-high-dimensional, and big-high-dimensional. The performance of the models is applied to the real data to explore and compare the results. For each research question, there are two parts, the first addresses the simulation performance of the models, and the second undertakes the application of the three models to the empirical example of real data. I start with the exploration of the simulation results of each question and conclude with the second part of the real data analysis.

In this simulation, there are 3 different sample sizes, 100, 500, and 1000. For each sample size, there will be 9 covariates sizes categorized as follows, 20, 50, 80, 100, 200, 500, 1000, 2000, 5000. And two scenarios for correlated and uncorrelated errors. This is 54 simulations for each of the 3 research questions, which is making it 162 simulations.

The covariates sizes are categorized into three levels to assist in the performance comparison between the cases of the simulations, low-high-dimensional (LHD) for p = (20, 50,

80), moderate-high-dimensional (MHD) for p = (100, 200, 500), and big-high-dimensional (BHD) for p = (1000, 2000, 5000).

Those simulations are being operated using both the personal computers and the high-performance computing cluster (HPC), which has assisted to make an informative performance comparison between the two computing paradigms and serves as a reference for future replication. The personal computers were operated with different cores ranging from 4-20 cores, where the high-performance computing cluster (HPC) was run through both standard and parallelization computing.

The identification model considered in this research framework is based on partial linear regression in the semiparametric approach. Noting that to infer the low dimensional causal treatment effect in the three models of this study, the semiparametric nuisance function g(x), m(x) is defined as high dimensional. The identification model is prescribed in two equations. The first equation accounts for the variation of the causal treatment effect T in the presence of the nuisance parameter $g_0(\mathbf{X})$. Where the propensity equation regresses the variation of the covariates on the treatment effect *T:*

$$Y = T\beta_0 + g_0(\mathbf{X}) + U, \mathbb{E}[\, U \mid \mathbf{X}, D\,] = 0,$$

$$T = m_0(\mathbf{X}) + V, \mathbb{E}[\, V \mid \mathbf{X}\,] = 0.$$

$\forall$ i $\in \{1, \dots, N\}$, the nuisance parameters are,

$$g_0(x_i) = \frac{\exp(x_i)}{1+\exp(x_i)} + \frac{1}{4}x_i,$$

$$m_0(x_i) = x_i + \frac{1}{4}\frac{\exp(x_i)}{1+\exp(x_i)} \quad .$$

Simulation Study Scenarios

Each Research Questions part (a) is answered under a simulation study for two scenarios settings, they are defined as follows:

Scenario 1

The covariate is:

$$x_i \sim \mathcal{N}(0, \Sigma),$$

$$\text{Toeplitz: } \Sigma_{kj} = 0.7^{|j-k|},$$

and the error are:

$$\mu_i \sim \mathcal{N}(0,1),$$

$$v_i \sim \mathcal{N}(0,1).$$

Scenario 2

The covariate is:

$$x_i \sim \mathcal{N}(0, \Sigma),$$

$$\text{Toeplitz: } \Sigma_{kj} = 0.5^{|j-k|},$$

and the error are:

$$\binom{U}{V} \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}\right).$$

Organization of the Simulation Tables and Graphs

The simulation results are organized in tables and graphs that show the different sample sizes, different covariates sizes, and under the two scenarios (correlated and uncorrelated errors). The first 6 tables are for answering Research Question 1, the support vector machine (SVM) under double machine learning (DML) with support point sample splitting (SPSS) model. The next 6 tables are for answering Research Question 2, the deep learning (DL) under double machine learning (DML) with support point sample splitting (SPSS) model. The last 6 tables are

for answering Research Question 3, the hybrid of super learner and deep learning (SDL) under double machine learning (DML) with support point sample splitting (SPSS) model. A summary tables shows a comparison of the best *MSE*, and best time efficiency for the three methods, SVM, DL, and SDL under DML framework and (SPSS) cross validation method.

All those tables and graphs will consider the three covariates' levels, low-high-dimensional data (LHD), moderate-high-dimensional data (MHD), and big-high-dimensional (BHD), and will distinguish each level simulation results separately both on tables and graphs.

Simulation Results of Research Question 1

The Research Question 1a mentioned in Chapter I and elaborated more in Chapter III is as follows:

> Q1a How does DML using support vector machine (SVM) and support points sample splitting (SPSS) perform in the simulation.

To compare the performance of the simulations results when varying the nuisance parameters' high dimensionality, three levels are considered, low-high-dimensional data when p = (20, 50, 80), moderate-high-dimensional data when p = (100, 200, 500) for, and big-high-dimensional data when p = *(*1000, 2000, 5000). The sample size for each simulation is N = (100, 500, 1000). The simulation results for different simulated data under Scenarios 1 and 2 are presented in the order mentioned above for this Research Question 1.

Results of Research Question 1 Simulations for Low-High-Dimensional Data

The results of the simulation study for low-high-dimensional data when p = (20, 50, 80) are displayed for Scenario 1, under the Research Question 1 concerning support vector machine model (SVM; see Table 2). The computational time for the low-high-dimensional data for

Scenario 1 was increased accordingly to the increase of the sample size *N* and the increase of the data dimension size *p*.

Table 2

Simulation Results of Research Question 1 for Scenario 1 with Low-High-Dimensional Data, When p = (20, 50, 80)

Scenario 1	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.0150	0.0882	0.0089	0.0080	4.8092
p = 20	500	0.0158	0.0382	0.0018	0.0017	18.7706
	1000	-0.0004	0.0294	0.0009	0.0009	65.6771
	100	0.2118	0.0657	0.0222	0.0492	5.5384
p = 50	500	0.1901	0.0360	0.0087	0.0374	41.2833
	1000	0.1283	0.0256	0.0041	0.0171	147.055
	100	0.2210	0.0721	0.0233	0.0541	6.2527
p = 80	500	0.1616	0.0349	0.0073	0.0273	66.0967
	1000	0.2008	0.0232	0.0064	0.0409	223.6176

Note. The number of replications is 500, *N* = sample sizes of (100, 500, 1000), Time = the running time of computing. These simulations were conducted using PC's.

Observe that in Table 2 for the case of *p* = 50, the bias was decreasing when the sample size of *N* = (100, 500, 1000) was increasing. But in the two cases of *p* = (20, 80) the bias was fluctuating when the sample sizes changed. The *SE* and *SE*-adjusted values were both following the same pattern of decreasing in all cases of covariates size *p* = (20, 50, 80) when the sample sizes were increasing *N* = (100, 500, 1000). The *MSE* values were decreasing in the case

covariates size is $p = 50$ for the increase of $N = (100, 500, 1000)$ and fluctuating for $p = (20, 80)$. Also, it has reached the lowest value of 0.0009 for the $p = 20$ and $N = 1000$.

The results of the simulation study for low-high-dimensional data under Research Question 1 for support vector machine (SVM) model**,** when $p = (20, 50, 80)$, for different sample sizes $N = (100, 500, 1000)$ under Scenario 2 when the errors are correlated, are displayed in Table 3.

Table 3

Simulation Results of Research Question 1 for Scenario 2 with Low-High-Dimensional Data, When p = (20, 50, 80)

Scenario 2	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.2644	0.0876	0.0279	0.0776	0.1908
$p = 20$	500	0.2653	0.0398	0.0120	0.072	20.1448
	1000	0.3251	0.0276	0.0103	0.1064	67.1507
	100	0.2919	0.0706	0.030	0.0902	6.1776
$p = 50$	500	0.2914	0.0310	0.0131	0.0859	36.2619
	1000	0.2863	0.0222	0.0091	0.0825	137.4567
	100	0.3289	0.0594	0.0334	0.1117	12.8976
$p = 80$	500	0.2921	0.0284	0.0131	0.0861	103.7343
	1000	0.3166	0.0209	0.010	0.1006	520.7393

Note. The number of replications is 500, N = sample sizes of (100, 500, 1000), Time = the running time of computing. These simulations were conducted using PC's.

It shows that both Bias and *MSE* were decreasing in the case of $p = 50$ and fluctuated in $p = (20, 80)$ when the sample size $N = (100, 500, 1000)$ increased. The SE and *SE*-adjusted were

decreasing in all cases of p = (20, 50, 80) under the increase of the sample size N = (100, 500, 1000). The *MSE* reached the lowest value of 0.072 for the N = 500 and p = 20. Also, for Scenario 2 in the low-high-dimensional data settings, the computational time was increased accordingly to the sample size N and the increase of the dimension size p.

Observe from Scenario 1 and Scenario 2 that the best causal estimator was for *MSE* value of 0.0009 for p = 20 and N = 1000 in Scenario 1 when the errors are uncorrelated compared to Scenario 2 when the errors are correlated. The changing trend in both scenarios was almost identical only in the case of the *MSE* p = 80 and N = 100 in Scenario 1 witnessed a decrease of *MSE* as the sample sizes increased but in Scenario 2, they fluctuated.

Results of Research Question 1
Simulations for Moderate-
High-Dimensional Data

The results of the simulation study for Research Question 1 concerning the support vector machine (SVM) model, in moderate-high-dimensional (MHD) data where p = (100, 200, 500), for each sample size N = (100, 500, 1000), under Scenario 1, are displayed in Table 4.

It shows that in Scenario 1, both *SE* and *SE*-adjusted values were decreasing in the three different covariate size cases when the sample size N = (100, 500, 1000) was decreasing. But both Bias and *MSE* were fluctuating in terms of the sample size increase, only for the case of p = 200 there was a decrease of the *MSE*. The *MSE* reached the lowest value of 0.0192 for p = 200 and N = 500. The computing time in Scenario 1 was increasing in terms of the increase of the sample sizes under this case of moderate-high-dimensional data under Research Question 1. All the simulations of this scenario were conducted using high-performance computing instead of the personnel computer considering the high covariates dimensions and high sample size.

Table 4

Simulation Results of Research Question 1 for Scenario 1 with Moderate-High-Dimensional Data when p = (100, 200, 500)

Scenario 1	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.2049	0.0706	0.0217	0.0470	10.4183
$p = 100$	500	0.1495	0.0325	0.0068	0.0234	22.916
	1000	0.1785	0.0234	0.0057	0.0324	55.3211
	100	0.1818	0.0743	0.0196	0.0386	7.7703
$p = 200$	500	0.1343	0.0342	0.0062	0.01920	23.7825
	1000	0.1976	0.0204	0.0062	0.0394	86.0439
	100	0.2284	0.0729	0.0240	0.0575	13.8357
$p = 500$	500	0.1467	0.0309	0.0067	0.0224	42.6681
	1000	0.1673	0.0221	0.0053	0.0285	130.3802

Note. The number of replications is 500, *N* = sample sizes of (100, 500, 1000), Time = the running time of computing. These simulations were conducted using the high-performance computing (HPC).

The results of the simulation study for Research Question 1 concerning the support vector machine (SVM) model, in moderate-high-dimensional (MHD) data where $p = (100, 200, 500)$, for each sample size $N = (100, 500, 1000)$, under Scenario 2, are displayed in Table 5.

It shows that in Scenario 2, Bias, *SE*, and *SE*-adjusted were fluctuating in all the covariate sizes $p = (100, 200, 500)$ in terms of the increase of the sample size $N = (100, 500, 1000)$. The *MSE* fluctuated in $p = 100$ but decreased in both cases $p = 200$ and $p = 500$. The *MSE* reached the lowest value of 0.0511 for $p = 200$ and $N = 100$. The computing time in Scenario 1 was increasing in terms of the increase of the sample sizes. under this case of moderate-high-

dimensional data under Research Question 1. All the simulations of this scenario were conducted using high-performance computing instead of the personnel computer considering the high covariates dimensions and high sample size.

Table 5

Simulation Results of Research Question 1 for Scenario 2 with Moderate-High-Dimensional Data When p = (100, 200, 500)

Scenario 2	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.2881	0.0606	0.0294	0.0867	6.2908
$p = 100$	500	0.3148	0.0296	0.0141	0.1000	21.6552
	1000	0.2773	0.0223	0.0088	0.0774	68.7314
	100	0.2179	0.0601	0.0226	0.0511	6.9184
$p = 200$	500	0.2668	0.0317	0.0120	0.0722	34.0644
	1000	0.2961	0.0228	0.0094	0.0882	101.6353
	100	0.2794	0.0793	0.0290	0.0843	8.9469
$p = 500$	500	0.2861	0.0331	0.0128	0.0829	42.1077
	1000	0.2857	0.0214	0.0091	0.0821	129.4497

Note. The number of replications is 500, *N* = sample sizes of (100, 500, 1000), Time = the running time of computing. These simulations were conducted using high-performance computing (HPC).

