Analysis of a Few Domain Adaptation Methods in Causality

Umar Isyaku Abdullahi

A thesis presented for the degree of
Doctor of Philosophy

School of Computer Science Electrical and Electronics Engineering
University of Essex, UK
January, 2020

Publications

U. I. Abdullahi, S. Samothrakis and M. Fasli," Causal Inference with Correlation Alignment," 2020 IEEE Conference on BigData

U. I. Abdullahi, S. Samothrakis and M. Fasli, "Counterfactual domain adversarial training of neural networks," 2017 International Conference on the Frontiers and Advances in Data Science (FADS), Xi'an, 2017, pp. 151-155, doi: 10.1109/FADS.2017.8253217.

Abstract

Understanding the effect of intervention is of great importance in many domains such as marketing, governance, health, economics, social science, etc. An ideal approach for estimating the effect of intervention requires conducting experiments which are often unethical, expensive, time consuming, or even impossible; leaving interesting business and research questions un-answered. Nowadays, data from businesses, government databases, and electronic medical records are generated in large amount and at unprecedented rate, making the use of observational study a viable alternative. However, the data posses bias between the treated and the control subjects posing a great challenge for this task. Machine Learning (ML) and Deep Learning (DL) models are recently deployed for causality and have achieved a state of the art results. "Correcting" the bias through aligning the distribution of the treated and control in the form of domain adaptation is shown to be an effective technique for estimating causal parameters. However, most often, these models involve complex DL architectures. There are tons of Domain Adaptation (DA) methods developed to align the shift between the source and target distribution in classical ML.

In this thesis, following the Potential Outcome framework with binary treatment setting, we bring the idea of correlation alignment methods, adversarial training, and a parallel two streams architecture from domain adaptation into causality. But we initially built simple baseline Neural Networks (NN) models in each case which are optimized and evaluated. This is to understand the effectiveness and performance of the simple models in causality without any form of distribution alignment mechanisms proposed in domain adaptation literature. Then the DA components of these models are incorporated as an additional loss to the baseline models in each case, and are evaluated on four most widely used benchmarks. Our results show that incorporating additional DA losses are generally not effective for causality. The simple baseline models were able to achieve state-of-the-art results on some metrics. Suggesting that DL models with hyperparameter tuning could estimate causal parameters without necessarily the need for specialized regularizations. Moreover, many of the metrics could be estimated effectively with linear versions of these models. It was found that no method is superior over others on all the datasets. However, methods based on shared weights have fairly performed better than model based on unshared weights. Further more, using geodesic and Euclidean distances for correlation alignments produced similar results, implying some robustness to distance measure.

Dedication

I dedicate this thesis to my late father Alhaji Isyaku Abdullahi who was my source of encouragement, and inspiration.

Acknowledgement

This journey was full of challenges and interesting memories which are common but come with peculiarities. I was lucky to have got dedicated supervisors in person of Prof Maria Fasli, and Dr Spyros Samothrakis whom have been very supportive right from the onset. Their patience, guide, and professional skills I acquired through their effective supervision would never be forgotten. I sincerely appreciate it. I also appreciate my board chair, and member respectively in person of Dr Micheal, and Dr.Sebastian for their constructive observations. It has become necessary to acknowledge my sponsor the Petroleum Technology Development Fund (PTDF) for the financial support. My special thanks to my dear wife, who has shown understanding, and offered an un quantifiable emotional support throughout this period. A deep appreciation to my lovely mother for her support and prayers. You have been the pillar of my life. I wont forget the goodwill messages from family and friends. My research colleagues, office mates and all members in the school of computing, Institute for Analytics and Data Science, and Local Government and Business Research Center, thank you for the contribution in one way or the other. Above all, I thank the almighty God who spared my life in good health to witness the end of this memorable journey.

Contents

List of Figures

List of Tables

List of Symbols

γ_i	eigenvalues
$\hat{y}$	estimated outcome
λ	Diagonal matrix of eigenvalues
λ	degree of adaptation parameter
λ_m	Lambda for r_m
λ_w	Lambda for r_w
$\mathbf{E}(X)$	expected value
$\mathcal{C}^F$	Control group
$\mathcal{D}$	dataset
$\mathcal{D}^F$	Factual data
$\mathcal{D}_t^F$,$\mathcal{D}_c^F$	Factual treated and control data
$\mathcal{H}\Delta\mathcal{H}$	Divergence measure
$\mathcal{M}$	Distribution
$\mathcal{M}_s, \mathcal{M}_t$	Distributions of the source and target
$\mathcal{R}^l$	internal state or representations
$\mathcal{R}_0^l, \mathcal{R}_1^l$	internal representations for the treated and control
$\mathcal{T}^F$	Treated group
$\mathcal{W}_c$	Control weights
$\mathcal{W}_t$	Treated weights
$\mathcal{X}$	Input Space
μ_i	eigenvectors
$\|.\|_F$	Frobenius norm
A	a matrix
C_0, C_1	Co-variance matrices of $\mathcal{R}_0^l$, $\mathcal{R}_1^l$

$Cov(x, y)$	covariance between x and y
d	dimension of activation layers
Dij	distance between two subsets
g	Labelling function
g_s, g_t	Labelling functions for source and target
L_t, L_c	Treated and control losses
n	number of samples
$S(X)$	covariance matrix of X
t	treatment variable
U	Matrix of eigenvectors
u	un observed confounder variable
W_i	model weights
X	data matrix X
y	variable
z	an instrumental variable
$\bar{x}$	mean of x
$\bar{y}$	mean of y
h	hypothesis function
r_m	MMD regularization
r_w	weight regularizer
s^2	sample variance
x	variable
$Y(1)$, $Y(0)$	potential outcomes

Chapter 1

Introduction

Understanding the effect of interventions (treatment, policy, advert, action/inaction, etc) is a vital activity for the strategic development/growth of any business/government. For instance, governments do roll out different policies and programs targeted at the various sectors (education, employment, healthcare, etc) of the society as intervention. A government social policy on employment may involve a program that provides training on say "*Machine Learning*" for its college graduates aimed at reducing unemployment rate. Similarly, a policy on healthcare could provide incentives to pregnant women as motivation for antenatal visits to reduce maternal mortality rate. On the other hand, to implement associated economic policy, government may be interested in understanding whether tax hike would be solely responsible for job losses.

In business, companies often spend so much on advertisement annually with a view to attract new customers as well as prevent customer churn. Whether these resources spent on the marketing actually attracted new customers or it was a mere coincidence (customers came on their own within the advertisement period), reliable ways are required to ascertain the veracity of such claims. Furthermore, it is common practice for online platforms, to have a new look, equipped with new features or changes added to systems. These are done with the hope to enhance customer satisfaction, efficiency or ease of use. There is the need to measure if those changes bring the desired outcome. Similar requirement is evident in pharmaceutical companies, where clinical trials are carried out to determine how effective a new drug is.

In social science, a piece of research may seek to study the effect of attending private schools compared to public schools on future accomplishment of persons.

In most of the examples highlighted in earlier, interventions involve heavy financial implications. The decision on whether to continue with intervention (policy, treatment) or not in each case rests on whether or not it is yielding the desired results. A decision maker needs some reliable way to estimate the effect of a given intervention. Unfortunately, questions such as: What is the effect of a policy on some people or business are causal and not association in nature.

A gold standard approach to answering questions of these nature is the Experimental study or Randomized Control Trials (RCTs) [1], [2] where people are randomly partitioned into groups. In the case of two partitions, one of the groups receive a new intervention and the other receive a conventional intervention or no intervention at all [1],[3],[4],[5].

In spite of it being the most ideal approach, there are scenarios where conducting experiments are unethical, economically unrealistic, or practically impossible to answer questions of interest [6],[7],[8]. Consider a situation where the study of interest is whether heroin addiction causes improvement in human intelligence. It will be grossly unethical to contemplate exposing subjects to such exposure [6]. More so, it is very difficult to avoid cases of dropout [8]. Another issue with RCTs is that in circumstances where conducting them is feasible, they are mostly expensive to run. For instance, it is reported that average costs of a multi- phase trials in vital thereapeutic areas like respiratory system, oncology, and anesthesia are $115.3 million, $78.6 million, and $105.4 million respectively [9].

In this regard, it is neither an option to allow limitations of experiments restrict our drive to investigate several important questions of interest nor it is an option to resort to opinion based inferences. Where experiments are unattainable, Observational Study remains a viable alternative [10]. Observational study relies on existing past records or history (eg, medical records) of subjects in databases,[11],[10], [12]. Availability of observational data brings a source of relief because we do not need to conduct experiments. However, there are concerns pertaining the use of observational data for causal inference; the mechanism used in assigning subjects to either the treatment or the control groups is unknown [13]. The treatment assignment is beyond the investigator's control, which potentially subject it to spurious association between the features and the treatment assignment. Therefore, direct causal effect estimation in case is not plausible. Consequently, drawing causal claims directly from these kind of data would be erroneous and unreliable[14]. For example, left ventricular assisting device (LVAD) treatment is mostly applied to high-risk patients with severe cardiovascular diseases before heart transplant, the distribution of features among these patients will be significantly different to the distribution among non-LVAD treated patients [15].

One of the challenges in causal inference with observational study is therefore, how to effectively estimate an unbiased causal effect of a treatment. Another challenge is that of recent advances in precision medicine [16],[17] and also personalized recommender systems [18], [19], [20] which demand intervention effect at individual level. This presents a new shift from making decision based on group level such as Average Treatment Effect (ATE) to measuring the effect on each patient/customer, a quantity called the Individualized Treatment Effect (ITE). The demand for ITE is plausible being that individuals respond differently to the same intervention, the so called treatment effect heterogeneity. It is reported that prescription of drugs to Amyotrophic Lateral Sclerosis (ALS) patients based on group level causal parameter could lead to death [16].

In problems like this, standard machine learning algorithms could not reliably estimate the effect of these interventions. The reason being that without further plausible assumptions, causal questions could not be answered validly using correlation based models. If a classical machine learning algorithm is deployed in say a marketing campaign, it would at best predict customers that are likely to buy a product. However, this outcome includes customers who could have bought the product anyway.

1.1 Advances in Causal Inference

Causal inference with non-experimental data (Observational study) as opposed to experimental studies has been a challenge mainly due to the presence of confounders (observed or un-observed). Thus, confounding—a situation where features relate with both treatment and the outcome poses identifiability problem as whether the effect observed is due to treatment or the confounders (more details of confounding in section 2). For causal estimates to be unbiased, confounders need to be adjusted. Not all confounders are observed, however, in general, we could not ascertain through test whether all confounders are observed or not. Hence, we often assume there is no un-measured confounder (s) [6]. Different approaches proffer ways of adjusting them to aid comparison between the two groups (treated and control). Without the adjustment, causal estimates would be misleading and unbias of the true treatment effect. Over the last three decades, many statistical techniques were deployed for causal inference with observational study . Matching covariates (features) to balance the treated and the control groups are done through: Exact Matching, Nearest Neighbour matching, Optimal Matching [21], [22] among others (details of these methods are in section 2). Under Modelling approaches, Propensity Score (PS)– probability of being treated given the features are often used in social sciences. PS approaches are deployed as a remedy to matching directly on features. Directly matching on high dimensional features causes computational challenges [23],[22]. One of the prominent methods is the Propensity Score Matching (PSM) [24],[22] [25] where matching is done on a single feature (the propensity score). Subclassification (Stratification) on the propensity score [26],[27] is another PS method where the scores are discretized to form strata of similar PS. Subjects in a strata with opposing treatments are compared. More details on these approaches are provided in section 2.