Observe that the best causal estimator was for an *MSE* value of 0.0192 for $p = 200$ and $N = 500$ in Scenario 1 when the errors are uncorrelated compared to Scenario 2 when the errors are correlated. The changing trend in both scenarios was almost identical, only Scenario 1 produced a decreased *SE* and *SE*-adjusted compared to Scenario 2 where both fluctuated.

Results of Research Question 1 Simulations for Big-High-Dimensional Data

The results of the simulation study for Research Question 1 concerning the support vector machine (SVM) model, in big-high-dimensional (BHD) data where p = (1000, 2000, 5000), for each sample size N = (100, 500, 1000), under Scenario 1, are displayed in Table 6.

It shows Scenario 1 for the case of big-high-dimensional data under research question 1. Both *SE* and *SE*-adjusted values were decreasing in the three different covariate size cases when the sample size N = (100, 500, 1000) was decreasing. But both Bias and *MSE* were fluctuating when the sample size increased. The *MSE* reached the lowest value of 0.0126 for p = 1000 and N = 100. The computing time in Scenario 1 was increasing in terms of the increase of the sample sizes. All the simulations of this scenario were conducted using high-performance computing instead of the personnel computer considering the high covariates dimensions and high sample size.

Table 6

Simulation Results of Research Question 1 for Scenario 1 with Big-High-Dimensional Data When p = (1000, 2000, 5000)

Scenario 1	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.0965	0.0573	0.0112	0.0126	15.1813
p = 1000	500	0.1615	0.0318	0.0074	0.0271	82.3944
	1000	0.1246	0.0235	0.0040	0.0161	257.2423
	100	0.1405	0.0663	0.0155	0.0241	49.631
p = 2000	500	0.1800	0.0315	0.0082	0.0334	158.3181
	1000	0.1894	0.0229	0.0060	0.0364	568.4668
	100	0.1962	0.0705	0.0209	0.0435	140.748
p = 5000	500	0.1960	0.0331	0.0089	0.0395	505.7694
	1000	0.1764	0.0252	0.0056	0.0318	1579.582

Note. The number of replications is 500, *N* = sample sizes of (100, 500, 1000), Time = the running time of computing. The simulations were conducted using the high-performance computing (HPC).

The results of the simulation study for Research Question 1 concerning the support vector machine (SVM) model, in big-high-dimensional (BHD) data where p = (1000, 2000, 5000), for each sample size N = (100, 500, 1000), under Scenario 2, are displayed in Table 7.

Table 7

Simulation Results of Research Question 1 for Scenario 2 with Big-High-Dimensional Data When p = (1000, 2000, 5000)

Scenario 2	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
p = 1000	100	0.3047	0.0675	0.0312	0.0974	14.876
	500	0.2898	0.0281	0.0130	0.0848	151.3939
	1000	0.2932	0.0216	0.0092	0.0864	253.2519
p = 2000	100	0.1715	0.0519	0.0179	0.0321	29.2644
	500	0.3046	0.0314	0.0136	0.0938	160.0492
	1000	0.2978	0.0233	0.0094	0.0892	578.5523
p = 5000	100	0.2335	0.0701	0.0244	0.0594	143.0843
	500	0.3226	0.0324	0.0145	0.1051	511.9322
	1000	0.2895	0.0225	0.0092	0.0843	1575.188

Note. The number of replications is 500, *N* = sample sizes of (100, 500, 1000), Time = the running time of computing. The simulations were conducted using the high-performance computing (HPC).

It shows that in Scenario 2 both *SE* and *SE*-adjusted values were decreasing in the three different covariate size cases when the sample size was increasing *N* = (100, 500, 1000). But both Bias and *MSE* were fluctuating as the sample size increased. The *MSE* reached the lowest value of 0.0321 for *p* = 2000 and *N* = 100. The computing time in Scenario 2 was increasing in terms of the increase of the sample sizes. All the simulations of this scenario were conducted using high-performance computing instead of the personnel computer considering the high covariates dimensions and high sample size.

Observe that the best causal estimator was for *MSE* = of 0.0126 for p = 1000 and N = 1000 in Scenario 1 when the errors are uncorrelated compared to Scenario 2 when the errors are correlated. The change trend behavior in both scenarios was almost identical.

Simulation Result of Research Question 2

The Research Question 2a mentioned in Chapter I and elaborated more in Chapter III is as follows:

Q2a How does DML using deep learning (DL) and support points sample splitting (SPSS) perform in the simulation.

To compare the performance of the simulations results when varying the nuisance parameters' high dimensionality, three levels are considered, low-high-dimensional data when p = *(*20, 50, 80), moderate-high-dimensional data when p = (100, 200, 500) for, and big-high-dimensional data when p = (1000, 2000, 5000). The sample size for each simulation is N = *(*100, 500, 1000). The simulation results for different simulated data under Scenarios 1 and 2 are presented in the order mentioned above for this Research Question 2.

Results of Research Question 2 Simulations for Low-High-Dimensional Data

The results of the simulation study for Research Question 2 concerning the deep learning model (DL) model, in low-high-dimensional (LHD) data where *p = (20, 50, 80),* for each sample size N = (100, 500, 1000), under Scenario 1, are displayed in Table 8.

It shows the results under the deep learning (DL) model with support point sample splitting (SPSS) technique for double machine learning Scenario 1 *SE*, *SE*-adjusted, and *MSE* were decreasing in all the covariate sizes p = (100, 200, 500) in terms of the increase of the sample size N = *(100,* 500, 1000). The Bias decreased when p = 80 as the sample size changed from N = 50 to N = 80 but fluctuated when p = 20. The *MSE* reached the lowest value of 0.0293

for p = 20 and N = 1000. The computing time in Scenario 1 under this case of low-high-dimensional for Research Question 2 was increasing in terms of the increase of the sample sizes.

Table 8

Simulation Results of Research Question 2 for Scenario 1 with Low-High-Dimensional Data When p = (20, 50, 80)

Scenario 1	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.1769	0.0779	0.0193	0.0374	1.5285
p = 20	500	0.1730	0.0350	0.0079	0.0311	40.1626
	1000	0.1694	0.0247	0.0054	0.0293	280.8999
	100	0.1716	0.0774	0.0188	0.0354	2.3883
p = 50	500	0.1704	0.0352	0.0078	0.0303	37.8766
	1000	0.1722	0.0247	0.0055	0.0303	125.0593
	100	0.1795	0.0782	0.0196	0.0383	6.0845
p = 80	500	0.1755	0.0351	0.0080	0.0320	178.9186
	1000	0.1702	0.0247	0.0054	0.0296	207.3817

Note. The number of replications is 500, N = sample sizes of (100, 500, 1000), Time = the running time of computing. These simulations were conducted using PC's.

The results of the simulation study for Research Question 2 concerning the deep learning model (DL) model, in low-high-dimensional (LHD) data where p = (20, 50, 80), for each sample size N = (100, 500, 1000), under Scenario 2, are displayed in Table 9.

It shows that in Scenario 2, Bias fluctuated. *SE, SE*-adjusted, and *MSE* were decreasing in all covariate sizes p= (100, 200, 500) in terms of the increase of the sample size N= (100, 500, 1000), but the MSE fluctuated for the case of p=80. The MSE reached the lowest value of 0.0844

for $p = 20$, $N = 1000$, and $p = 80$ with $N = 500$. The computing time in Scenario 1 was increasing in terms of the increase of the sample sizes. All the simulations of this scenario were conducted using a personnel computer.

Table 9

Simulation Results of Research Question 2 for Scenario 2 with Low-High-Dimensional Data When p = (20, 50, 80)

Scenario 2	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.2879	0.0833	0.0300	0.0898	1.6569
$p = 20$	500	0.2923	0.0382	0.0132	0.0869	32.5974
	1000	0.2894	0.0268	0.0092	0.0844	122.6058
	100	0.2982	0.0836	0.0310	0.0959	5.8937
$p = 50$	500	0.2896	0.0382	0.0131	0.0853	356.0577
	1000	0.2905	0.0268	0.0092	0.0851	126.3175
	100	0.3002	0.0855	0.0311	0.0974	3.9705
$p = 80$	500	0.2881	0.0377	0.0130	0.0844	33.6272
	1000	0.2908	0.0268	0.0092	0.0853	199.5886

Note. The number of replications is 500, N = sample sizes of (100, 500, 1000), Time = the running time of computing. The simulations were conducted on PC's.

Observe that the best causal estimator from table 8 and table 9 under low-high-dimensional data estimator was for MSE value of 0.0293 for $p = 200$ and $N = 1000$ in Scenario 1 when the errors are uncorrelated compared to Scenario 2 when the errors are correlated. The changing trend in both scenarios was almost identical, only Scenario 1 produced a decreased SE and SE-adj compared to Scenario 2 where both fluctuated.

Results of Research Question 2
Simulations for Moderate-
High-Dimensional Data

The results of the simulation study for Research Question 2 concerning the deep learning model (DL) model, in moderate-high-dimensional (MHD) data where p = (100, 200, 500), for each sample size N = (100, 500, 1000), under Scenario 1, are displayed in Table 10.

Table 10

Simulation Results of Research Question 2 for Scenario 1 with Moderate-High-Dimensional Data When p = (100, 200, 500)

Scenario 1	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.1731	0.0775	0.0187	0.036	4.1952
p =100	500	0.173	0.0349	0.0079	0.0311	8.7394
	1000	0.1701	0.0248	0.0054	0.0296	24.4785
	100	0.1771	0.0783	0.0192	0.0375	3.1304
p =200	500	0.1692	0.0349	0.0077	0.0299	15.0348
	1000	0.1712	0.0247	0.0055	0.0299	43.3119
	100	0.1762	0.0791	0.0191	0.0373	6.8924
p =500	500	0.1719	0.0351	0.0078	0.0308	35.6161
	1000	0.1703	0.0248	0.0054	0.0296	109.1765

Note. The number of replications is 500, N = sample sizes of (100, 500, 1000), Time is the running time of computing. These simulations were conducted using the high-performance computing (HPC).

It shows that in Scenario 1 under Research Question 2 for big-high-dimensional data, Bias, *SE*, and *SE*-adjusted, and *MSE* decreased for all covariate sizes p = (100, 200, 500) in

terms of the increase of the sample size N = (100, 500, 1000). The *MSE* reached the lowest value of 0.0296 for p = 500 and N = 1000 and p =100 and N = 1000. The computing time in Scenario 2 was increasing in terms of the increase of the sample sizes. All the simulations of this scenario were conducted using high-performance computing instead of the personnel computer considering the high covariates dimensions and high sample size.

The results of the simulation study for Research Question 2 concerning the deep learning model (DL) model, in moderate-high-dimensional (MHD) data where p = (100, 200, 500), for each sample size N = (100, 500, 1000), under Scenario 2, are displayed in Table 11.

It shows that in Scenario 2 under Research Question 2 for big-high-dimensional data, Bias, *SE*, and *SE*-adjusted, and *MSE* decreased for all covariate sizes p = (100, 200, 500) in terms of the increase of the sample size N = (100, 500, 1000). The *MSE* reached the lowest value of 0.0839 for p = 500 and N = 1000. The computing time in Scenario 2 was increasing in terms of the increase of the sample sizes. All the simulations of this scenario were conducted using high-performance computing instead of the personnel computer considering the high covariates dimensions and high sample size.

Observe that the best causal estimator was for *MSE* value of 0.0296 for p = 500 and N = 1000 for p = 500 and N = 500 in Scenario 1 when the errors are uncorrelated compared to Scenario 2 when the errors are correlated. The changing trend in both scenarios was identical.

Table 11

Simulation Results of Research Question 2 for Scenario 2 with Moderate-High-Dimensional Data When p = (100, 200, 500)

Scenario 2	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.2936	0.0846	0.0305	0.0933	3.9972
p =100	500	0.2927	0.0380	0.0132	0.0871	8.691
	1000	0.2898	0.0268	0.0092	0.0847	27.2387
	100	0.2996	0.0845	0.0311	0.0969	3.1202
p =200	500	0.2908	0.0378	0.0131	0.0860	19.7233
	1000	0.2903	0.0269	0.0092	0.0850	43.7439
	100	0.3013	0.0845	0.0312	0.0979	12.4327
p =500	500	0.2928	0.0380	0.0132	0.0872	35.3807
	1000	0.2885	0.0269	0.0092	0.0839	109.6356

Note. The number of replications is 500, *N* = sample sizes of (100, 500, 1000), Time is the running time of computing. The simulations were conducted using the high-performance computing (HPC).

Results of Research Question 2 Simulations for Big-High-Dimensional Data

The results of the simulation study for Research Question 2 concerning the deep learning model (DL) model, in big-high-dimensional (BHD) data where p = (1000, 2000, 5000), for each sample size N = (100, 500, 1000), under Scenario 1, are displayed in Table 12.

Table 12

Simulation Results of Research Question 2 for Scenario 1 with Big-High-Dimensional Data When p = (1000, 2000, 5000)

Scenario 1	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.1867	0.0782	0.0204	0.041	10.7405
p = 1000	500	0.1763	0.0352	0.0080	0.0323	67.0095
	1000	0.1713	0.0248	0.0055	0.0300	221.0456
	100	0.1782	0.0783	0.0194	0.0379	23.6871
p = 2000	500	0.1747	0.0352	0.0080	0.0318	139.5206
	1000	0.1693	0.0247	0.0054	0.0293	500.3704
	100	0.1792	0.0779	0.0195	0.0382	94.3971
p = 5000	500	0.1731	0.0349	0.0079	0.0312	493.4778
	1000	0.1718	0.0248	0.0055	0.0301	1430.124

Note. The number of replications is 500, *N* = sample sizes of (100, 500, 1000), Time = the running time of computing. These simulations were conducted on the high-performance computing (HPC).

It shows that in Scenario 1 under Research Question 2 for big-high-dimensional data, Bias, *SE*, and *SE*-adjusted, and *MSE* decreased for all covariate sizes p = (100, 200, 500) in terms of the increase of the sample size N = (100, 500, 1000). The *MSE* reached the lowest value of 0.0293 4 for p = 2000 and N = 1000. The computing time in Scenario 1 was increasing in terms of the increase of the sample sizes. All the simulations of this scenario were conducted using high-performance computing instead of the personnel computer considering the high covariates dimensions and high sample size.

The results of the simulation study for Research Question 2 concerning the deep learning model (DL) model, in big-high-dimensional (BHD) data where p = (1000, 2000, 5000), for each sample size N = (100, 500, 1000), under Scenario 2, are displayed in Table 13.

Table 13

Simulation Results of Research Question 2 for Scenario 2 with Big-High-Dimensional Data When p = (1000, 2000, 5000)

Scenario 2	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.2987	0.0834	0.0311	0.0962	10.8812
p = 1000	500	0.2936	0.0381	0.0132	0.0877	64.2134
	1000	0.2904	0.0269	0.0092	0.0850	218.7591
	100	0.3074	0.0843	0.0319	0.1016	25.2808
p = 2000	500	0.2955	0.0381	0.0133	0.0888	140.8162
	1000	0.2894	0.0268	0.0092	0.0844	502.6168
	100	0.3023	0.0835	0.0313	0.0984	98.7331
p = 5000	500	0.2954	0.0381	0.0133	0.0887	500.0961
	1000	0.2887	0.0267	0.0092	0.0841	1477.899

Note. The number of replications is 500, N = sample sizes of (100, 500, 1000), Time = the running time of computing. The simulations were conducted using the high-performance computing (HPC).

It shows that in Scenario 2 under Research Question 2 for big-high-dimensional data, Bias, *SE*, and *SE*-adjusted, and *MSE* also decreased for all covariate sizes p = (100, 200, 500) in terms of the increase of the sample size N = (100, 500, 1000). The *MSE* reached the lowest value of 0.0841 for cases p = 5000 and N = 1000. The computing time in Scenario 1 was increasing in

terms of the increase of the sample sizes. All the simulations of this scenario were conducted using high-performance computing instead of the personnel computer considering the high covariates dimensions and high sample size.

Observe that the best causal estimator was for an *MSE* value of 0.0293 for $p = 2000$ and $N = 1000$ in Scenario 1 when the errors are uncorrelated compared to Scenario 2 when the errors are correlated. The changing trend in both scenarios was identical.

Simulation Result of Research Question 3

The Research Question 3a mentioned in Chapter I and elaborated more in Chapter III is as follows:

Q3a How does DML using a hybrid model with the super learner and deep learning with support points sample splitting (SPSS) perform in the simulation.

To compare the performance of the simulations results when varying the nuisance parameters' high dimensionality, three levels are considered, low-high-dimensional data when $p = (20, 50, 80)$, moderate-high-dimensional data when $p = (100, 200, 500)$, and in big-high-dimensional data when $p = (1000, 2000, 5000)$. The sample size for each simulation is $N = (100, 500, 1000)$. The simulation results for different simulated data under Scenarios 1 and 2 are presented in the order mentioned above for this Research Question 3.

Results of Research Question 3
Simulations for Low-High-Dimensional Data

The results of the simulation study for Research Question 3 concerning the hybrid model of super learner and deep learning (SDL), in low-high-dimensional (LHD) data where $p = (20, 50, 80)$, for each sample size $N = (100, 500, 1000)$, under Scenario 1, are displayed in Table 14.

It shows that in Scenario 1 of Research Question 3, Bias, and *SE*-adjusted were fluctuating in a couple of the covariate size $p = (20, 50, 80)$ as the sample size increases $N =$

(100, 500, 1000). The *MSE* and the *SE* decreased for all covariate sizes. The *MSE* reached the lowest value of 0.0008 for $p = 80$ and $N = 1000$. The computing time in Scenario 1 was increasing in terms of the increase of the sample sizes. under this case of low-high-dimensional data under Research Question 3.

Table 14

Simulation Results of Research Question 3 for Scenario 1 with Low-High-Dimensional Data When p = (20, 50, 80)

Scenario 1	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.029	0.0751	0.0097	0.0065	82.8915
$p = 20$	500	-0.0179	0.0338	0.0019	0.0015	117.4088
	1000	-0.0249	0.0240	0.0011	0.0012	234.6404
	100	0.0547	0.0749	0.0105	0.0086	78.4494
$p = 50$	500	-0.0094	0.0337	0.0019	0.0012	172.9981
	1000	-0.0189	0.0239	0.001	0.0009	453.1429
	100	0.0529	0.0759	0.0102	0.0086	85.5482
$p = 80$	500	-0.004	0.0338	0.0019	0.0012	1872.9109
	1000	-0.016	0.0239	0.0010	0.0008	677.2905

Note. The number of replications is 500, *N* = sample sizes of (100, 500, 1000), Time = the running time of computing. These simulations were conducted using PCs.

The results of the simulation study for Research Question 3 concerning the hybrid model of super learner and deep learning (SDL), in low-high-dimensional (LHD) data where *p = (20, 50, 80),* for each sample size *N* = (100, 500, 1000), under Scenario 2, are displayed in Table 15.

Table 15

Simulation Results of Research Question 3 for Scenario 2 with Low-High-Dimensional Data When p = (20, 50, 80)

Scenario 2	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.1739	0.0776	0.0191	0.0363	76.6939
$p = 20$	500	0.1259	0.0348	0.0058	0.0171	111.9249
	1000	0.1138	0.0246	0.0037	0.0136	229.4855
	100	0.1985	0.0786	0.0213	0.0456	78.835
$p = 50$	500	0.1299	0.0348	0.006	0.0181	173.805
	1000	0.1173	0.0245	0.0038	0.0144	454.5324
	100	0.2141	0.0797	0.0228	0.0522	89.4091
$p = 80$	500	0.1330	0.0347	0.0061	0.0189	306.3423
	1000	0.1195	0.0246	0.0039	0.0149	662.765

Note. The number of replications is 500, *N* = sample sizes of (100, 500, 1000), Time = the running time of computing. These simulations were conducted using PC's.

It shows that in Scenario 2 for Research Question 3 under the case of low-high-dimensional data, where $p = (20, 50, 80)$, Bias, *SE*, and *SE*-adjusted, *MSE* decreased in all the covariate size $p = (100, 200, 500)$ in terms of the increase of the sample size $N = (100, 500, 1000)$. The *MSE* reached the lowest value of 0.01361 for $p = 200$ and $N = 1000$.

Observe that the best causal estimator was for an *MSE* value of 0.0008 for $p = 80$ and $N = 1000$ in Scenario 1 when the errors are uncorrelated compared to Scenario 2 when the errors are correlated. The change trend behavior of *MSE* and SE was identical in both scenarios but different in both bias *SE*-adjusted. The computing time in both scenarios was increasing in terms of the increase of the sample sizes.

Results of Research Question 3
Simulations for Moderate-
High-Dimensional Data

The results of the simulation study for Research Question 3 concerning the hybrid model of super learner and deep learning (SDL), in moderate-high-dimensional (MHD) data where p = = (100, 200, 500), for each sample size N = (100, 500, 1000), under Scenario 1, are shown in Table 16.

Table 16

Simulation Results of Research Question 3 for Scenario 1 with Moderate-High-Dimensional Data When p = (100, 200, 500)

Scenario 1	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.0667	0.0743	0.0108	0.0100	93.6581
p = 20	500	-0.0053	0.0336	0.0018	0.0012	120.6646
	1000	-0.0152	0.0239	0.0010	0.0008	136.0735
	100	0.0758	0.0749	0.0113	0.0114	80.1428
p = 50	500	-0.0011	0.0335	0.0018	0.0011	130.3356
	1000	-0.0133	0.0238	0.0010	0.0007	244.8391
	100	0.0843	0.0761	0.0120	0.0129	85.8635
p = 80	500	0.0043	0.0334	0.0019	0.0011	175.2444
	1000	-0.0107	0.0238	0.0009	0.0007	241.7942

Note. The number of replications is 500, N = sample sizes of (100, 500, 1000), Time = the running time of computing. These simulations were conducted using the high-performance computing (HPC).

It shows that in Scenario 1 for Research Question 3 under the case of moderate-high-dimensional data, where p = (100, 200, 500), *SE*, and *SE*-adjusted, *MSE* decreased in all the covariate size p = (100, 200, 500) in terms of the increase of the sample size N = (100, 500, 1000). Only the Bias fluctuated. The *MSE* reached the lowest value of 0.0007 for p = 200 with N = 1000 and p =500 with N = 1000.

The results of the simulation study for Research Question 3 concerning the hybrid model of super learner and deep learning (SDL), in moderate-high-dimensional (MHD) data where p = *(100, 200, 500)*, for sample size N = (100, 500, 1000), in Scenario 2, are shown in Table 17.

It shows that in Scenario 2 for Research Question 3 under the case of moderate-high-dimensional data, where p = (100, 200, 500), Bias, *SE*, and *SE*-adjusted, *MSE* decreased in all the covariate size p = (100, 200, 500) in terms of the increase of the sample size N = (100, 500, 1000). The *MSE* reached the lowest value of 0.015 for p = 100 with N = 1000.

Observe that the best causal estimator for Research Question 3 deep learning (DL) model under the case of moderate-high-dimensional data was in Scenario 1 with a value of 0.0007 for p = 200 with N = 1000, and p =500 with N = 1000 when the errors are uncorrelated compared to Scenario 2 when the errors are correlated. The change trend behavior of *MSE* and *SE*. Standard error-adjusted (*SE*-adjusted) was identical in both scenarios but different in Bias. The computing time in both scenarios was increasing in terms of the increase of the sample sizes.

Table 17

Simulation Results of Research Question 3 for Scenario 2 with Moderate-High-Dimensional Data When p = (100, 200, 500)

Scenario 2	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.2234	0.0802	0.0238	0.0563	77.2179
p = 100	500	0.1374	0.0346	0.0063	0.0201	116.3234
	1000	0.1201	0.0246	0.0039	0.015	171.6015
	100	0.2369	0.0789	0.0251	0.0624	117.0373
p = 200	500	0.1390	0.0351	0.0064	0.0206	169.5755
	1000	0.1234	0.0244	0.004	0.0158	211.5309
	100	0.2485	0.0801	0.0262	0.0682	86.5252
p = 500	500	0.1418	0.0348	0.0065	0.0213	134.612
	1000	0.1264	0.0244	0.0041	0.0166	244.7366

Note. The number of replications is 500, *N* = sample sizes of (100, 500, 1000), Time = the running time of computing. These simulations were conducted using the high-performance computing (HPC).