A slight mis-specification of PS could result in huge bias, Covariate Balancing PS [28] mitigates it by choosing parameters that best maximize the balance between the treated and control groups.

Inverse Probability of Treatment and Weighting (IPTW) [29] uses weights based on PS to create a synthetic sample in such away that the features are independent of the treatment assignment. Kernel Matching [30] uses a kernel function to transform PS to a distance and constructs a counterfactual of each treated subject using all control subjects based on their distance from the treated subject. In the event of model mis -specification, Doubly Robust [31] combines both PS and regression adjustment, such that at least one of the two methods if correctly specified suffices to produce an un-biased estimate of the treatment effect. In the presence of unobserved (unmeasured) confounders, Instrumental Variables(IV) are used in estimating ATE [32],[33], [34], [35], [36]. Regression Discontinuity Design(RDD) [37], [38], [39] use a cut-off criteria and extract subjects just above and below the cut-off to create a "region" of similar feature distribution. Regression Adjustment is also used in controlling confounders [40]. Most of the methods discussed earlier are used to estimate ATE and ATT.

Further more, under the modelling approach, Machine learning and Deep learning techniques are deployed more recently for causality [41], [42], [43], [44], [45], [46],[47]. For instance, [47] uses BART ensemble methods [48] for learning heterogeneous causal effect . Assuming no unmeasured confounders, some Deep learning

methods for causality learn representations of the features and then fit a function of the representation and the treatment on the factual outcome [41],[42]. Within the machine learning literature, [41] draws a connection between causal inference with observational study and domain adaptation –a sub-field in machine learning that trains models while taking into account the distribution shift between the source and the target data distribution. Domain adaptation methods for causality have shown state of the art performance [41], [45].

[41] shows that causality could be formulated as a domain adaptation problem which followed the development of many deep learning architectures such as: Multi-task Gaussian Process (CMGP) [49], Generative Adversarial nets for Inference of Individualized Treatment Effect (GANITE) [50], and local Similarity Preserved Individual Treatment Effect (SITE) [51]. Many of these deep learning architectures are often complex. Thus, in this thesis, we consider developing relatively simple NN models optimized with standard loss functions without any additional regularization or training procedure for causality. We then extend these simple baseline models to include additional losses and training procedure pioneered in domain adaptation literature for causality under the assumption of no-unmeasured confounders, and binary treatment setting.

1.2 Thesis contribution

Motivated by the work of [41] which draws the connection between Causality and Domain Adaptation, in this thesis, we first develop simple NN model architectures relative to the complex models proposed recently for causality and then build on these simple baseline models by incorporating some regularizations and training procedure proposed in classical domain adaptation for causal inference . In our work, we leverage the idea proposed in [41] which connects causality and domain adaptation. With the understanding that on the one hand, the main challenge in causality with observational study is the bias between the distributions of treated and the control subjects, and on the other hand, domain adaptation methods align different distributions for effective prediction. It sounds a feasible idea to bring some DA methods as distribution alignment mechanisms for aligning the treated and the control into causality. But at first, we develop relatively simple NN models which are optimized with standard loss function (MSE) without any additional regularization. We consider them as baselines. The idea of building these baseline models is to evaluate the effectiveness and performance of the simple models in causality without any form of distribution alignment mechanisms proposed in domain adaptation literature. Then we extend these baseline models by incorporating the distribution alignment losses proposed in [52], [53], [54], and [55] for causality. The choice of these methods is done to reflect the diverse, yet popular domain adaptation approaches. We chose [52], [53], and [54] methods based on weight sharing, although the training in [54] is adversarial. While the approach in [55] involves unshared weights. We are interested in investigating and answering the following questions: Could simple NN models optimized with standard loss functions effectively estimate causal parameters? Could incorporating domain adaptation alignment losses primarily developed for DA offer any benefit for causality? Also, to investigate if any class (or methods) chosen performs better than others (e.g. Does geodesic distance offer better performance over Euclidean for correlation alignment methods?). The four mod-

els are implemented under the no-unmeasured confounder assumption with binary treatment (treated and control). Out of the four models, two of them are based on correlation alignment. These methods differ only in the distance metric used in computing the alignment. Once their domain adaptation part is set to zero, the two models collapsed into one and the same baseline model. The alignment loss in each case is added to the MSE and jointly optimized in a closed loop. Aligning the distribution of treated and the control is intended to remove (or reduce) the bias between the distributions of treated and the control. More over, the third DA method introduced for causal inference involve adversarial training. The alignment mechanism here is a classifier not a distance function as is the case in the correlation alignment methods. The classifier attempts to distinguish between the subjects from the two groups (treated and control), and the overall aim is to prevent the classifier from doing so by producing invariant representations through adversarial process proposed in [54] in each forward propagation. The fourth model is based on two parallel streams architecture. Initially, a baseline model for this model is developed by training a NN using two streams, one for the treated and the other for the control examples with standard loss (MSE). After the baseline model is optimized, additional domain adaptation regularizations reported in [55] is then incorporated. As highlighted earlier, even though other contributions in causality like [45],[41] also learn invariant features, no work in causality has deployed the use correlation alignment [52], [53] , adversarial training [54], and separate streams [55] for causality. In addition, many the deep learning/ NN models proposed for causality are often complex, in this thesis, we also trained simple NN baseline models that were able to achieve state-of the- art results. And in some cases achieve better results than the state-of-the-art with no special regularizations or complicated NN architecture. We evaluate each of them using four (4) most popular causality benchmarks, namely: IHDP, NEWS, JOBS, and TWINS.

1.3 Thesis Structure

The rest of the thesis is organized as follows: we provide background and literature review in chapter 2. The chapter contains six sections. In section 2.1, we discussed *Causation* being the central idea in our work alongside with *Correlation* a concept related but often mistaken for the former. Section 2.2 introduced statistical methods for causality under which the concept of *Matching*, *Instrumental variables*, *Regression Adjustment*, *Randomization* etc. are discussed. Paradigms for Causal Inference: *experimental*, and *observational study* are in section 2.3 . Moreover, in the section, the difference between *Causal Inference* and *Causal discovery* as frameworks for answering *what* if and *why* questions respectively are discussed. A key statistical framework used in this thesis called the *Potential Outcome* framework together with causal assumptions are detailed in subsection 2.3.4. The chapter also comprises of section 2.4, a section detailing general introduction to ML approach, *Neural Networks and Deep Learning*, with their associated concepts such as loss functions, activation functions, and gradient descent. More so, the concept of *Representation Learning*, and *Maximum Mean Discrepancy (MMD)* are introduced in the section. In addition, the chapter discussed domain adaptation in section 2.5 which include framework for domain adaptation, domain adaptation types and approaches. The chapter finally closes with section 2.6 which presents ML methods for causality.

Chapter 3 provides detailed description of the new methods with section 3.1 dedicated to problem formulation. In the chapter, section 3.2 discusses the *two correlation alignment* methods under two subsections 3.2.1 and 3.2.2 respectively. Discussion on the *Counterfactual Adversarial and the Counterfactual Two Streams* methods are respectively in sections 3.3, and 3.4 . The chapter concludes with section 3.5 which summarizes the contribution of the chapter. Chapter 4 contains three sections: section 4.1 discusses the four benchmarks, one in each subsection. Section 4.2 describes all the evaluation metrics reported, and section 4.3 explains model training procedure for each model. Chapter 5 is the Results and Analysis chapter .It has sections 5.1, 5.2, 5.3, and 5.4 of the chapter dedicated to discussion pertaining the results of all the models and their variants on IHDP, NEWS, JOBS, and TWINS respectively. Whereas section 5.5 compares the performance of our models against each other. Test of significance on the best results is presented in section 5.6. In section 5.7 of the chapter, computational time relative to the models is discussed. In section 5.8, the implication of our results with regards to our research questions and related literature were analyzed. Whereas, section 5.9 compares the performance of our models against state of the art and other baseline.

Finally, chapter 6 closes the thesis with reflection on the need for the research and the major challenges faced. These are detailed in section 6.1, which highlights the research gap and contribution of this thesis, and in section 6.2 which pointed out limitations of the work and recommendations for future work.

Chapter 2

Background and Literature Review

2.1 Correlation vs Causation

The concept of ***Causation*** is at the center of this thesis and it is often mistaken for ***Correlation***. The expression – *Correlation is not causation* is popularized even at college level to stress that having two events changed together does not necessarily mean one causes the other. We would be discussing what these concepts are, and how they are related. Interestingly, in addition to ***Causation*** being central to this work, one of the methods implemented uses the concept of ***Correlation***, making the understanding of the duo crucial.

2.1.1 Correlation

Two events are said to be correlated or statistically dependent if they tend to occur together. Moving together could be, when one of the events increases, the other increases as well, in the case of positive correlation shown in Figure 2.1. It could also

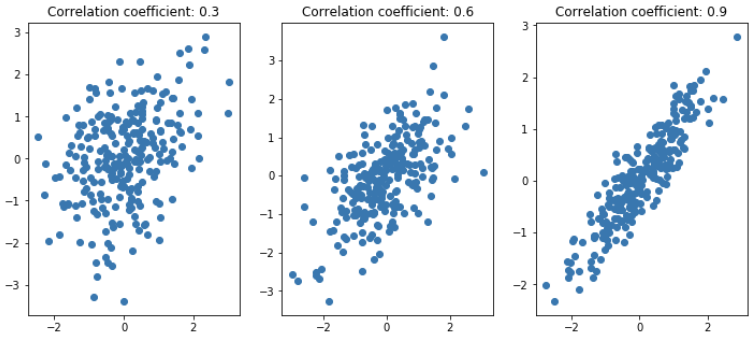


Figure 2.1: Positive Correlation with varying coefficients
Source: [56]

be that as one increases, the other decreases- in the case of a negatively correlated relationship shown in Figures 2.2, and 2.3. The Pearson's correlation coefficient is a common quantity used for measuring the strength of a linear relationship. This co-efficient has a value in the range [-1,1].

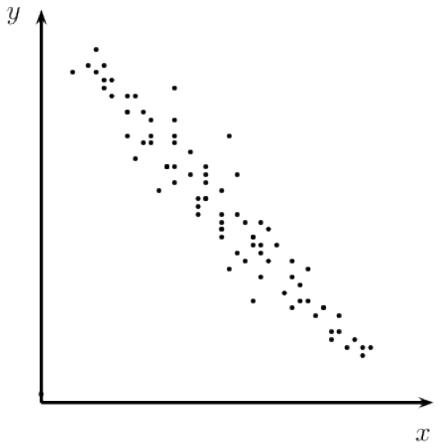


Figure 2.2: Strong Negative Correlation.