Results of Research Question 3 Simulations for Big-High-Dimensional Data

The results of the simulation study for Research Question 3 concerning the hybrid model of super learner and deep learning (SDL), in big-high-dimensional (BHD) data where *p* = (1000, 2000, 5000), for each sample size *N* = (100, 500, 1000), in Scenario 1, are shown in Table 18.

It shows that in Scenario 1 for Research Question 3 under the case of big-high-dimensional data, where *p* = (1000, 2000, 5000), *SE*, *SE*-adjusted, and *MSE* decreased in all the covariate size *p* = (100, 200, 500) in terms of the increase of the sample size *N* = (100, 500,

1000), where the bias fluctuated. The *MSE* reached the lowest value of 0.0006 for p = 1000 with N = 1000, p =2000 with N = 1000, and p = 5000 with N = 1000.

Table 18

Simulation Results of Research Question 3 for Scenario 1 with Big-High-Dimensional Data When p = (1000, 2000, 5000)

Scenario 1	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.1088	0.0773	0.0135	0.0178	99.0878
p = 1000	500	0.0058	0.0333	0.0018	0.0011	188.4077
	1000	-0.007	0.0237	0.0009	0.0006	385.9213
	100	0.1109	0.0762	0.0136	0.0181	141.0441
p = 2000	500	0.0127	0.0333	0.0019	0.0013	304.3408
	1000	-0.0038	0.0237	0.0009	0.0006	704.1351
	100	0.1225	0.0772	0.0146	0.0210	402.0184
p = 5000	500	0.0135	0.0332	0.0018	0.0013	915.3463
	1000	0.0015	0.0237	0.0009	0.0006	1984.354

Note. The number of replications is 500, N = sample sizes of (100, 500, 1000), Time = the running time of computing. These simulations were conducted using the high-performance computing (HPC).

The results of the simulation study for Research Question 3 concerning the hybrid model of super learner and deep learning (SDL), in big-high-dimensional (BHD) data where p = (1000, 2000, 5000), for each sample size N = (100, 500, 1000), in Scenario 2, are shown in Table 19.

Table 19

Simulation Results of Research Question 3 for Scenario 2 with Big-High-Dimensional Data When p = (1000, 2000, 5000)

Scenario 2	*N*	Bias	*SE*	*SE*-adjusted	*MSE*	Time
	100	0.2509	0.0808	0.0263	0.0695	100.9268
p = 1000	500	0.1508	0.0344	0.0069	0.0239	211.6720
	1000	0.1304	0.0246	0.0042	0.0176	403.7650
	100	0.2597	0.0813	0.0273	0.0741	160.6140
p = 2000	500	0.1544	0.0343	0.0071	0.025	307.7292
	1000	0.1351	0.0244	0.0043	0.0188	710.9581
	100	0.2668	0.0823	0.0279	0.0780	453.629
p = 5000	500	0.1594	0.0345	0.0073	0.0266	897.9079
	1000	0.1394	0.0244	0.0045	0.0200	1970.1150

Note. The number of replications is 500, *N* = sample sizes of (100, 500, 1000), Time = the running time of computing. These simulations were conducted using the high-performance computing (HPC).

It shows that in Scenario 2, *SE* and *SE*-adjusted decreased in all the covariate size *p* = (100, 200, 500) in terms of the increase of the sample size *N* = (100, 500, 1000). The Bias and *MSE* fluctuated. The *MSE* reached the lowest value of 0.0176 for *p* = 1000 and *N* = 1000.

Observe that the best causal estimator was for *MSE* value of 0.0006 for *p* = 1000 with *N* = 1000, and *p* = 2000 with *N* = 1000, in Scenario 1 when the errors are uncorrelated compared to Scenario 2 when the errors are correlated. The changing trend in both scenarios was almost identical. The computing time in Scenario 1 was increasing in terms of the increase of the sample sizes.

Figure 8 displays the *MSE* in Research Question 1 under the support vector machine (SVM) model, for Scenario 1 and Scenario 2 in the case of low-high-dimensional data when p = (20, 50, 80).

Observe that the mean square error (*MSE*) was small for p is small but gets higher once p is larger for both scenarios.

Figure 8

Comparison of Mean Square Error in Low-High-Dimensional Data for Support Vector Machine Model

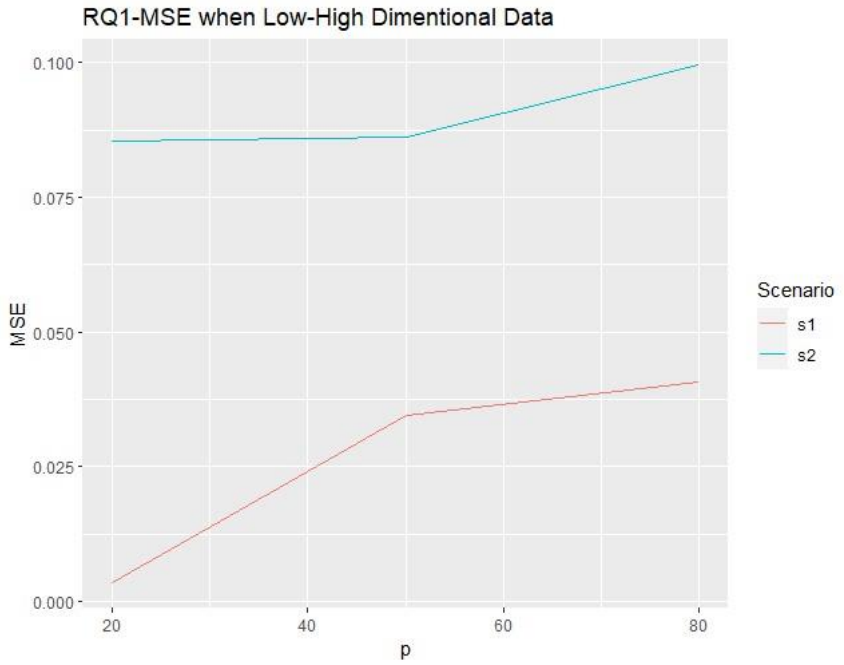


Figure 9 shows the comparison of *MSE* in Research Question 1 under the support vector machine (SVM) model, for Scenario 1 and Scenario 2 in the case of moderate-high-dimensional data when p = (100, 200, 500).

Observe that the mean square error (*MSE*) was fluctuating as the covariates size changes for both scenarios.

Figure 9

Comparison of Mean Square Error in Moderate-High-Dimensional Data for Support Vector Machine Model

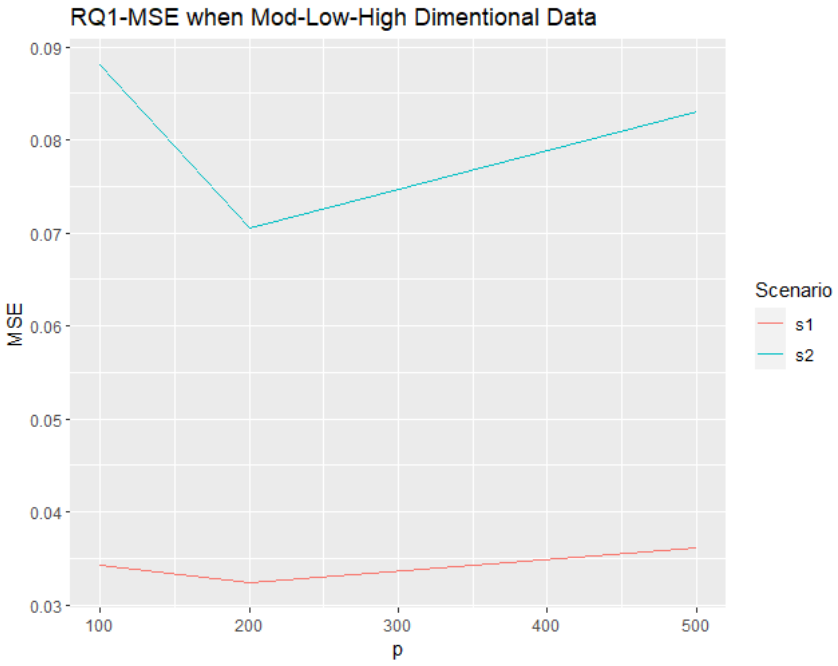
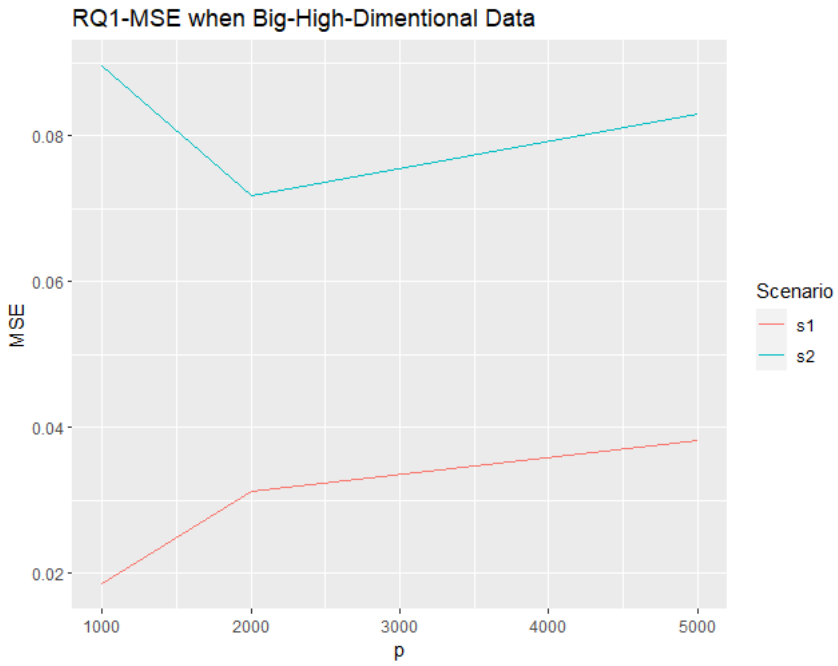


Figure 10 shows the comparison of the *MSE* in Research Question 1 under the support vector machine (SVM) model, for Scenario 1 and Scenario 2 in the case of big-high-dimensional data when p = (1000, 2000, 5000).

Observe that the mean square error (*MSE*) was fluctuating also as the covariates size changes for both scenarios.

Figure 10

Comparison of Mean Square Error in Big-High-Dimensional Data for Support Vector Machine Model



The following are graphs that show the mean square error in the model of deep learning machine (DL) under the three high dimensional data levels, low-high-dimensional, moderate-high-dimensional, and big-high-dimensional.

Figure 11 displays the comparison of the mean square error (*MSE*) in Research Question 2 under deep learning model (DL), for Scenario 1 and Scenario 2 in the case of low-high-dimensional data when p = (20, 50, 80).

Observe that the mean square error (*MSE*) was quite stable when the covariates size changes, for both scenarios.

Figure 11

Comparison of Mean Square Error in Low-High-Dimensional Data for Deep Learning Model

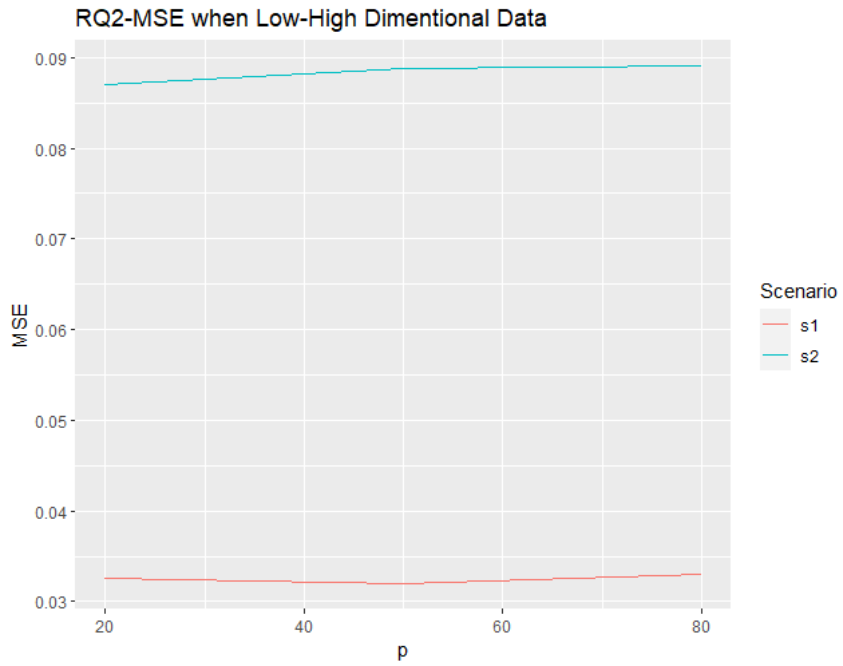


Figure 12 displays the comparison of mean square error (*MSE*) in Research Question 2 for Scenario 1 and Scenario 2, under deep learning model (DL), in the case of moderate-high-dimensional data when p = (100, 200, 500).

Observe that the mean square error (*MSE*) was quite stable when the covariates size changes, for both scenarios.

Figure 12

Comparison of Mean Square Error in Moderate-High-Dimensional Data for Deep Learning Model

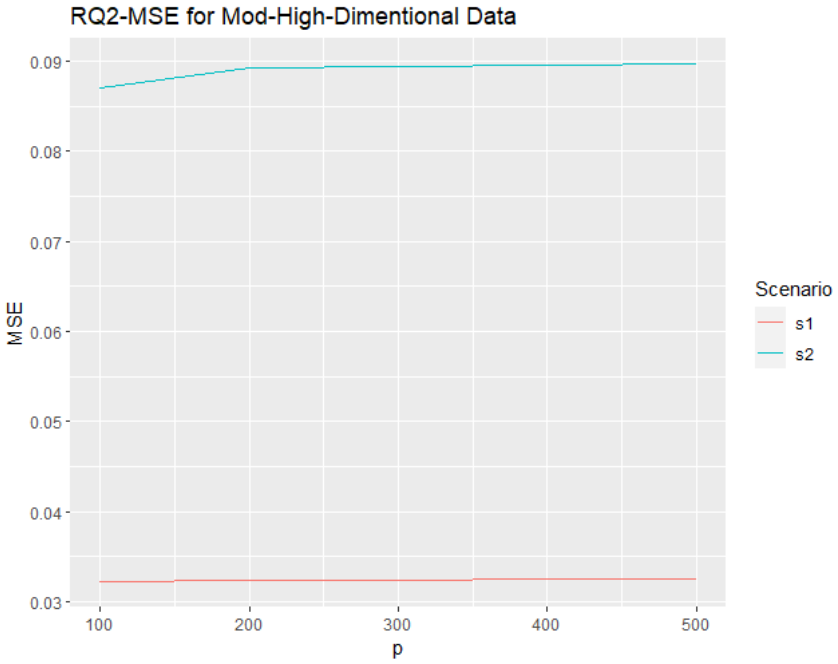
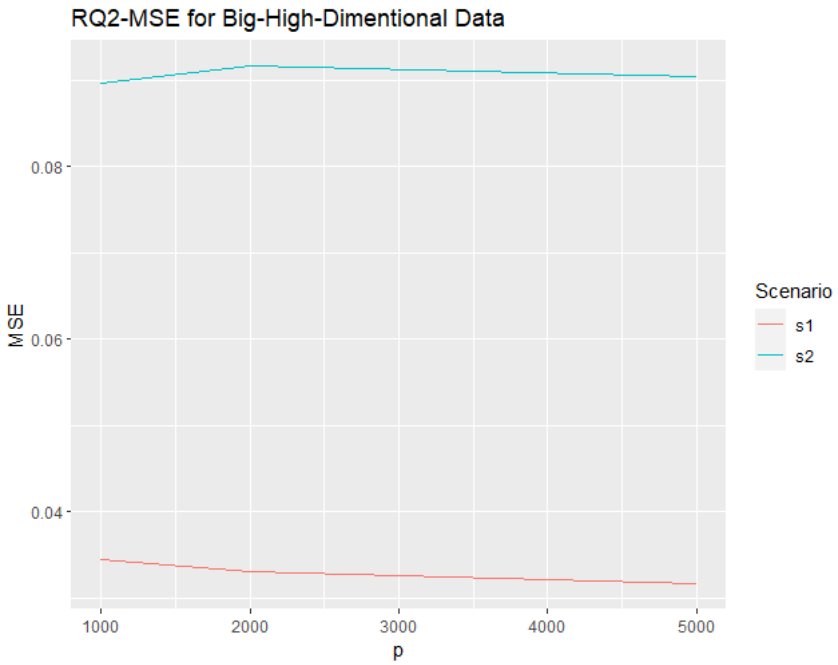


Figure 13 displays the comparison of the mean square error (*MSE*) in Research Question 2, under deep learning model (DL), for Scenario 1 and Scenario 2 in the case of big-high-dimensional data when p = (1000, 2000, 5000).

Observe that the mean square error (*MSE*) was quite stable when the covariates size changes, for both scenarios.

Figure 13

Comparison of Mean Square Error in Big-High-Dimensional Data for Deep Learning Model



The following are graphs that show the mean square error in the model of super deep learning (SDL), the hybrid model of super learner and deep learning, under the three high dimensional data levels, low-high-dimensional, moderate-high-dimensional, and big-high-dimensional.

Figure 14 shows the comparison of the *MSE* in Research Question 3, under the hybrid model of super learner and deep learning (SDL), for Scenario 1 and Scenario 2 in the case of low-high-dimensional data when p = (20, 50, 80).

Observe that the mean square error (*MSE*) was quite stable when the covariates size changes for Scenario 1 but was increasing when the covariate size increases in Scenario 2.

Figure 14

Comparison of Mean Square Error in Low-High-Dimensional Data for Super Deep Learning Model

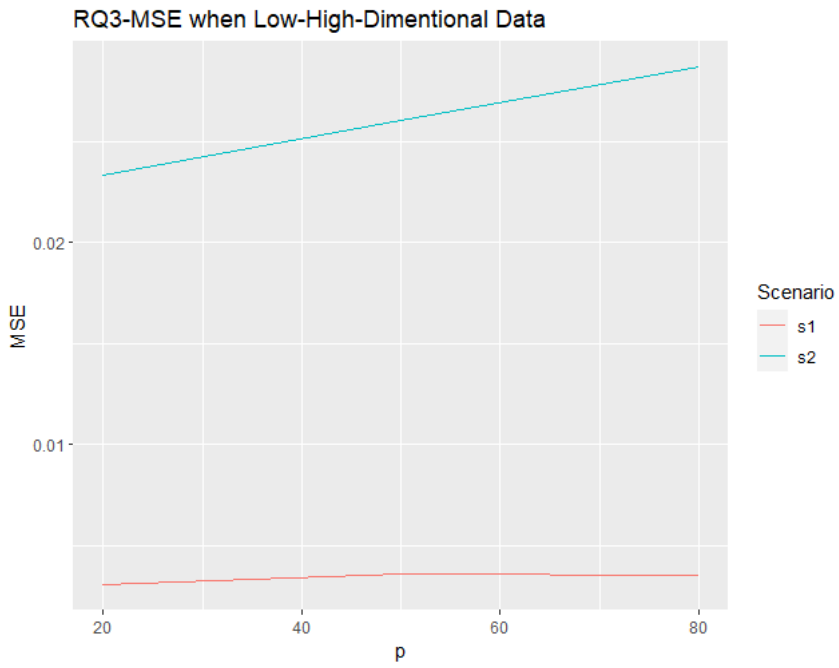


Figure 15 displays the comparison of *MSE* in Research Question 3, under the hybrid model of super learner and deep learning (SDL), for Scenario 1 and Scenario 2 in the case of moderate-high-dimensional data when p = (100, 200, 500).

Observe that the mean square error (*MSE*) was quite stable when the covariates size changes for Scenario 1 but was increasing when the covariate size increases in Scenario 2.

Figure 15

Comparison of Mean Square Error in Moderate-High-Dimensional Data for Super Deep Learning Model

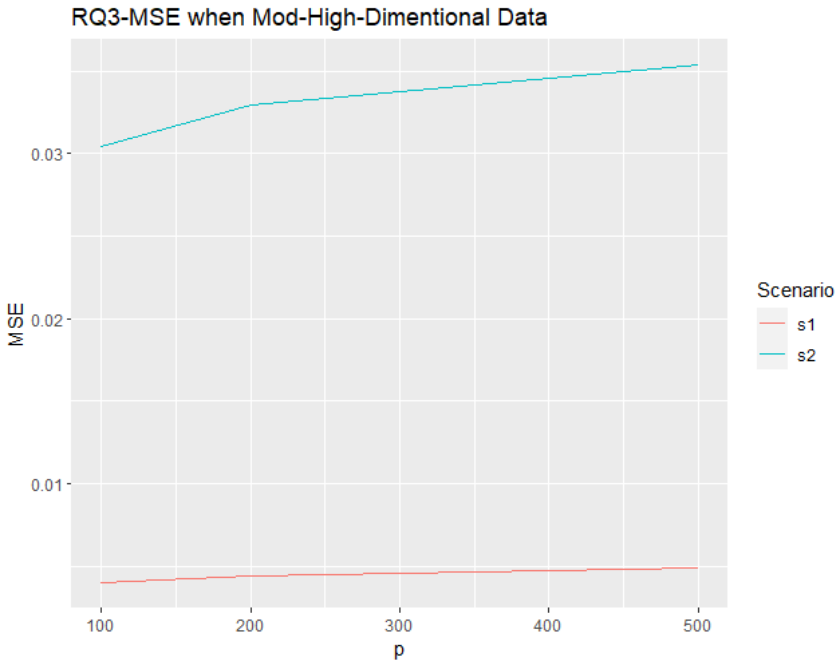
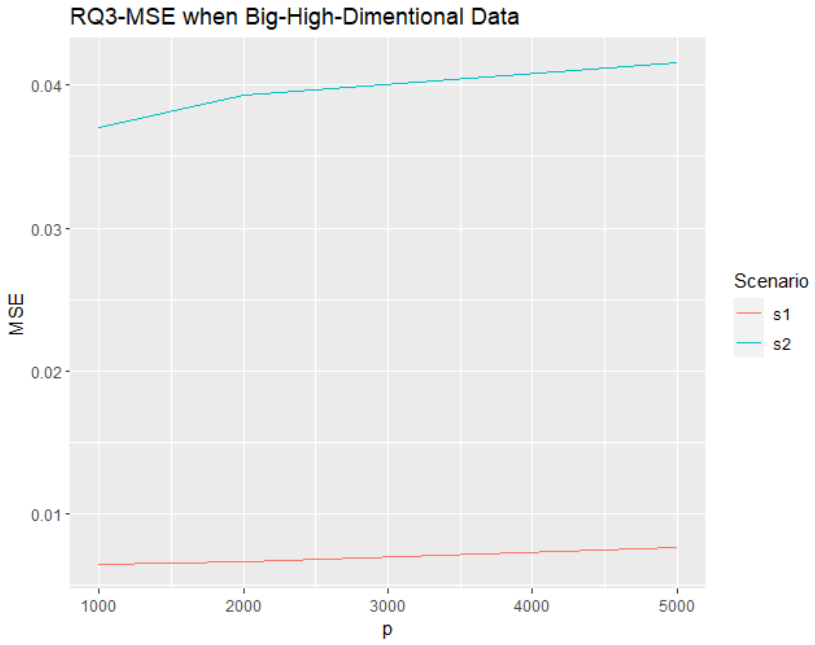


Figure 16 shows the comparison of the *MSE* in Research Question 3, under the hybrid model of super learner and deep learning (SDL), for Scenario 1 and Scenario 2 in the case of big-high-dimensional data when p = (1000, 2000, 5000).

Observe that the mean square error (*MSE*) was quite stable when the covariates size changes for Scenario 1 but was increasing when the covariate size increases in Scenario 2.

Figure 16

Comparison of Mean Square Error in Big-High-Dimensional Data for Super Deep Learning Model



The following are summaries of the comparisons of the mean square error values, and the simulations time, cross the three models, support vector machine (SVM), deep learning (DL), the hybrid of super learner and deep learning (SDL). Under the three high dimensional data levels, low-high-dimensional (LHD), moderate-high-dimensional (MHD), and big-high-dimensional (BHD).