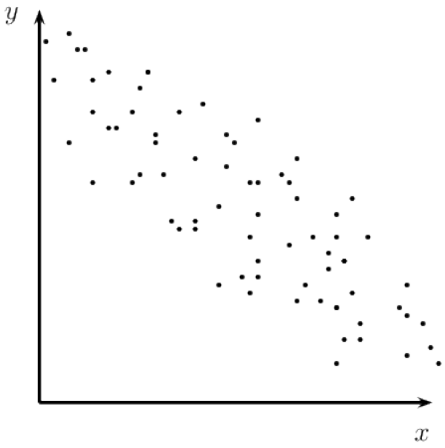


Figure 2.3: Weak Negative Correlation

Source: [57]

Correlation coefficient tells quite well that two events move in tandem, but does not tell by what amount Y goes up as X moves [57]. In situations where the relationship between X and Y is non-linear as shown in figure 2.4, the coefficient would be inaccurate.

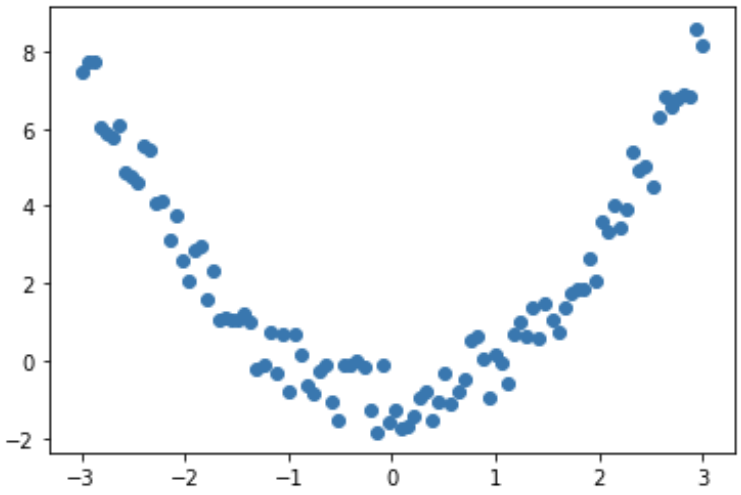


Figure 2.4: Non-linear relationship between X and Y
Source: [56]

Covariance is a quantitative measure of the extent to which the deviation of a variable (x) from it's mean matches the deviation of the other (y) from it's mean [58]. For any two variables x , y, with sample size n, the covariance between them is given by

$$Cov(x, y) = \sum_{i=1}^{n} \frac{(x_i - \bar{x})(y_i - \bar{y})}{n - 1} \tag{2.1}$$

Where $\bar{x}$, $\bar{y}$ are the means of x and y respectively. The covariance between a variable by it's self reduces equation 2.1 to a sample variance s^2.

$$s^2 = Cov(x, x) = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n - 1} \tag{2.2}$$

For an n x p data matrix X covariance matrix denoted by $S(X)$ is given by

$$S(X) = Cov(X, X) = \begin{Bmatrix} s_1^2 & s_{12} & s_{13}.. & .. & .. & s_{1p} \\ s_{21} & s_2^2 & s_{23} & .. & .. & s_{2p} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ s_{n1} & s_{n2} & s_{n3} & .. & .. & s_p^2 \end{Bmatrix} \tag{2.3}$$

For every ith and jth entries of matrix X, the covariance of the two variables s_{ij} is computed. In the cases where $i = j$, then s_{ij} is a sample variance (covariance of a variable by it'self). These occur along the leading diagonal of the covariance matrix S. Each s_{ij} tells us how the two variables i and j move together. The Correlation between any ith and jth entry in S denoted by r_{ij} is the $Cov(i, j)$ normalized by their respective variance [58]. The correlation matrix P of the data matrix X could be written as

$$P(X) = \begin{Bmatrix} 1 & r_{12} & r_{13}.. & .. & .. & r_{1p} \\ r_{21} & 1 & r_{23} & .. & .. & r_{2p} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ r_{n1} & r_{n2} & r_{n3} & .. & .. & 1 \end{Bmatrix} \tag{2.4}$$

The ones at the leading diagonal of P are the correlation values between a variable by it'self.

Statistical analysis exemplified by estimation, hypothesis testing, and regression techniques are aimed at estimating population parameters from a sample data. Such parameters could be used to infer correlation (relationships) between features, estimate their uncertainties (probabilities) or beliefs for past or upcoming events. We could as well use those parameters to update these beliefs in the light of new observed evidence. These techniques tackled these problems pretty well as long as experimental conditions remain unchanged (static). For example, it is easier to assert that there is a statistical relationship (correlation) between disease and symptoms. Statements like "*conditional probability of a disease given specific symptoms*" are formal probability language used in statistical analysis, which conveys information about how likely it is to observe a disease given that we observed some symptoms. On the other hand, Causal analysis does not only make inference under static conditions, but also involve doing so under a changing (dynamic) condition. Such changes could be as a result of external manipulations like treatment or intervention. Causation as opposed to correlation requires not only the statistical relationship but also the direction of influence between objects.

2.1.2 Causation

The notion of Causation had been discussed in the field of Philosophy before it was discussed Statistics. For instance, the Regularity model of causation is one of the early causal models [59]. In the book *Systems of Logic*, Mill argues that the discovery of empirical laws of patterns is the goal of science [59]. Mill's popular method of Direct Difference assumed the use of two different instances that are the

same in all aspects except for the intervention of interest. An assumption found to be unrealistic especially in the social sciences [59].

David Hume agrees with Mill in his book titled *An inquiry into Human Understanding* [60]. He argues that *How could one know that flame causes heat?* He continues, *such questions could only be addressed by referring to the history if such cause-effect has been regular and consistent* [60]. He showed that such requires an empirical approach that is only necessary if contiguity, succession, and constraint conjuncture are satisfied [60].

Albeit, contrary to Hume and Mill, David Lewis made a crucial contribution in causation based on counterfactual reasoning [61]. His contribution has played a key role in popularizing this philosophical reasoning. Which is, viewing causation in terms of *what if I had done things differently.* Counterfactual reasoning is an idea about imagination of an outcome if something was done in a different way. Imagining about if something was done differently implicitly means that something was done in one way and an outcome was observed. We are now imagining what the outcome would be if we decide to take a different action [61]. Within the context of our work, this concept provides a way of comparing two treatment outcomes, that is, comparison between a treatment (actually) given and it's outcome observed and another treatment we only "imagined" if it were to be given what would have been the outcome (i.e. the counterfactual). This idea is central to the Potential Outcome framework discussed in subsection 2.3.4.

2.1.3 Confounder and Confounding Bias

A confounding variable or confounder is a feature/factor that is not part of a causal study, yet is associated with both the factor being investigated (exposure, treatment, risk factor, etc.) and the outcome (dependent variable). A confounding factor must:

- be associated with both the exposure and the outcome.
- be distributed unevenly between the two groups.
- neither be an effect of the exposure nor a factor in the causal pathway of the outcome [62].

These properties are illustrated in Figure 2.5. For instance, suppose a study is set to answer a hypothesis that coffee drinkers have more heart diseases than does who don't. It is known that coffee drinkers may smoke more cigarettes than those who do not drink coffee. Smoking is strongly associated with coffee drinking and is also known to be a risk factor for heart diseases (outcome). In this case, smoking "mixed" or "blurred" the true effect of coffee drinking thus is a confounder.

Consider another case where an investigator is interested in studying the effect of a new expensive cancer drug on patients. Socio-economic status is a potential confounder, being that the treated group will contain predominantly wealthy patients, whereas the control group would contain patients from lower socio-economic class. Controlling for socio-economic factor would ensure that each group has a mixture of different socio-economic class. Other potential confounders such as age, gender, race etc needs to be identified and controlled before they could be compared. Direct comparison without adjusting for confounder has serious implication as illustrated in Figures 2.6.

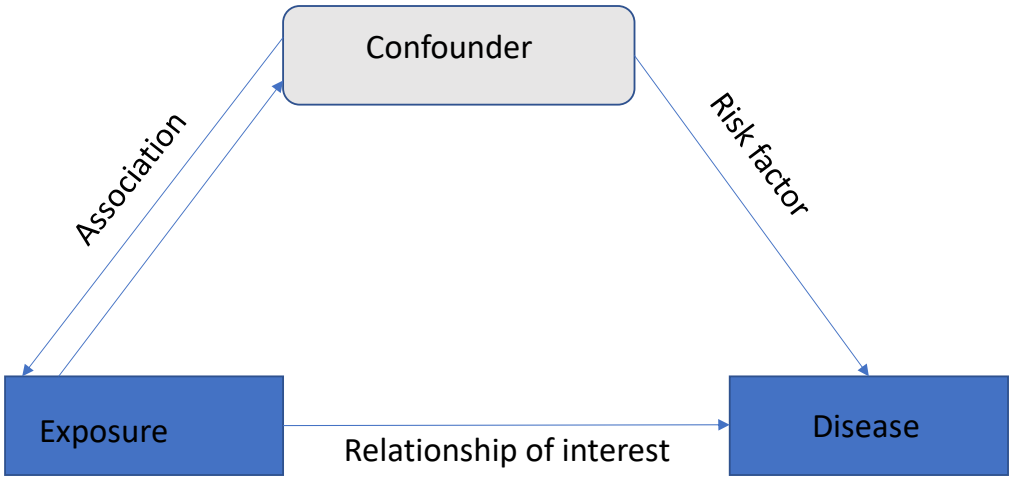


Figure 2.5: Confounder properties

A situation in which a factor contaminates the true effect of an exposure treatment variable under investigation is called confounding and the factor responsible is called a confounder (confounding variable). Before making any causal claims, such situation must be addressed by "controlling" or "adjusting" for the confounder. Randomized Control Trials naturally escape this through carefully designed experiment that ensures confounders (both observed and unobserved) are balanced across different treatment/ risk factors group. In contrast, Observational data are not balanced across the two treatment groups and thus, violates requirement number 2 listed earlier. Therefore, an investigator must identify the confounding factors/features and deploy techniques and assumptions to ensure balance between the groups before asserting any causal claims. Methods used to control or adjust for any confounder create stratum for each strong predictor of the outcome and ensure they are balanced across the different treatment groups. Balancing confounders mimics randomized experiments after which comparisons are made and subsequently causal claims inferred. Confounding can be addressed during study design phase by randomization, stratification, restriction, and matching [63].