Observe that Table 20 shows that the best *MSE* was under SDL method with *MSE* = 0.0006 in BHD data, *MSE* = 0.0007 under MHD data, and *MSE* = 0.0008 under LHD data, followed by SVM method with *MSE* = 0.009 for LHD.

Table 20

Mean Square Error Comparison for the Three Methods (Support Vector Machine, Deep Learning, and Super Deep Learning) Under the Three Data Levels for Scenario 1 and Scenario 2

	High Dimensional Data Levels	Scenarios	SVM	DL	SDL
MSE	Low-High-Dimensional (LHD)	S1	0.0009	0.0293	0.0008
		S2	0.072	0.0844	0.0136
	Moderate-High-Dimensional (MHD)	S1	0.0192	0.0296	0.0007
		S2	0.0511	0.0839	0.0150
	Big-High-Dimensional (BHD)	S1	0.0126	0.0293	0.0006
		S2	0.0321	0.0841	0.0176

Note. DL = deep learning model SDL = hybrid of super learner and deep learning model, SVM = support vector machine model.

Table 21 shows that the lowest total computational duration was under DL method in moderate-high-dimensional and big-high-dimensional data, and in Scenario 2 for low-dimensional data. In the case of low-high-dimensional data Scenario 1, the SVM has delivered a better time efficiency.

Table 21

Time of Computation Comparison for the Three Models (Support Vector Machine, Deep Learning, and Super Deep Learning) Under the Three Data Levels Sor Scenario 1 and Scenario 2

	High Dimensional Data Levels	Scenarios	SVM	DL	SDL
Time (Minutes)	Low-High-Dimensional (LHD)	S1	579.1006	880.3000	3775.281
		S2	904.7537	882.3153	2183.793
	Moderate-High-Dimensional (MHD)	S1	393.1361	250.5752	1308.616
		S2	419.7998	263.9633	1329.160
	Big-High-Dimensional (BHD)	S1	3342.152	2969.632	5025.568
		S2	3417.592	3039.296	5217.317

Note. DL = deep learning method, SDL = the hybrid of super learner and deep learning method, SVM is support vector machine method, Time = the running time for simulation.

The figures below show the comparison between the three models, Support Vector Machine (SVM), Deep Learning (DL), and Super Deep Learning (SDL), under the three data levels for Scenario 1 and Scenario 2 in terms of mean square error (MSE) and time of computation.

Figure 17 shows the values of the mean square error comparison under low-high-dimensional data (LHD), which is when the covariates sizes are p = (20, 50, 80), cross the three models, support vector machine (SVM), deep learning (DL), and the hybrid of super learner and deep learning (SDL). The lowest mean square error was in super deep leaning (SDL)model for both scenarios, compared to the other two models, support vector machine (SVM), and deep Learning (DL). However, the support vector machine has shown a low mean square error too in Scenario 1, when the error is uncorrelated.

Figure 17

Mean Square Error Comparison for the Three Models Under Low-High-Dimensional Data

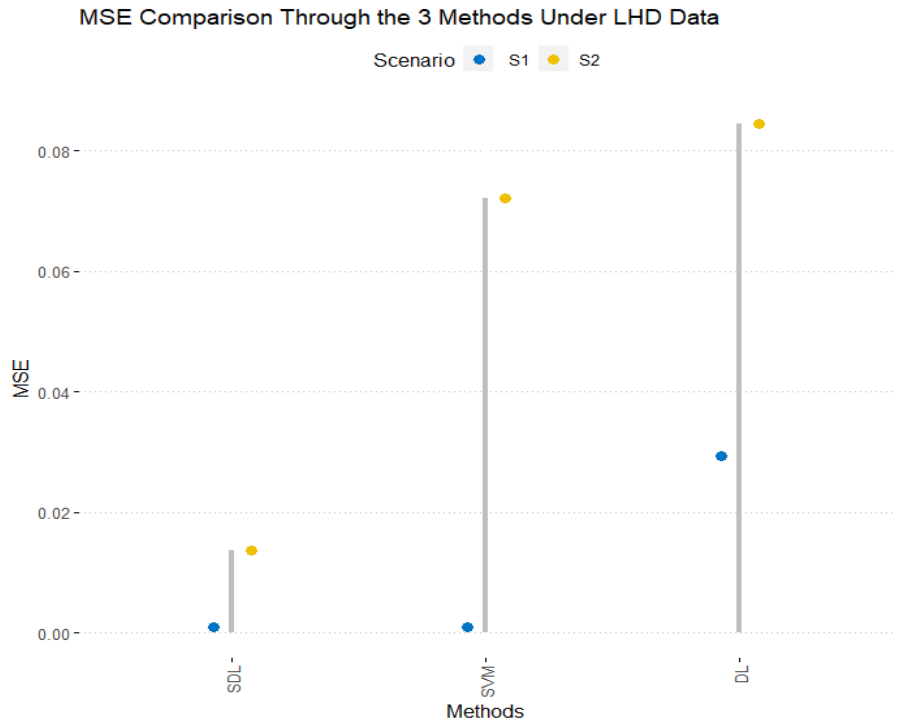


Figure 18 shows the values of the mean square error comparison under moderate-high-dimensional data (MHD), which is when the covariates sizes are p = (100, 200, 500), cross the three models, support vector machine (SVM), deep learning (DL), and the hybrid of super learner and deep learning (SDL). The lowest mean square error was in super deep leaning (SDL) model for both scenarios, compared to the other two models, support vector machine, and deep Learning.

Figure 18

Mean Square Error Comparison for the Three Methods Under Moderate-High-Dimensional Data

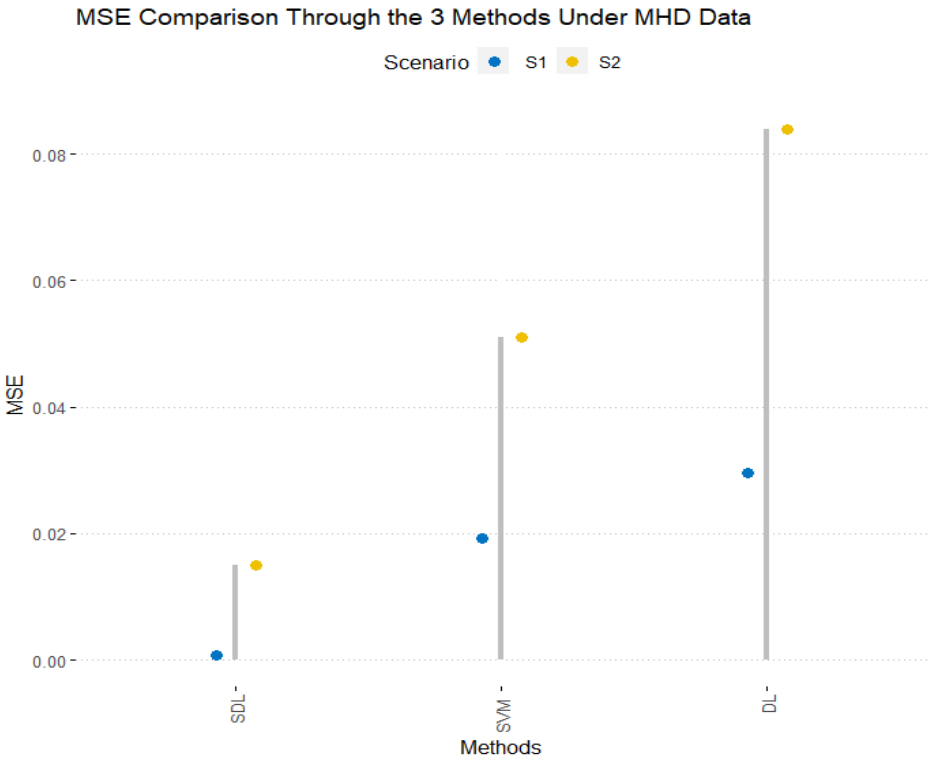


Figure 19 shows the values of the mean square error comparison under big-high-dimensional data (BHD), which is when covariates are sizes p = (1000, 2000, 5000), cross the three models, support vector machine (SVM), deep learning, and the hybrid of super learner and deep learning. The lowest mean square error was in super deep leaning (SDL) model for both scenarios, compared to the other two models, support vector machine (SVM), and deep Learning (DL). Which is the same observation for in Figure 18 for moderate-high-dimensional data.

Figure 19

Mean Square Error Comparison for the Three Methods Under Big-High-Dimensional Data

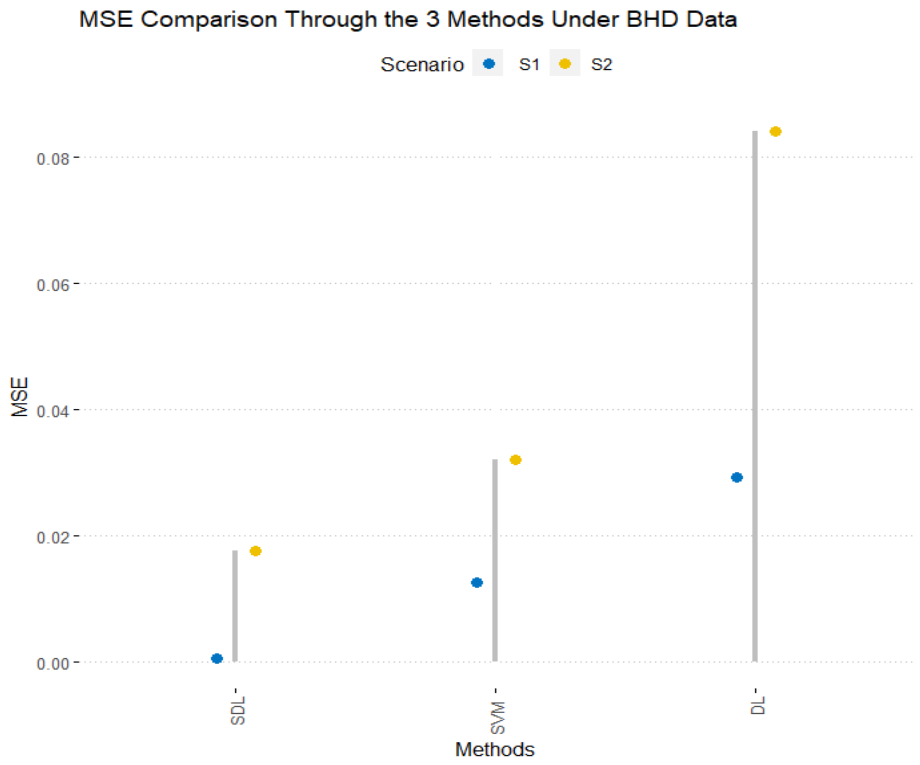


Figure 20 shows the time of computation comparison under low-high-dimensional data (LHD), which is when covariates are sizes are p = (20, 50, 80), cross the three models, support vector machine, deep learning, and the hybrid of super learner and deep learning. The most efficient time of computation was in deep leaning (DL) model for both scenarios, compared to the other two models, support vector machine (SVM), and super deep Learning (SDL).

Figure 20

Time Comparison for the Three Methods Under Low-High-Dimensional Data

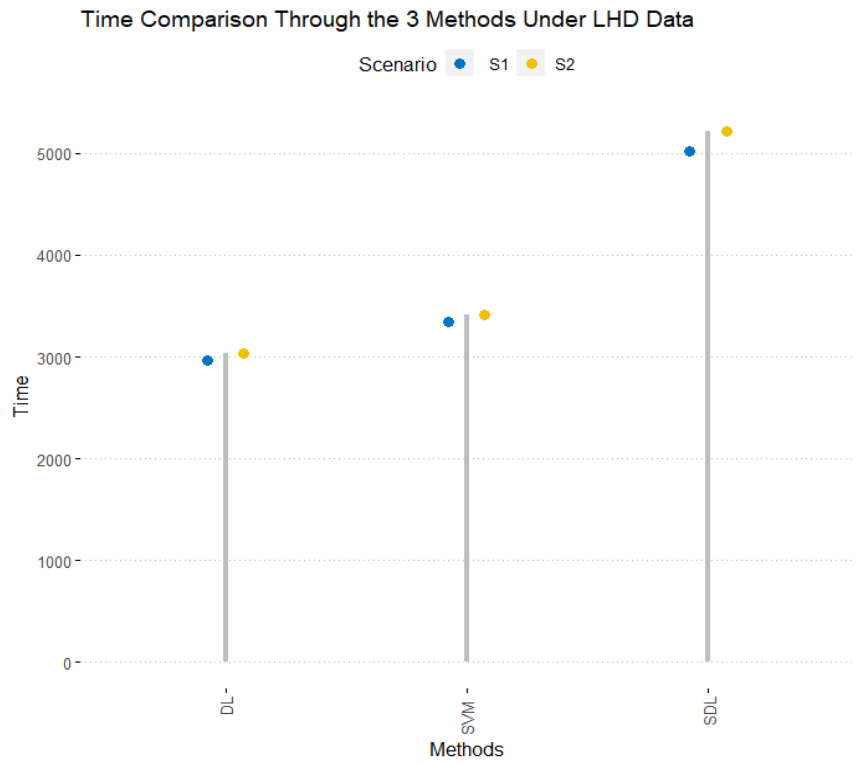


Figure 21 shows the time of computation comparison under moderate-high-dimensional data (MHD), which is when the covariates are sizes are p = (100, 200, 500), cross the three models, support vector machine, deep learning, and the hybrid of super learner and deep learning. The most efficient time of computation was in deep leaning (DL) model for both scenarios, compared to the other two models, support vector machine (SVM), and super deep Learning (SDL), which is the same observation for in Figure 20 for low-high-dimensional data.

Figure 21

Time Comparison for the Three Methods Under Moderate-High-Dimensional Data

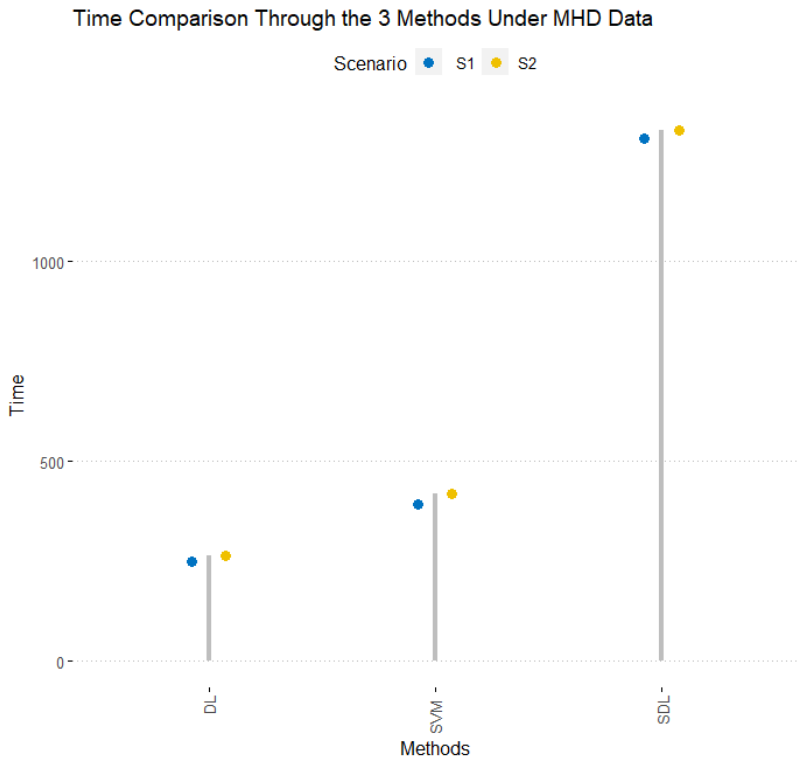
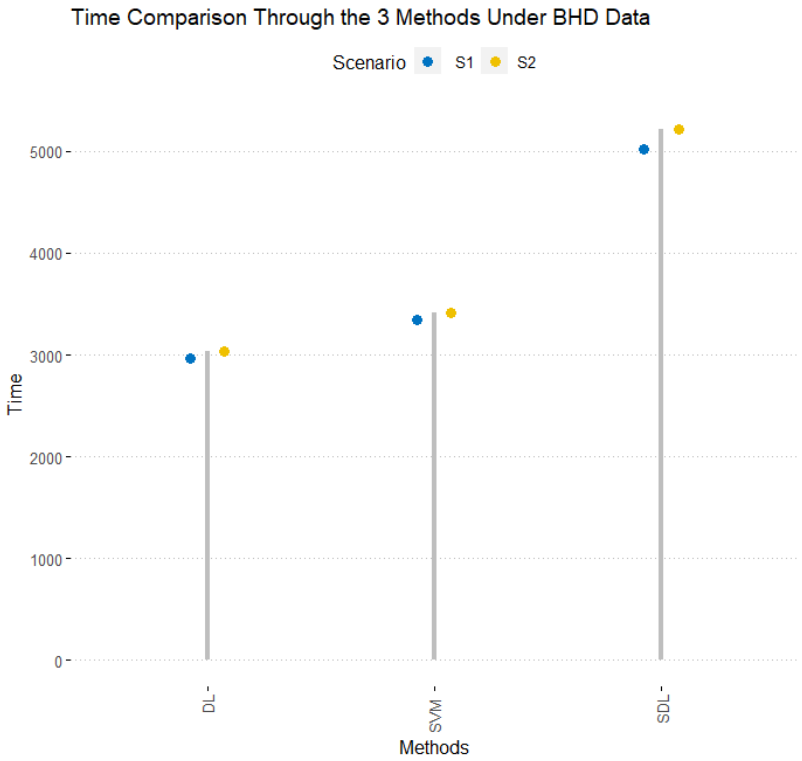


Figure 22 shows the time of computation comparison under big-high-dimensional data (BHD), which is when the covariates are sizes are p = (1000, 2000, 5000), cross the three models, support vector machine, deep learning, and the hybrid of super learner and deep learning. The most efficient time of computation was in deep leaning (DL) model for both scenarios, compared to the other two models, support vector machine (SVM), and super deep Learning (SDL). Which is the same observation for in Figure 20, Figure 21, for low-high-dimensional and moderate-high-dimensional data, respectively.

Figure 22

Time Comparison for the Three Methods Under Big-High-Dimensional Data



Real Data Analysis

The real data used is the 401k plan. This is pension accounts sponsored by employers' data that consists of 11 variables. employer offers 401(k), net total financial assets, age of participants, income, family size, years of education, individual has defined benefit pension, marital status. individual participates in IRA plan, homeowner, two-earner household. The goal is to determine the effect of the eligibility of the plan on the accumulated assets.

The data has been analyzed using these research methods, support vector machine (SVM) with double machine learning and support points splitting (SPSS), deep learning (DL) with double machine learning and support point sample splitting, and the hybrid method (SDL) of super learner (SL) with double machine learning and support point sample splitting. A

comparison to the literature work of Chernozhukov et al. (2018) using Lasso and k-fold sample splitting. The data analysis has adopted the normalization of the variables for the sake of simplifying the results and accommodating the support points sample splitting requirement for normalization.

Table 22 shows the real data of (401) k analysis results and comparison after the normalization of the variables. The literature work of Chernozhukov et al. (2018) used here is the lasso method with double machine learning (DML) and k-fold sample splitting. I compare it to this research methods, the method of support vector machine (SVM) with double machine learning and support points splitting (SPSS), deep learning with double machine learning and support point sample silting, and with the hybrid method (SDL) of super learner (SL) with double machine learning and support point sample splitting.

The comparison shows that the lowest time of computation is under the hybrid method super deep learning (SDL) with 0.0429 followed by the deep learning method (DL) with 1.1610. The best estimation was under the lowest SE= 0.0006 with the support vector machine method (SVM) followed by the deep learning method (DL), where *SE*= 0.0056.

Table 22

Comparison of Real Data Analysis Between the Literature Method and Support Vector Machine, Deep Learning, and Super Deep Learning Methods

	Chernozhukov et al. (2018) Method	This Research Methods		
	Lasso-DML-K Fold	SVM-DML-SPSS	DL-DML-SPSS	SDL-DML-SPSS
Estimator	0.0030	0.0056	0.0095	0.0063
SE	0.0071	0.0006	0.0056	0.0065
Time (Seconds)	3.4870	28.7207	1.1610	0.0429

Note. DL-DML-SPSS is the deep learning (DL) with double machine learning and support point sample splitting, SDL-DML-SPSS = the hybrid method of super learner with double machine learning and support point sample splitting, SVM-DML-SPSS = the support vector machine (SVM) with double machine learning (DML) and support points splitting method.

CHAPTER V

CONCLUSION

This study contributed to the body of knowledge by adding a new understanding of performance exploration of the three models of causal inference under double/ debiased machine learning framework using support points sample splitting (SPSS) instead of random splitting. The best-performing model that produced the best estimation of the double machine learning causal effect with lowest mean square error (*MSE*) was the super deep learning (SDL), the hybrid model of the two learners, super learner (SL), and the deep learning (DL) using the support points sample splitting (SPSS), under both scenarios when data errors were correlated or uncorrelated, and in the three levels of the high dimensional data, low-high-dimensional data (LHD), the moderate -high-dimensional (MHD), and in big-high-dimensional (BHD), compared to the two other models.

But the deep learning model (DL) for double machine learning (DML) with support point sample splitting (SPSS) was the best in terms of simulation time efficiency, under both data scenarios, and for the three levels of the high dimensional data, the low-high-dimensional (LHD), the moderate-high-dimensional (MHD), and in big-high-dimensional (BHD). The support vector machine in the double machine learning framework using the support points sample splitting with high dimensional data settings was not performing well either in estimating the treatment effect of the causal double machine learning estimation (CML) or in the simulation time compared to two other models.

However, the SVM has shown a good estimation in low dimensional data framework for Scenario 1 and in low dimensional real data of 401(k), but it hasn't outperformed the hybrid model of super learner and deep learning (SDL) on the simulation study under moderate-high-dimensional (MHD), and in big-high-dimensional (BHD).

Based on this conclusion, I recommend using the super deep learning (SDL) model if the user is targeting to get a result that is optimum in terms of the quality of the treatment effect of the causal double machine learning (CML). However, if the users want to get time-efficient results from the statistical data analysis using causal double machine learning with support points sample splitting (SPSS), then the deep learning method will be the best option.

This study does not recommend leaving behind advanced machine learning methods such as super learner (SL) and deep learning (DL) and using the support vector machine. As the latter algorithm did not show better performance compared to the two former ones either in terms of the quality of the causal double machine learning (CML) treatment effect estimation or in terms of the time efficiency of the computation.

Limitations

Machine learning algorithms are based on the artificial intelligence paradigm, so to perform effectively it needs high computing hardware. Finding high-performing computing hardware was expensive and a limitation of this study due to the costly machines that can guarantee the high computing performance which the machine learning algorithms require. So, to assuage this challenge and limitation, I have used the Rocky Mountain Advanced Computing Consortium (RMACC) provided by the University of Colorado Boulder (UCB). I recognized that using the high-performance computing from RMACC demanded new skills to develop, to learn, and to master which required an investment in both time and effort.

Future Work

Based on the limitation I have encountered in terms of the computational high-performance needs, I would recommend a future work about this perspective, how to make this high-end hardware easily reachable for common use. Or making the machine learning algorithms less computationally intensive by developing specific computing machines to manage that.

The causal inference from observational data is gaining popularity recently due to the strong inference that can produce and because of the strong result, we can get from the relationship between the cause-effect compared to only the prediction. So, developing this area will contribute not only to the statistics field but also to the applied sciences, such as health sciences, social sciences, and economic science, yet, I have found limited resources to handle causal inference in observational data. Furthermore, developing the causal double machine learning field is becoming a need, and developing its theory from what has been studied so far is crucial.

Also, as this work is based on the conditional independence assumption (CIA), future work could explore the case when this assumption is not met which will require the artificial intelligence (AI) approach.

Finally, I will recommend along with the development of computational resources and elaborating more on the theory of causal double machine learning (CML), developing statistical practices, such as developing new models and new methods that can produce a better performance in terms of the quality of the treatment effect and in terms of the computational time efficiency.

REFERENCES

Agboola, O. D., & Yu, H. (2023). Neighborhood-based cross fitting approach to treatment effects with high-dimensional data. *Computational Statistics & Data Analysis*, 186, 107780.