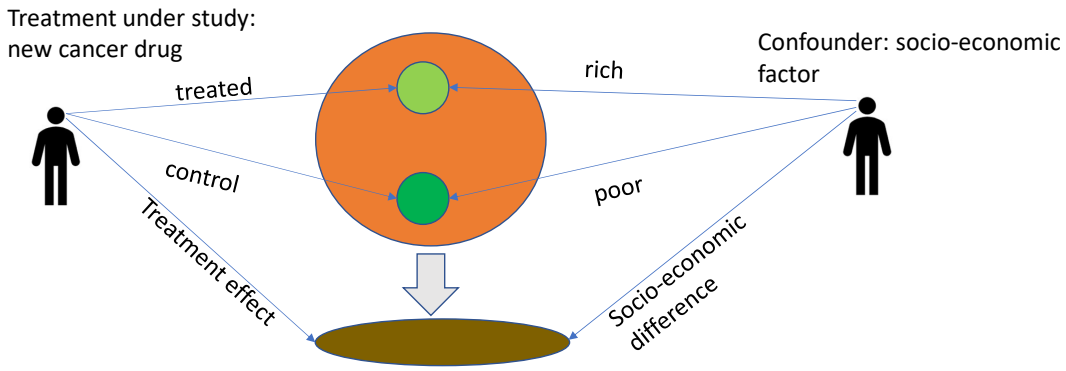


Figure 2.6: Effect of Confounding

2.2 Statistical Methods for Causality

We discuss *Randomization*, *Restriction*, *Stratification*, *Matching*, *Regression Adjustment*, *Logistics Regression*, and *Instrumental variables* as notable ways of controlling or adjusting for confounding [64],[63]. A study typically involves two stages namely: the design and the analysis phases. *Randomization*, *Restriction*, and *Matching* are deployed to control for confounders during the design phase. *Stratification*, and *Regression Adjustment* are used at the analysis phase [64]. Most research contributions in statistics and more recently ML/deep learning methods use at least one or combine two or more of these approaches in some way.

2.2.1 Randomization

Randomization, is an important property inherent in Experimental study and likely missing in observational data [24] which provides a guarantee that both observed and unobserved are only randomly different . Therefore, any method whose aim is to balance the distribution of the treated and the control is considered to be a matching method. Randomization is often deployed during the design phase to minimize confounding which involves random allocation of subjects into the study groups. This helps in addressing the problem of measured (variables or factors known by the investigator) and unmeasured (variables that are extraneous, that is, variable(s) outside the causal system being studied and are unknown by the investigator) confounders. The allocation is completely due to chance, which eliminates the confounding effect.

2.2.2 Restriction

Restriction– to adjust for a confounder by restriction, for example age, investigator choose to restrict the study to a particular study group based on the confounder. A study group could focus on subjects from 60 years and above. Excluding other age groups and analyzing the outcome of the exposure of interest with the chosen age group will give a true causal effect, having that the age confounder has been adjusted in the group [63]. Restriction is used often at the design phase. The downside of using restriction is that generalizing causal claims to a wider population for a homogeneous study group may be difficult.

2.2.3 Stratification

Stratification– the main idea is to adjust for confounding if any by producing groups based on the confounder factor so that within each stratum, the confounder does not vary, doing this means the confounding is controlled. The evaluation then follows, where the effect of the exposure of interest is estimated in each stratum of the confounder, the outcome obtained from each stratum is an adjusted outcome (an outcome obtained after the confounder is contained). It ensures that within each stratum, confounder cannot "mix" or "blurred" the true effect of the exposure of interest because it does not vary across the exposure - outcome. A statistical method like Mantel-Haenszel(M-H) can then be use to estimate the overall adjusted result according to groups [64]. If a difference exist between the crude (estimate before the adjustment) and the adjusted results, it shows that confounding exist,

otherwise, it does not. Each stratum becomes smaller in size as the number of strata becomes larger, which reduced the power to detect the association.

2.2.4 Matching

Questions like : *Does smoking causes cancer?*, *Does more sleep improves teenage academic performance?*, *Does daily Aspirin intake reduces risk of heart attack?* all intend to answer whether or not a treatment (intervention) causes an effect. Answering these kind of questions in the actual sense would require a time machine, and adherence to unethical procedures. For instance, to ascertain if smoking actually causes cancer, we would have to get a person who does not smoke to smoke for certain number of years (which is unethical) and measure the outcome. Then travel back using the time machine to the years where he was not smoking and observe to see if cancer is present or not. Estimating the Average Treatment Effect would mean repeating the procedure for many different persons [65]. It is beyond our capability to travel back in time to see the outcome of person with smoking and without. However, one can identify two persons with very similar features (characteristics) except that one smoked and the other did not. As a workaround, researchers in Statistics, Epidemiology, and psychology use ***"Matching Methods"***. The key idea in matching is to identify for every study participant from the treated group, a pair from the control group with the most similar characteristics and compare them to assess the difference in their treatment effect. These participants come from cohort study. The features matched on are the confounders identified such as age, gender, race, socio-economic factors etc which are controlled. We can use matching methods to approximate the effect of smoking and thus provide an answer to whether or not smoking causes cancer. In essence, when using observational data for treatment effect estimation, it is desirable to mimic as close as possible properties of randomization so as to ensure that the distribution of features from the treated and control are similar. Matching methods provide us with ways in which similarity or balance between the features of the two groups is achieved [21].
Causal inference could be viewed as a two stage process consisting: 1) Design, and 2) Outcome analysis. Matching falls in the first category. Matching methods have many advantages which include; complementing regression adjustment discussed in subsection 2.2.5, in fact, combining regression adjustment and propensity score matching defined in equation 2.7 produced better results [31]. Matching methods highlight areas where there is no sufficient overlap between the treated and the control distributions. Additionally, their diagnostic process is straightforward for assessing their performance and they also help reduce bias in treatment effect estimation [21]. The four main steps involved in matching are:

- *Closeness*– define a distance function that would be used to determine two subjects are similar or not.
- *Implement* a matching method that would use the distance function above
- *Evaluate* the quality of the matching
- *Analyze* the effect of the outcome and estimation based on the third point [21].

There are four distances primarily used in Matching [21]:

1. Exact:
$$D_{ij} = \begin{cases} 0 & if \quad X_i = X_j, \\ \infty & if \quad X_i \neq X_j \end{cases} \tag{2.5}$$
As shown in the exact matching, the distance between two subjects is zero for two similar subjects, and the more dissimilar they are, the larger their difference [21].

2. Mahalanobis :
$$D_{ij} = (X_i - X_j)^{'} \Sigma^{-1} (X_i - X_j) \tag{2.6}$$
Σ is the variance covariance matrix of X when computing Average Treatment on Treated (ATT) defined in equation 4.6. For Average Treatment Effect (ATE) defined in equation 3.2, Σ is a variance covariance matrix of X. Categorical variables are converted to series of binary indicators. This distance is known to work best with continuous variables [21].

3. Propensity Score [21] is a feature (covariate) balancing technique that mimics RCT. For each instance, the distance
$$D_{ij} = |e_i - e_j|, \tag{2.7}$$
where e_k is the propensity score. For subject k, treatment t, and features X, the propensity score is the conditional probability of a subject exposed to treatment given it's features. It is expressed as
$$e_k(X_k) = P(t_k = 1|X_k) \tag{2.8}$$
t_k is the treatment variable, X_k are features for subject k.

4. Linear Propensity Score [21]
$$D_{ij} = |logit(e_i) - logit(e_j)| \tag{2.9}$$

Linear Propensity Score is found to be particularly effective in reducing bias [66],[27], [67]. Mahalanobis is shown to do pretty well for fewer features (less than 8) [68], [69]. In high dimensional X however, Exact and Mahalanobis are difficult to work with because matching subjects along all the high dimensions is difficult, which could leave many subjects unmatched, thereby increasing the bias [25]. As a remedy, Coarse Exact Matching (CEM) (to be discussed later) can be used instead of matching exactly on the continuous features which is difficult, it matches on interval or range. It does so by converting continuous variables to a range of categories [70]. For instance, a continuous income variable could be converted to a range of income with a lower and upper bound. Any subject with income within that interval falls in that category. It is much easier to match on the interval than on a continuous value. The introduction of propensity score in 1983 was a great advancement, being that it summarizes the whole features into a single scalar [24], [28], [23], [71]. It is a balancing score, thus, if two subjects from treated and control, have the same propensity score, then their features are similar. Also, if two groups have the same propensity score, so their distribution [24]. Grouping them based on their propensity scores is equivalent to mimicking a pseudo randomized experiment at least with respect to the observed features. If the treatment assignment is ignorable given the features, then

it is also ignorable given the propensity score [24]. And if treatment assignment is ignorable given a propensity score, then the difference in mean between the treated and the control is an unbiased estimate of the treatment effect at the propensity score value [24].

Nearest Neighbor Matching (NNM) is believed to be the easiest and most effective. k:1 NN matching [22] in most cases is used in estimating ATT. k:1 implies that one treatment pair could be matched to many other control subjects. When k=1, a 1:1 matching finds one and exactly one closest control for each treated subject. Matching 1:1 discards a large number of controls with no treated pair. To determine which control is closest to a treated unit, a distance measure has to be defined. A distance or similarity measure has to be determined. If a propensity score is considered as the distance measure, then any two subjects whose propensity score difference is not more than the chosen threshold are considered close. The threshold is proposed to be around one quarter of the standard deviation of the logit of the propensity score [72]. The caliper (a threshold or a tolerance) is typically set at 0.2, or 0.25 as a maximum allowable cap for two subjects to be close [72]. If the distance measure is Mahalanobis, calculating the distance involves matrix inversion of the variance- covariance matrix, the computation becomes expensive and numerically unstable. The raw data is usually standardized by converting the variance-covariance matrix to an identity matrix using spectral decomposition [73]. Which transform the Mahalanobis distance to a Euclidean distance. The Euclidean distance computes the closeness as the square root of the sum of square difference between the n-components of the two vectors of the (treated and control) [73]. This type of matching is considered greedy with the order of the treated subject affecting the quality of the matching. The procedure is greedy because at each step in the process, the nearest neighbour subject from the control group is chosen even if the chosen control subject would be a better matching pair to a subsequent treated subject. This process is repeated without considering how other subjects have been paired. Therefore, it does not aim to optimize any criterion (that is, it does not try to minimize the overall distance). In contrast, *Optimal matching* [72] provides a solution, as it takes into account the entire set of matches when subjects are chosen by finding matched samples which produce the lowest average absolute distance across the matched pairs [13].

Propensity Score Matching Matching based on propensity score are used to match subject from the control group with subject from treated group having similar PS. Propensity Score Matching (PSM) works with the assumption that conditioned on some observed features, control subjects can be compared with the treated subjects as if they come from randomized experiment. For a successful matching, data should contain all relevant features that predict both the treatment and the outcome. The procedure in PSM involves the four steps listed earlier and also ensure common support is satisfied. Which is, subjects with similar features have positive probability of being both treated and untreated. Subsequently, the matching is done on the propensity scores [24].

Coarse Exact Matching (CEM)– In CEM, exact matching is required on all the covariates/ features, but at the initial stage, before all groups are formed all the covariates with continuous variables are discretized. Subjects from each treatment groups (treated and control) are placed in the appropriate (where the subject's feature falls in) strata. Only groups that have at least one treated and one control

subject in them are retained, while others are excluded from the analysis [74]. Then weights are assigned so that each subject from a stratum is assigned a 1 if treated and is assigned a score defined by the product between the ratio of the total number of control subjects in the strata to the total number of treated subjects in the strata and the ratio of the total number of the treated in the stratum where the subject under consideration comes to the total number of control subjects in the stratum where the subject comes from [74].

Sub-classification – Matching exactly on all the features between two subjects is computationally expensive, even more difficult with continuous features. Sub-classification form groups for each feature such that distribution of subjects in each subclass are very similar. How similar the distributions are is measured by the similarity distance such as propensity score. The propensity score is estimated using logistic regression. Treatment effect is estimated and analyzed within each subclass. It is important to identify relevant features for the sub-classification to avoid unnecessarily many sub-classes. For example, in a study to investigate the link between smoking and lung cancer, 5 features were used, each feature as a subclass. The study showed that adjusting just one subclass (based on age feature) reduce confounding bias by 90 %. [75]. [25] further showed 90 % of confounding bias is removed with 5 sub-classes of propensity with all features going into the propensity score computation. Sub-classes between 10 - 20 are reported to be feasible and appropriate as well [76].

Other types of matching are ratio matching [77],[78], and Nearest Neighbor(NN) matching with replacement [79]. The NN with replacement reduces bias by allowing multiple use of controls for different treated subjects. One control could be used as the closest subject to more than one treatment subjects.

Re-weighing Adjustment provides a way of directly using the propensity score as inverse weights to estimate ATE. This approach is called Inverse Propensity of Treatment Weighing (IPTW) [80], [81],[76]. The weight for each subject i depends on whether the subject is in the treated ($t = 1$) or control ($t = 0$) and is defined by

$$w_i = \frac{t_i}{\hat{e}_i} + \frac{1 - t_i}{1 - \hat{e}_i} \tag{2.10}$$

where $\hat{e}_i$ is the propensity score estimated for subject i. IPTW creates a pseudo-population that resembles experiment in which treatment is independent of measured confounders. The method assign smaller weights to over represented samples, whereas under represented samples are assigned larger weights to achieve an unbiased estimate of the treatment effect. [80], [81],[76]. Other weighing alternatives are : weighting by odds for estimating ATT [82] and kernel weighting [83] used mostly in econometrics. Generally, weighting methods are shown to have a large variance at the extreme propensity score (close to 0 or 1). The large variance is of less concern when the model is correctly specified.

Causal Inference using the Potential Outcome framework is a missing value problem and the main focus is to predict the missing counterfactuals. When outcomes are missing for any reason beyond the investigator's control, 2 ways are often used to adjust the parameter estimate for the covariates /features that may be related to the missing outcomes. One way is to model the relationships between the independent covariates and the outcomes, then use the model (i.e., a model that describes the

population of the responses) to input the missing outcomes [84]. The second way is to model the probabilities of the missingness (i.e., a model that describes the process by which data is filtered/selected) given the covariates/ features then include them as a weighted estimates [84]. Doubly Robust Estimator(DRE) [31] combines simultaneously these two methods, thus produce a consistent and unbiased estimate if either of the two models is completely specified. Model specification is a statistical modelling building process that involve choosing the right relationships and important features for the model to reflect the relevant aspect of the true data generating process [31], [84].

While these approaches have shown great success, we sometimes fail to measure or we simply cannot measure some features within the causal system. In this case, approaches that work well in the presence of unobserved confounders (exogenous variables) are adopted. One of such approach is the Regression Discontinuity Design(RDD) [37], [38], [39]. In RDD, a cut-off (threshold) value is set as a criterion based on a continuous variable chosen amongst the features. So that some subjects will have a value above the pre-defined threshold while the value of some subjects will fall below the mark. The subjects of interest are those whose marks are just above, and those just below the cut-off mark [39], [37]. Within this region, a quasi-experimental property is achieved being that subjects lying just in either side of the threshold are expected to be very similar and therefore exchangeable [37]. That is, reversing the treatments between the groups would not change the observed effect. The outcomes of these two groups could be compared. And an unbiased estimate of the Average Treatment Effect could be estimated by taking the difference between the mean of the two groups lying closely in either side of the mark [37].

2.2.5 Regression Adjustment

Using regression for causal inference often leads to bias estimate of the true treatment effect [68]. Some features are correlated with the outcome and are also not balanced amongst the two treatment groups. Separating between the effect due to the exposure of interest and those from the features is required to identify the true causal effect. Multiple linear regression analysis could be used to assess whether a confounding exists, since it allows us to estimate relationships between a given independent variable and an outcome, holding other variables constant. Providing a way of adjusting for (or accounting for) potential confounding features that have been included in the model [68]. Suppose a risk factor or an exposure (treatment) variable denoted by t (where t could be number of years in school), and an outcome y could be earnings which depends on t. A simple linear regression relating these two variables is

$$y = \alpha + \beta t \tag{2.11}$$

where β is the regression coefficient estimated quantifying the statistical relationship between the number of years in school (t) and earnings (y). If we are interested in whether a third feature (variable) u is a confounder, for example, u could be IQ level, age, or race, the three variables can be expressed in a multiple linear regression as

$$\hat{y} = \alpha + \beta t + \gamma u \tag{2.12}$$

Here, β is a coefficient that estimates the relationship between t and y, while γ is the coefficient that estimates the association between the potential confounder u

and the outcome y. Typically, confounding is assessed by comparing β in the simple linear regression in equation 2.11, and the β in the multiple regression in equation 2.12. As a rule of thumb, if there is more than 10% [85] changes from the simple linear regression, then u is said to be a confounder . Once identified, a multiple linear regression is used to estimate y adjusting for u.

2.2.6 Instrumental Variables

Under unmeasured confounders, Instrumental Variables (IVs) is a popular approach for estimating causal effect in non-experimental observable data [32],[33], [34], [35]. It is tailored towards determining the exogenous variation in treatments for causal inference estimation [86]. Empirical researchers are rarely equipped with all the necessary features to define a causal system [34]. In this case, IVs provide a way of determining exogenous variation (external causes) in treatment which could be used for causal effect estimation. IVs are statistical tools that could be applied to any data; experimental or observational that failed to satisfy the assumptions for an unbias estimate [86]. IVs use some features called instruments to determine the part of the data whose variation is not related to the unobserved confounders. Feature z is said to be an instrument with respect to treatment assignment t and outcome y if the following are satisfied:

1. *relevance* assumption: z is independent of all confounders (u) that can influence y. That is , z has a causal effect on y.

2. *exclusion* assumption: z is not independent of t so that z effects y only through t.

3. *exchangeability* assumption [87]: this provides that z does not share common causes with y. This assumption is also called *ignorability* [88], *independence* [89], [90], *no confounding* [36].

These matching approaches discussed earlier are amongst important contributions from statistical literature which many recent contributions in ML/deep learning models implement in some way. For instance, there are deep learning causal inference methods based on instrumental variables [91], and some based on propensity score [92]. Both our work and the matching methods discussed aimed to adjust for confounders through some matching / balancing techniques.

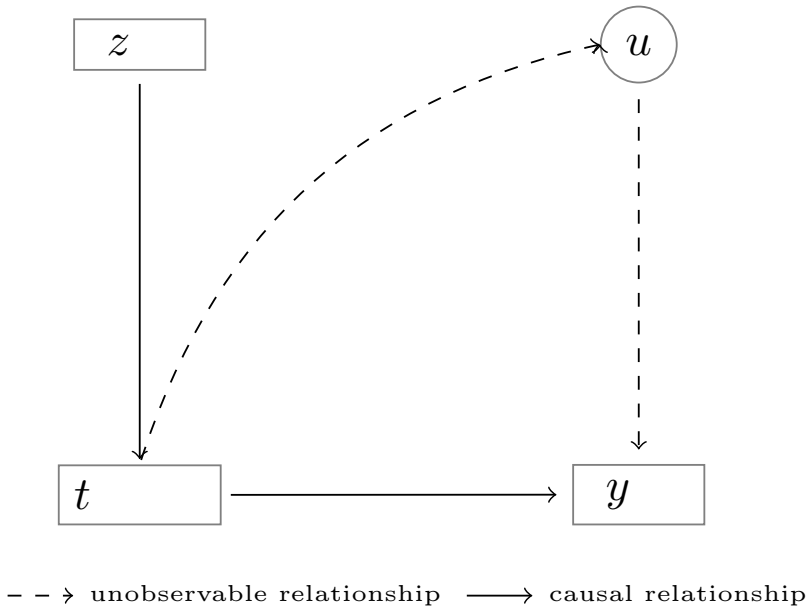


Figure 2.7: A valid Instrument for t

2.3 Paradigms for Causal Inference

There are two paradigms primarily used for studying causal questions. Such questions could either be answered through the use of Experimental or Observational study [93].

2.3.1 Experimental Study

Experimental paradigm is the most ideal for answering causal questions [1], [21],[94]. This paradigm is also called Experimental Study, Randomized Control Trials (RCTs), or A/B testing [1]. It is aimed at investigating the efficacy of a treatment. An investigator carries out a study under a controlled restrictive condition with individuals assigned to the exposure by chance [95]. The chance comes from randomly splitting of a sample into two or more split's. In the case of two split's, one of the split's is exposed to the intervention of interest, while the other split is not. Random split induces balance in both the observed and the unobserved features of the treated and the control groups; making it the most widely used paradigm in estimating Average Treatment Effect (ATE), and Average Treatment Effect on Treated (ATT). In fact, it is considered as a gold standard in estimating these parameters. It is a robust method mostly used to determine whether a cause- effect relationship exists.

Unconscious bias could be introduced to the study if the investigator knows who is and who is not receiving the intervention. To free RCTs of this bias during

allocation, the concept of *Blinding* is used; a blinding that keeps away only the participants from knowing who would be treated or not is called the *Single Blind*. In the case of *Double Blind*, both the investigator and the participant are not aware of who gets treated or not [96]. The randomization process could be achieved through block or stratified methods.

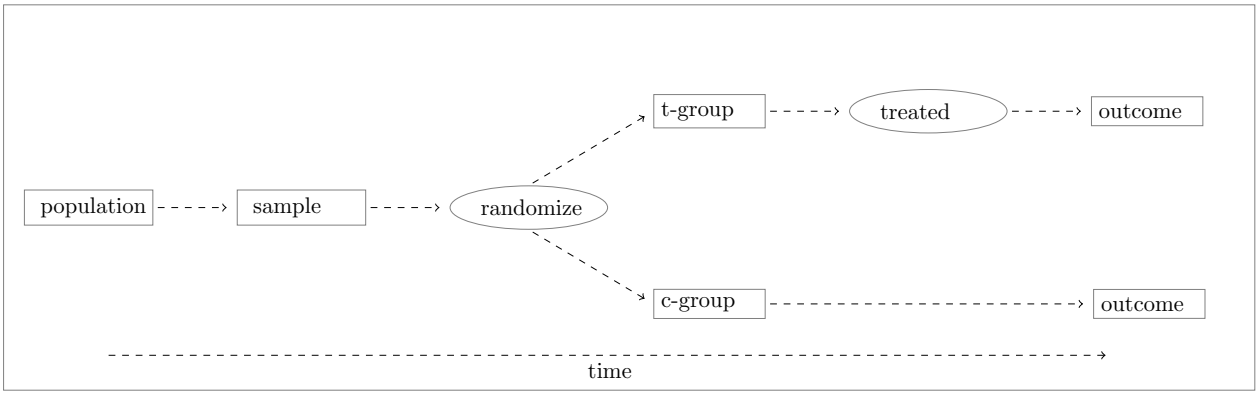


Figure 2.8: Randomized Control Trial Schema

Randomly splitting the sample cancelled out possible observed and unobserved confounding, thus eliminating bias; providing a basis for comparison. Once bias is removed, any observed difference between the exposed and the unexposed groups (split's) could be attributed to the effect of the intervention of interest. *Randomization* implies assigning patients or individuals to a treatment by chance. This paradigm has to it's credit the following advantages [1], [97], [98]:

- The investigator has the control to decide who gets exposed.
- High internal validity – that is, experiments are carefully designed in away that observed and un-observed confounders are taken care off well enough to conclude that the difference or effect observed are solely due to the intervention of interest. Thus, it provides a strong support for causal conclusions. In other words, minimized bias in the study due to the careful design gives the confidence that the difference in outcome could be attributed to the treatment [98].
- Creating balance between the observed and the unobserved confounders.
- Ability to assert causal relationship.