Alanazi, S. S. (2022). An ensemble machine learning approach to causal inference in high-dimensional settings [Ph.D. Dissertation, University of Northern Colorado]. Scholarship & Creative Works @ Digital UNC. https://digscholarship.unco.edu/dissertations/919/

Bach, P., Chernozhukov, V., Kurz, M. S., & Spindler, M. (2021). *DoubleML-An object-oriented implementation of double machine learning in R*. arXiv. https://doi.org/10.48550/arXiv.2103.09603

Bach, P., Chernozhukov, V., Kurz, M. S., & Spindler, M. (2022). DoubleML-An Object-Oriented Implementation of Double Machine Learning in Python. *Journal of Machine Learning Research*, *23*, 1-6.

Belloni, A., Chen, D., Chernozhukov, V., & Hansen, C. (2012). Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica*, 80(6), 2369-2429.

Belloni, A., Chernozhukov, V., & Hansen, C. (2014). Inference on treatment effects after selection among high-dimensional controls†. *The Review of Economic Studies*, *81*(2), 608-650.

Belloni, A., Chernozhukov, V., Ivan, F.-V., & Hansen, C. (2017). Program evaluation and causal inference with high-dimensional data. *Econometrica*, *85*(1), 233-298.

Benkeser, D., Ju, C., Lendle, S., & van der Laan, M. (2018). Online cross-validation-based ensemble learning. *Statistics in Medicine, 37*(2), 249-260. https://doi.org/10.1002/sim.7320

Bickel, P., Klaassen, C., Ritov, Y., & Wellner, J. (2000). Efficient and adaptive estimation for semiparametric models. *The Indian Journal of Statistics*, 62(A), 157-160.

Bickel, P. J., Ritov, Y., & Stoker, T. M. (2006). Tailor-made tests for goodness of fit to semiparametric hypotheses. *The Annals of Statistics, 34*(2), 721-741. https://www.jstor.org/stable/25463434

Breiman, L. B. (1996). Bagging Predictors. *Machine learning*, 24, 123-140. https://doi.org/10.1007/BF00058655

Chambaz, A., Zheng, W., & Van der Laan, M. (2016, April 12). Data-adaptive inference of the optimal treatment rule and its mean reward [Working paper Series]. U.C. Berkeley Division of Biostatistics.

Che, J., & Wang, J. (2014). Short-term load forecasting using a kernel-based support vector regression combination model. *Applied Energy*, 132, 602-609. http://dx.doi.org/10.1016/j.apenergy.2014.07.064

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal, 21*(1), C1-C68.

Chernozhukov, V., Hansen, C., & Spindler, M. (2015). Valid post-selection and post regularization inference: An elementary, general approach. *Annual Review of Economics Annual, 7*(1), 649-688.

Chernozhukov, V., Newey, W. K., & Singh, R. (2022). Automatic debiased machine learning of causal and structural effects. *Econometrica, 90*(3), 967-1027. https://doi.org/10.3982/ECTA18515

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273-297. https://doi.org/10.1007/BF00994018

Cox, D. R. (1975). A note on data-splitting for the evaluation of significance levels. *Biometrika, 62*(2), 441-444. https://doi.org/10.1093/biomet/62.2.441

Drucker, H., Burges, C., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems*, 28, 779-784.

Fang, K. T., & Wang, Y. (1994). Number-theoretic methods in statistics. Chapman & Hall.

Firpo, S. (2007). Efficient semiparametric estimation of quantile treatment effects. *Econometrica,* 75 (1), 259-276.

Flury, B. (1990). Principal points. *Biometrika*, 77, 33-41.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *International Conference on Machine Learning*, 96, 148-156.

Frisch, R. (1938). Autonomy of economic relations: Statistical versus theoretical relations in economic macro-dynamics. In *Found of econometric analysis* (pp. 407-424). Cambridge University Press.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. The MIT Press.

Guo, Z., Ćevid, D., & Bühlmann, P. (2022). Doubly debiased lasso: High-dimensional inference under hidden confounding. *Annals of Statistics, 50*(3), 1320-1347.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. Springer Series in Statistics.

Heiler, P., & Knaus, M. C. (2022). *Effect or treatment heterogeneity? Policy evaluation with aggregated and disaggregated treatments*. https://doi.org/10.48550/arXiv.2110.01427

Herzberg, P. A. (1969). The parameters of cross-validation. *Psychometrika Monograph Supplement, 34*(2), 1-70.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, 36(3), 1171-1220.

Hollander, M., Wolfe, D. A., & Chicken, E. (2015). *Nonparametric statistical methods*. John Wiley & Sons, Inc. https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

Joseph, V. R., & Vakayil, A. (2021). SPlit: An optimal method for data splitting. *Technometrics, 64*(2), 166-176. https://doi.org/10.1080/00401706.2021.1921037

Kebonye, N. M. (2021). Exploring the novel support points-based split method on a soil dataset. *Measurement: Journal of the International Measurement Confederation, 186*, 110131.

Kennard, R. W., & Stone, L. A. (1969). Computer aided design of experiments. *Technometrics*, *11*(1), 137-148. https://doi.org/10.1080/00401706.1969.10490666

Klosin, S. (2021). Automatic double machine learning for continuous treatment effects. https://doi.org/10.48550/arXiv.2104.10334

Knaus, M. C. (2021). A double machine learning approach to estimate the effects of musical practice on student's skills. *Journal of the Royal Statistical Society. Series A: Statistics in Society, 184*(1), 282-300.

Kosorok, M. R. (2006). *Introduction to empirical processes and semiparametric inference*. Springer Science Business Media. http://www.bios.unc.edu/~kosorok/current.pdf

Kosorok, M. R. (2009). What's so special about semiparametric methods? *Sankhya. Series B. [Methodological.], 71-A*(2), 331-353. https://pubmed.ncbi.nlm.nih.gov/20640048/

Larson, S. C. (1931). The shrinkage of the multiple correlation coefficient. *Journal of Educational Psychology*, 22, 45-55.

Lewis, G., & Syrgkanis, V. (2021). *Double/debiased machine learning for dynamic treatment effects via g-estimation*. arXiv. https://doi.org/10.48550/arXiv.2002.07285

Mak, S., & Joseph, V. R. (2018). Support points. *The Annals of Statistics*, *46*(6A), 2562-2592. https://www.jstor.org/stable/26542875

Marron, J. S. (1994). Visual understanding of higher-order kernels. *Journal of Computational and Graphical Statistics*, 3(4), 447. https://doi.org/10.2307/1390905

Max, J. T., & Zang, E. (2019). *Semiparametric methods. Encyclopedia of gerontology and population aging*. Springer.

Meng, Z., McCreadie, R., Macdonald, C., & Ounis, I. (2020). *Exploring data splitting strategies for the evaluation of recommendation models*. https://doi.org/10.48550/arXiv.2007.13237

Mosteller, F., & Tukey, J. (1968). Data Analysis, including Statistics. In *Revised handbook of social psychology, Vol. 2* (pp. 80-203). Addison Wesley.

Müller, M., Härdle, W., Sperlich, S., & Werwatz, A. (2004). *Nonparametric and semiparametric models*. Springer.

Niederreiter, H. (1992). *Random number generation and quasi-Monte Carlo methods*. SIAM. http://dx.doi.org/10.1137/1.9781611970081

Pearl, J. (1995). Causal diagrams for empirical research. *Biometrika*, *82*(4), 702-710. https://doi.org/10.1093/biomet/82.4.702

Pearl, J. (2009). Causal inference in statistics: An overview. *Statist. Surv, 3*, 96-146. https://doi.org/10.1214/09-SS057

Pfanzagl, J., & Wefelmeyer, W. (1982). *Contributions to a general asymptotic statistical theory*. Springer-Verlag.

Picard, R. R., & Berk, K. N. (1990). Data splitting. *The American Statistician, 44*(2), 140-147. https://doi:10.1080/00031305.1990.10475704

Robinson, P. M. (1988). Root-N-consistent semi-parametric regression. *Econometrica*, 56, 931-54.

Rodriguez-Poo, J. M., & Soberón, A. (2017). Nonparametric and semiparametric panel data models: Recent developments. *Journal of Economic Surveys*, 31(4), 923-960. https://doi.org/10.1111/joes.12177

Schölkopf, B. (2000). The Kernel trick for distances. *Advances in Neural Information Processing Systems,13*, 301-307.

Schölkopf, B., & Smola, A. J. (2018). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. The MIT Press.

Shao, J. (1997). An asymptotic theory for linear model selection. *Statistica Sinica*, 7(2), 221-264.

Simon, F. H. (1971). *Prediction methods in criminology including a prediction study of young men on probation*. H.M.S.O.

Simon, H. A. (1953). Causal ordering and identifiability. In W. C. Hood & T. Koopmans (Eds.,), *Studies in econometric method* (pp. 49-74.). Wiley and Sons, Inc.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199-222.

Snee, R. D. (1977). Validation of regression models: Methods and examples. *Technometrics, 19*(4), 415-428. https://doi.org/10.1080/00401706.1977.10489581

Speckman, P. (1988). Kernel Smoothing in Partial Linear Models 1988. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(3), 413-436. https://doi.org/10.1111/j.2517-6161.1988.tb01738.x

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B, Methodological*, *36*(2), 111-147. https://doi.org/10.1111/j.2517-6161.1974.tb00994.x

Székely, G. J., & Rizzo, M. L. (2013). Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, *143*(8), 1249-1272. https://doi.org/10.1016/j.jspi.2013.03.018

Vakayil, A., & Joseph, V. R. (2022). Data twinning. *Statistical Analysis and Data Mining: The ASA Data Science Journal*. https://doi.org/10.1002/sam.11574

Van der Laan, M. J., & Dudoit, S. (2003, April 7). *Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples* [Working paper]. U.C. Berkeley Division of Biostatistics.

Van der Laan, M. J., Dudoit, S., & van der Laan, A. W. (2004, February 26). *The cross-validated adaptive epsilon-net estimator* [Working paper]. University of California Berkeley.

Van der Laan, M. J., & McKeague, I. W. (1998). Efficient estimation from right-censored data when tailure indicators are missing at random. *The Annals of Statistics*, *26*(1), 164-182. https://www.jstor.org/stable/119982

Van der Laan, M. J., Polley, E., & Hubbard, E. A. (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology*, *6*(1), 1-23.

Vapnik, V. (1982). *Estimation of dependences based on empirical data*. https://link.springer.com/book/10.1007/0-387-34239-7

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer Verlag.

Vapnik, V., & Chervonenkis, A. (1974). *Pattern recognition theory, statistical learning problems*. Nauka, Moskva.

Varaku, K. (2021). Essays on causal inference and treatment effects in productivity and finance: double machine learning with deep neural networks and random forest [Doctoral Dissertation, Rice University]. Rice University Electronic Theses and Dissertations. https://hdl.handle.net/1911/110418.

Wolfgang, K. H., Müller, M., Sperlich, S., Werwatz, A., Wolfgang, H., & Mnller, M. (2004). *Nonparametric and semiparametric models*. Springer.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*(2), 241-259. https://doi.org/10.1016/S0893-6080(05)80023-1

Wyss, R., Schneeweiss, S., van der Laan, M., Lendle, S. D., Ju, C., & Franklin, J. M. (2018). Using super learner prediction modeling to improve high-dimensional propensity score estimation. *Epidemiology*, *29*(1), 96-106. https://doi.org/10.1097/EDE.0000000000000762

Yadav, S., & Shukla, S. (2016). Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification [Conference session]. *6th International Conference on Advanced Computing (pp.* 78-83). *IEEE.* https://doi.org/10.1109/IACC.2016.25

Yang, J., Chuang, H., & Kuan, C. (2020). Double machine learning with gradient boosting and its application to the Big N audit quality effect. *Journal of Econometrics*, 216(1), 268-283.

Yang, Y. (2007). Consistency of cross validation for comparing regression procedures. *The Annals of Statistics*, 35(6), 2450-2473.

Yao, L., Chu, Z., Li, S., Li, Y., Gao, J., & Zhang, A. (2021). A survey on causal inference. *ACM Transactions on Knowledge Discovery from Data, 15*(5), 1-46.

Young, S., Abdou, T., & Bener, A. (2018). Deep super learner: A deep ensemble for classification problems. *Advances in Artificial Intelligence Canadian AI*, 10832, 84-95.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). *Dive into deep learning*. Cambridge University Press. https://d2l.ai/

Zhang, W., Zhao, X., Zhu, Y., & Zhang, X. (2010). A new composition method of admissible support vector kernel based on reproducing kernel. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control, Information Engineering,4,* 432-440.