However, the fact that there are many questions of interest in which conducting experiments are unethical or completely impossible limit's the use of RCTs. Consider a situation where a researcher seeks to understand whether smoking causes cancer. It would be grossly unethical to expose a group of persons to smoking. More so, researcher would have to wait for a years to observe the effect [1], [99]. Similarly, if a research seeks to understand the effect of attending public or private schools on future accomplishment of persons, such investigation would take a long time to be achieved. In between these years, there could be cases of participant death or dropout. As discussed in the introductory section, the cost of running experiments is expensive and would naturally restricts the number of participants (sample size).

RCTs also suffer from the problem of limited external validity – that is, the power to generalize findings due to strict inclusion- exclusion criteria. *Inclusion- exclusion criteria* is the strict criteria about the eligibility of individuals in the recruitment (or selection) process [96]. These are typically carefully chosen to trade-off between a set of narrow and a very broad criteria. Very strict criteria would mean fewer number of samples, which would affect the generalizability of the findings. On the contrary, if the criteria are too broad, detecting the true effect would become more difficult [96]. With all these shortcomings, and the fact that many questions of interest are yet to be solved, Observational Study becomes a viable alternative. Observational study becomes even more plausible being that in recent years, data is generated at an unprecedented rate.

2.3.2 Observational Study

In contrast to RCTs, under the Observational Study paradigm, an investigator or a researcher only observes individuals and measure the outcomes without attempting to manipulate which individual is exposed to the intervention or not. Therefore, no randomization is used in selecting participants (treated or control). This necessitates the analysis to be carefully conducted to enhance the validity of the study [100]. The primary types of Observational Study are: *Cross-sectional*, *Case - Control*, and *Cohort study*. A Cohort study could be *Prospective* or *Retrospective*. A cohort study is said to be *Prospective* if the treatment is recorded before the occurrence of the outcome. If the treatment is recorded after the outcome, it is termed *Retrospective*.

In a *Case-Control* study, sample is formed by selecting the subject who showed the desired outcome (effect). The subjects with the effect of interest are called the *Cases* while the *Controls* are those without the outcomes [100]. Finding such controls is non trivial. This type of study does not answer *an intervention then effect* relationship because the effect precedes the cause. Whereas in causal inference, intervention comes first then the effect. Thus, it limit's the study to a discussion around whether a subject with effect of interest is more or less likely to have received the treatment compared to the control [100].

Cross-sectional study is a study in which both outcome and treatment are determined at the same time [100]. It provides a picture of the population status at one point in time. Lack of temporality accosted by the fact that both the treated (cause) and the outcome (effect) are recorded simultaneously prohibit's it's use for causal inference [100].

A *Cohort study* on the other hand provides the highest level of evidence as far as observational study is concern regarding the cause - effect relationship. The study sample is selected based on treatment of interest, and then later in the future, the outcome (effect) of the treatment (cause) is evaluated. A researcher does not allocate subjects into the groups. The temporality of the treatment - outcome relationship is established having selected the sample based on the treatment, and the outcome occurs at a later date (cause precedes effect). These controls are selected to match the already selected cases. This could be electronic records from existing databases. Nowadays, it does not cost much to obtain data, and could be on a large scale, which

allows subjects to be matched. It's drawback include: Identifying control could be difficult, treatment may be linked to a hidden confounder, blinding could be difficult, and lack of randomization [97]. Our work follows this alternative paradigm and more precisely the *Cohort study*. Unfortunately, data obtained through observational study posses a self selection bias or systematic bias (judgment from the researcher) [21], [22]. Meaning that after estimating the treated and the control outcomes, and the investigator decided to undo the treatment (by some imagination) given to the two groups and change the assignment mechanism so that those in the treated group become the control and vise-versa. Then administer treatment options to the new groups, the outcomes that would be observed would be different from the outcomes initially observed. Showing that the difference in outcomes initially observed are confounded due to the imbalance in the groups. If the groups were balanced, as obtained in experiments, the treatment outcome would not change also, the control outcome would remain the same. This situation is written compactly in equations 2.13 and 2.14 below.

$$\mathbb{E}[Y(1)|t = 1] \neq \mathbb{E}[Y(1)] \tag{2.13}$$

also,

$$\mathbb{E}[Y(0)|t = 0] \neq \mathbb{E}[Y(0)] \tag{2.14}$$

2.3.3 Causality and Causal Discovery

Causal questions could be tackled through forward reasoning or reverse forward reasoning. The former corresponds to Causality or Causal Inference, whereas the latter is referred to as Causal Discovery [93], [101]. In **Causal Discovery**, the focus is to identify causal factors responsible for the effect observed. Therefore, it answers questions of the form *Why*. Questions like *Why does inflation go down?*, *Why do poor people vote for Republican?* all seek to find factors responsible for the effect [101]. In other words, Causal Discovery finds *causes of effect*. **Causality** on the other hand involves estimating the effect caused by factors. Essentially, it is aimed at answering causal questions of the form *What if*. For instance, *What if a new drug is administered to the patient?*, *Does smoking causes cancer?* etc are valid forward reasoning questions. One could see from these questions that the treatment or intervention of interest is chosen before time. Answering these kinds of questions mean finding the *effect of causes* [101]. This thesis trains simple neural network methods and explores the impact of deploying more complex regularizations in estimating the effect of interventions. Thus, this work falls within the area of Causal Inference also known as Causality.

2.3.4 The Potential Outcome Framework

In statistics, Potential Outcomes (PO) aka Counterfactual approach is the most popular framework particularly to empirical researchers for learning causal inference with observational study [93], [102]. It posit's that the causal effect of a treatment of interest t on a subject i is a comparison between the outcome $Y_i(1)$ that would have been observed, had the subject received the treatment versus the outcome $Y_i(0)$ that would have been observed without the treatment (under the control). These two outcomes that would be observed depending on which treatment the subject

received are called the potential outcomes $(Y_i(0), Y_i(1))$. However, only one of the

$$Y_i(t) = t_i * Y_i(1) + (1 - t_i) * Y_i(0) \tag{2.15}$$

would be observed depending on which of the treatment options a subject actually received. That is, if subject i received a treatment $t_i = 1$, then $(1 - t_i)$ would be zero, and the potential outcome observed would be $Y_i(1)$. For similar reason, $Y_i(0)$ would be observed if $t_i = 0$. Comparing these quantities involves reasoning about counterfactuals, an approach traced down to the work of Neyman in 1923 [103], [102]. Many decades later, an extensive improvement of the work was carried out by Donald Rubin[6]. Rubin showed the counterfactual approach to causal inference is a missing data problem [6]. He therefore defined conditions under which it's inference is possible. He also developed a more general framework of the model especially with it's implication to observational study [6]. The framework is thus called the Neyman-Rubin model or the Potential Outcome model. In 1986, Holland made a holistic review of the model bringing out the consequences of the model assumptions [104]. His invaluable contributions made some researchers called it the Neyman-Rubin-Holland model reflecting the contributions of the trio. Significant to Rubin's contribution are two assumptions: The *ignorability* also known as *unconfoundedness*, and the *common support* assumptions [6] to be defined later.

An essential assumption needed to be satisfied in causal inference regardless of paradigm is the Stable Unit Treatment Value Assumption (SUTVA)[105]. This assumption embodied two conditions. The *No interference* and the *uniqueness*.

Assumptions in Causal Inference

Assumption 1 : Stable Unit Treatment Value Assumption (SUTVA) [105]. This comprises of:

(i) *no interference*: it asserts that the treatment received by any subject does not affect the treatment status of another.

(ii) *uniqueness*: it holds that there is only one version of any treatment option.
Assumption 2 : Strong Ignorability [24] comprises of:

(i) *independence (unconfoundedness)*:

$$\{Y(1), Y(0)\} \perp\!\!\!\perp \mathcal{T}|x \tag{2.16}$$

This implies that $P(Y(1)|x, \mathcal{T}) = P(Y(1)|x)$. Similarly, $P(Y(0)|x, \mathcal{T}) = P(Y(0)|x)$ The unconfoundedness assumption in equation 2.16 requires that given the pre-treatment covariates (features), the potential outcomes are independent of the biased treatment assignment $(\mathcal{T})$ which is our major concern . This assumption tries to assert equality in equations 2.13, and 2.14 discussed earlier.

(ii) *Common support (positivity)*: ensures

$$0 < P(\mathcal{T} = 1|x) < 1 \tag{2.17}$$

That is, the positivity or overlap assumption ensures similarity in the probability

distribution between the two groups. It would be unfair to compare distribution of males against females, or young against older people in a study investigating the effect of drugs for instance. Every subject has a chance of receiving each treatment which does depends on it's features only.

A major problem with the counterfactual approach is the fact that estimating causal effect for each subject i requires comparison of the subject's outcomes under the two possible treatment regimes. At a particular point in time, we can only observe one of the two outcomes but not both, a situation referred to as *Fundamental Problem of Causal Inference* [106], [104]. This problem limit's us from directly estimating the intervention effect called Individual Treatment Effect(ITE) for a particular unit . Under this framework, causal inference estimation can thus be viewed as a missing data problem [107]. Subsequently, the problem boils down to predicting the missing unobserved potential outcomes a.k.a counterfactual outcomes.

2.4 Machine Learning

According to [108], Artificial Intelligence or AI for short is an area of Computer Science that "involves using methods based on the intelligent behaviour of humans and other animals to solve complex problems." Our work is a machine learning (ML) method, a type of AI, see figure 2.9. We provide gentle yet brief introduction to ML to enable the reader to have the necessary background to understand this work. Although we wish to make this thesis a self contained document, this is by no means exhaustive as there are comprehensive ML resources available in hard copies and online.

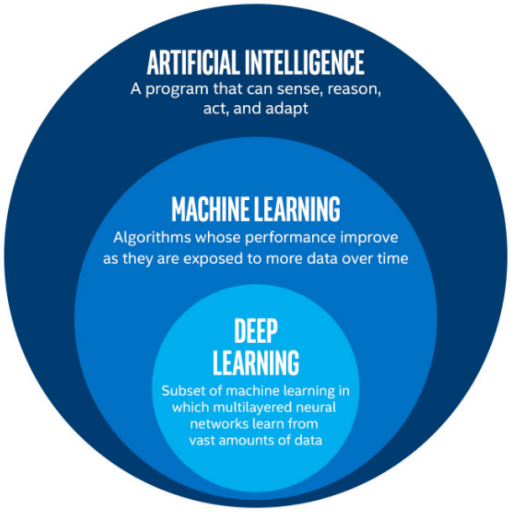


Figure 2.9: AI and ML
Source: [109]

We have witnessed data revolution in recent years, data is being generated in an unprecedented rate which comes in different varieties (from text, video, audio, and time series), and size. *"A computer program is said to learn from experience E*

with respect to some class of tasks T and performance measure P, if it's performance at tasks T, as measured by P, improves with experience E" [110]. ML's ability to learn from data automatically without being explicitly programmed is one of it's core advantages that puts it ahead of rule based systems, an idea widely celebrated . See figure 2.10.

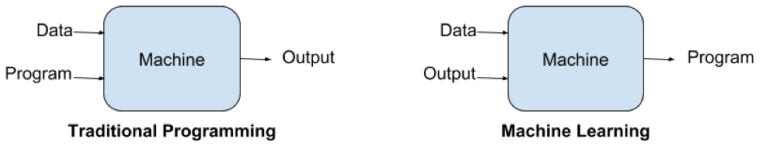


Figure 2.10: ML vs Traditional Programming
Source: [111]

In other words, *"machines are said to learn whenever it's structure, data or program changes based on it's inputs or response to an external information in such a manner it's performance improves in the future"*[112]. ML has 3 learning paradigms namely: Supervised, Unsupervised, and Reinforcement Learning [113]. In the ***Supervised*** learning paradigm, data used for the training the algorithm often called the training data is labelled. The labels are "answers" which guide the algorithm during the training. For n number of examples, the training data is of the form

$$\{(x_i, y_i)\}_{i=1}^{n} \tag{2.18}$$

Where x_i is a vector describing an object , and $y_i s$ are the categories (labels) or values corresponding to each x_i. A data instance could be the image of a dog or cat represented as a vector. The task in this case could be identifying whether an image is a cat or a dog. A supervised task could be to predict future house prices, in which case, y_i would be real values. If a task involves distinguishing between say cat and dog or more than two categories, such a task is known as classification. A classification task on two categories is called a binary classification and the categories are typically labelled as $\{0, 1\}$. The set up for supervised learning requires 2 sets of datasets: a training (source) data and the test (target) set.

Part of the training data, is set aside as a validation set. The validation set acts as a "mock test" to provide how well the model is learning the pattern. Validation also helps to adjust (or select) the best parameters during training . The second set of data is the test data which is never seen by the model during training, and is used to test how well the trained model can performed on the unseen data. The performance of the trained model on the test data gives a good indication about it's likely performance in the real world. A trained model should be able to perform the intended task quite well if the model actually learnt the underlying pattern (assuming the pattern exists) in the data. Failure to do so could mean that learning did not take place. The error (mistakes) a model commit's while making a prediction on an unseen test data measures how unwell the model does in generalizing. It is important to note that it is easy for the model to have a zero error on the training data by simply memorizing the labelled training set. Such model would likely not going to perform well on the test data, and we say that the model *overfit's*. If it does, then the model could not extrapolate outside it's training region. Conversely, a poor generalization could also mean that the model *underfit's*, meaning that it has only learnt some pattern but not all patterns in the data. This could be due to insufficient

data, stopping training early etc. There are other reasons that could hinder model generalization , this include if the data it'self does not hold the pattern intended for the task. A well trained model is expected to provide answers (predict the correct outcomes) when deployed in the real world. Popular algorithms for this task are Support Vector Machines (SVM), K-nearest neighbour(KNN), Logistic Regression, Random Forest and Deep Learning. Unfortunately, majority of real world data come unlabelled . This is one of the major drawbacks of supervised learning, as often requires all training data to be labelled. Labelling comes at a cost: time, and financial.

Unsupervised learning is another way of doing ML without data being labelled. The task may be to uncover hidden groups or to detremine how the distribution of the data is in space. Unsupervised learning are much harder task than supervised — how to ascertain if results are meaningful without labelled target is non trivial [114]. It is used in Clustering where data is grouped into clusters. Popular clustering algorithms include: K-means clustering, Hierarchical clustering etc. Unsupervised learning is used in anomaly detection such as: detecting fraudulent transactions, and in network intrusion [115]. Association mining is also an area where unsupervised learning is used. For example, market basket analysis is used to identify items in retail that are often bought together [115]. One popular algorithm for this is the Apriori algorithm. This paradigm is also used in data pre-processing to reduce feature dimensions–Latent Variable model [115].
Another type of learning paradigm is *Reinforcement Learning* where learning does not involve a teacher as is the case in supervised learning, and has a feedback mechanism contrary to unsupervised paradigm.

We would be discussing Neural Networks in more details next as it is central to our work.

2.4.1 Neural Networks

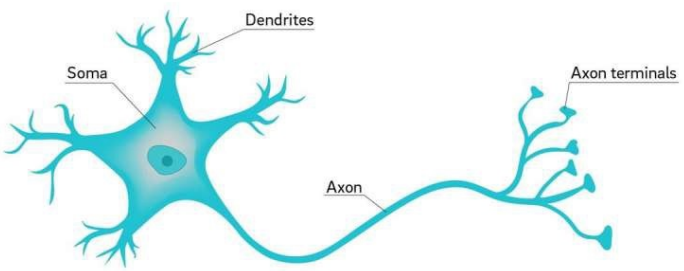


Figure 2.11: Biological Neuron
Source: [116]

For decades, neurologists have been studying the amazing functionalities of human brain which is said to contain billions of cells called neurons [116]. A typical biological neuron as shown in Figure 2.11 has a cell body called *Soma* which houses

basic cell functions and energy processing. *Dendrites* receive information from other neurons to the cell body, while the *Axon* sends information out of the cell body to other neurons through *Axon terminals* called *Synapses*. The Axon terminals of the originating neuron gets connected to a dendrites of the receiving neuron through a synapse. Transmitting information depends on the strength of a spike (signal).

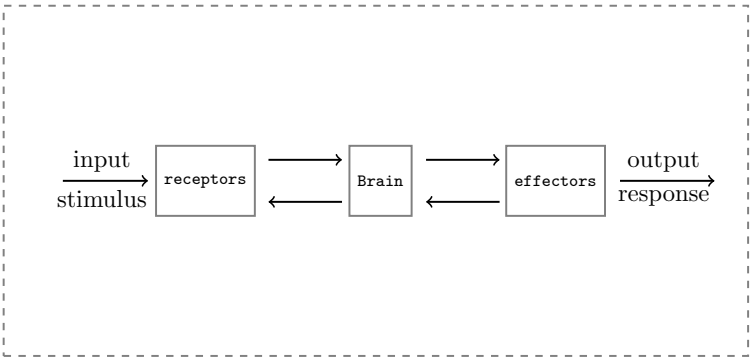


Figure 2.12: Human Nervous system

Information transmission from a neuron is achieved in 3 phases as shown in Figure 2.12. A stimulus generated from a neuron travels through the Axon. Receptors collect information from an environment say tongue while effectors generates interactions with the environment [117].

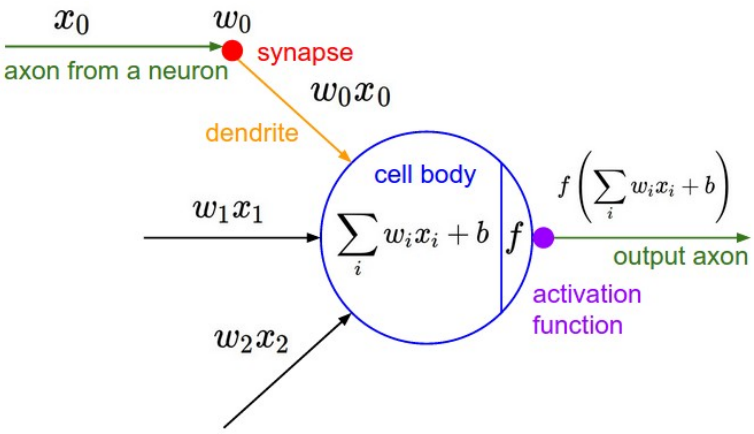


Figure 2.13: Artificial Neuron
Source: [118]

Artificial Neural Network (ANN) or Neural Network (NN) for short are seen as over simplified versions of biological neurons where inputs (output from other neurons) as shown in figure 2.13 are passed through dendrites with the help of a snap connector. In ANNs, inputs (signals) x with 3 features for instance are represented by a vector of features,

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \tag{2.19}$$

The input in equation 2.19 is passed into the network alongside weights (w_0, w_1, w_2) which are assigned to each as shown in figure 2.13. These weights could be written

as a row vector

$$w = \begin{bmatrix} w_0 & w_1 & w_2 \end{bmatrix} \tag{2.20}$$

Figure 2.13 shows the simplest NN architecture called perceptron [119], [117], [118]. This network has only an input and an output layer with no hidden layer. It is typically a binary classifier. The inputs are combined linearly with the connection weights to produce the output. Unlike biological neuron shown in Figure 2.12, an artificial neuron is a mathematical function which takes the inputs and weigh them separately, then sum all the weighted input. The weights are the "right amount" needed to produce the correct outcome [117]. The "right amount" is the proportion of the input (each feature has a corresponding w) needed to predict the correct label (or outcome).

An architecture with at least one hidden layer is called *Multi Layer Perceptron (MLP)*. More hidden layers could be added resulting in deeper networks.

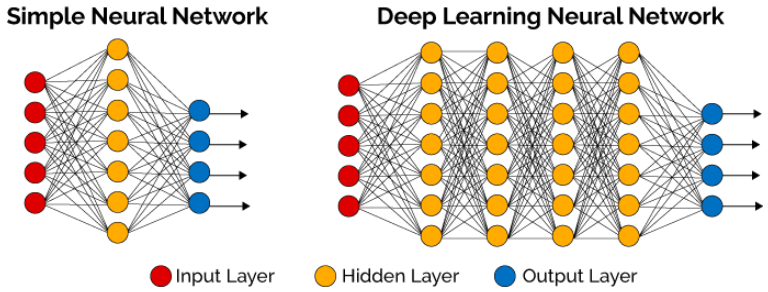


Figure 2.14: Neural Network with Hidden layers
Source: [118]

Other names for these types of network in Figure 2.14 are *Deep Neural Network*, *Deep learning*, and *Feedforward Neural Network* . The goal of Deep Neural Network is to approximate some functions. The word "Deep" relates to the number of layers (in yellow) between the input and the output layers. It ranges from simple to more complex architectures as shown in the figure. Increasing the number of neurons leads to a wider network [117], [118]. Any number of (hidden, or intermediary) neurons could be used starting from one. An architecture with large number of neurons and layers would results to a much wider and deeper network. In such networks, both the inputs and the connection weights are represented as matrices not vectors. The architectures in figure 2.14 are *Fully Connected* or *dense*, meaning each input is connected to every other subsequent hidden neuron. It is worth mentioning that there are architectures whose network is *sparse*. That is, the inputs are connected to some other neuron, [117], [118]. It follows obviously that complex networks would lead to complex computation. Consider a dense network with 1 hidden layer with 2 neurons and 3 input features x_1, x_2, x_3, in which every input node is connected to every other hidden node, the computation at the first node would be

$$node1 = w_{1,1}x_1 + w_{1,2}x_2 + w_{1,3}x_3 \tag{2.21}$$

While the second node would be

$$node2 = w_{2,1}x_1 + w_{2,2}x_2 + w_{2,3}x_3 \tag{2.22}$$

More compactly, the sum of both nodes is written as

$$output = \sum_{i=1}^{2} \sum_{j=1}^{3} w_{i,j} x_j \tag{2.23}$$

The sum of the linear output in equation 2.23 is then passed to an *Activation function*.

2.4.2 Activation Functions

These are important transformation functions useful in NN, with a linear activation function being the most simple and basic. We would focus on the non- linear ones which serve as a powerful tools for introducing non-linearity in neural networks. They are also called "squashing" functions for squeezing the real outcomes from the network to a smaller interval.

Sigmoid function — sigmoid is one of the most widely used non-linear activation functions. It has values between $(0, 1)$ [120], [121]and is expressed mathematically as:

$$sigm(x) = \frac{1}{1 + \exp(-x)} \tag{2.24}$$

It has an "S" like shape as shown in figure 2.15.

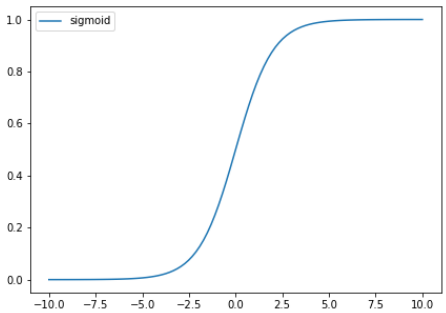


Figure 2.15: Sigmoid function

Tanh – (Hyperbolic Tangent) function: This is very similar to sigmoid, except that it's values are between $(-1, 1)$ [120], [121] see figure 2.16. It is defined by

$$tanh(x) = \frac{2}{(1 + \exp(-2x)) - 1} \tag{2.25}$$

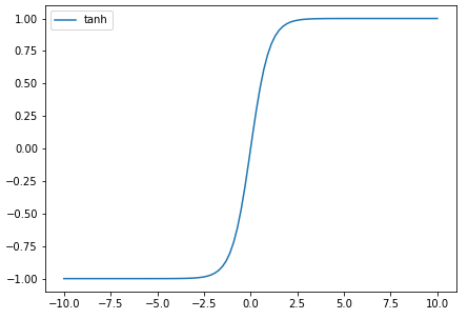


Figure 2.16: Tanh function

Rectified Linear Unit (ReLu) [120], [121] — is a very popular non-linear activation function. it's key property is that, it does not activate all neurons at the same time. Meaning, negative output of linear transformation would be deactivated. It's graph is shown in figure 2.17. It is given by

$$Relu(x) = max(0, x) \tag{2.26}$$

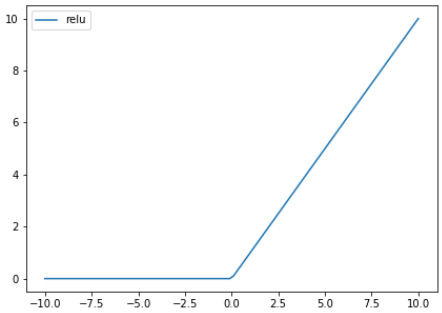


Figure 2.17: Relu function

Softmax —: The softmax function is a combination of many sigmoid functions. It is a generalization of sigmoid. Just like sigmoid, it's values are between $(0, 1)$, in addition, the outputs must sum to 1. Thus softmax outcomes are treated as probabilities [120], [121]. They are mostly used in a multi class problems. It is expressed mathematically by

$$softmax(x) = \left[\frac{\exp(x_i)}{\sum_j \exp(x_j)} \right] \tag{2.27}$$

Exponential Linear Unit (ELU)— This activation tends to converge faster, and has very good accuracy. It smoothes slowly until the output equals -α. It can produce negative values. One of it's drawbacks is that activations can diverge for $x > 0$. Elu is defined as

$$Elu(x) = \begin{cases} x & x > 0 \\ \alpha(\exp(x) - 1), & otherwise \end{cases} \tag{2.28}$$

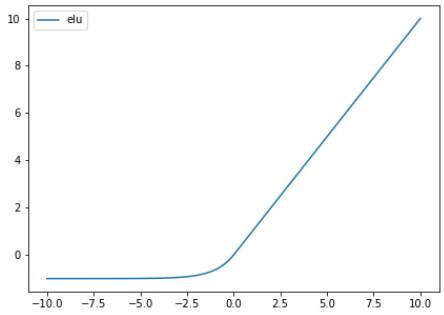


Figure 2.18: ELU

2.4.3 Gradient Descent

The goal of building neural networks is to find a set of "right" parameters that achieve the desired task (correct predictions). These parameters are the weights and biases. At the initial stage of learning, we do not know them, and the best practical step is to initialize them with random values (parameters). Looking for the "right" parameters means searching for them in the error space of possible solutions which is an optimization problem. The idea is like that of a man on a hill whose eyes are closed wanting to reach to the bottom. He takes a small step around moving in the direction of a higher decline. The gradient (slope) tells us the direction of the greatest increase. Which means, he needs to go opposite the direction of the gradient to get to the bottom. The small steps taken is called the *learning rate* in NN terminology.

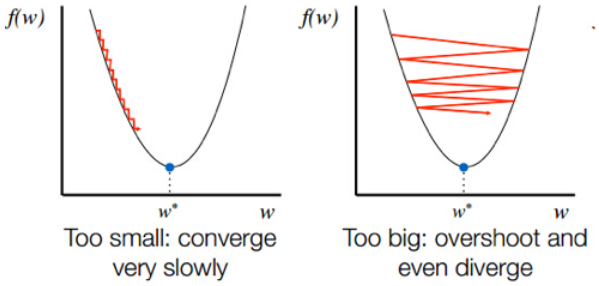


Figure 2.19: Choice of learning rate
Source: [122]

At every step taken, a new gradient is calculated corresponding to the new set of parameters. The gradient updates could be written mathematically as:
Repeat until convergence:

$$\textit{Repeat until convergence:} w_j \leftarrow w_j - \eta \frac{\partial f(w)}{\partial w_j} \tag{2.29}$$

$\frac{\partial f(.)}{\partial w_j}$ is a partial derivative symbol, w is the weight matrix (or a vector) and and η is the learning rate. The update in equation 2.29 subtract the new change from the current weight (w_j) at j and assign it the current weight at j. The process continues until convergence is achieved [117]. Popular variants of Gradient descent

algorithms are *Batch*, *Stochastic*, and *Mini-Batch* [123]. ***Batch gradient descent*** is an optimization in which the entire training data is used for parameter update. In ***Stochastic gradient descent***, parameters are updated on each single training point. While ***Mini-batch*** uses more than one data points and less than the entire training data. These are called the vanilla optimizers, and are not without problems [123]. One of the challenges using them is difficulty in choosing the right learning rate [123], [117]. A small choice could frustratingly slower the convergence, while a large learning rate could cause a noisy oscillation around the valley, hindering convergence. These cases are demonstrated in figure 2.19. The downsides of villa optimizers lead to the development of adaptive optimizers in which learning rate once initialized, the adaptive optimization algorithms keep updating it during training [124], [123]. Examples of these adaptive optimizers are : *Adam*, *Adadelta*, and *Adagrad* [124], [123]. The outcome produced by the parameters during the feedforward may be way off the correct values (or class). Meaning, mistakes or errors are made by choosing those parameters.

The idea of ***Back Propagation*** is a well celebrated concept in the history of NN [117]. This idea makes it possible to share the "blame" (errors) incurred during the forward pass by calculating the errors and propagating it back through the network. Each component of the network receives it's fraction of the "blame" proportional to it's contribution to the error during the forward pass. Weights (and biases) are adjusted in such a manner as to reduce the error in the next forward pass [117].

2.4.4 Loss functions:

After every forward pass, there is need a to measure how well the network does compared to the ground truth (the actual correct measurements) . The functions used for these are called *Loss Functions*. Typically, a large value from the loss function indicates a deviation from the correct value. These functions are chosen based on the nature of task (Regression, classification), type of ML algorithm, and ease of gradient computation among others. They are broadly classified based on Regression or classification losses.

Regression Loss Functions– A very popular loss function used in regression problems is the *Mean Square Error (MSE)* [125], [126] aka *Quadratic Loss* (L_2 loss) defined by

$$MSE = \frac{\sum_{i \in \mathcal{D}} (y_i - \hat{y}_i)^2}{|\mathcal{D}|} \tag{2.30}$$

Where $\mathcal{D}$, $|\mathcal{D}|$, y, and $\hat{y}$ are respectively the dataset, the number of instances in $\mathcal{D}$, the true and the predicted outcome. This loss penalizes heavily predictions that are far away from the correct values more than the small deviations. it's smooth property allows easy computation of derivatives , hence a good choice. Another Regression loss function is the *Mean Absolute Error (MAE)* aka $L1$ [127], [125] loss defined by

$$MAE = \frac{\sum_{i \in \mathcal{D}} |y_i - \hat{y}_i|}{|\mathcal{D}|} \tag{2.31}$$

It measures the average sum of absolute deviations between the predictions and the correct values. It is more robust to outliers, although computing gradients may

require complicated tools such linear programming.

A very popular loss function under classification losses is the *Cross-Entropy loss*. It measures the difference between two probability distributions [126], [125] and it is defined as:

$$Cross - Entropy = y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i) \tag{2.32}$$

This loss penalizes heavily wrong but confident predictions.

2.4.5 Representation Learning

Intuitively, *representation learning* or *feature learning* are techniques used in learning concepts or characteristics unique to classes to be classified. For instance, objects could be represented by their colors, shapes, size etc. Learning these concepts before classification or prediction tasks helps in easing the job of a learner. In text analytics for example, words are represented as vectors such as those seen in the popular Word2Vec etc [128]. Finding the appropriate representations of the data is shown to be very useful for machine learning tasks [129]. Deep learning almost naturally by it's nature exploit this concept in a way that each layer in the network captures a certain level of abstraction by mapping the input data to an inner representation of the data. These abstractions learned become more and more informative for the ML task which occurs typically at the last layer of the network. This means, initial layers in a deep network actually perform some kind of representation learning task with layers stacked in a hierarchy –typically non-linear [130]. These representations are detected automatically in deep learning in contrast to manually choosing them based on domain expert knowledge. One of the challenges in representation learning is lack of clear objective in training, unlike in a classification task for instance where the target is almost obvious (distinguish between categories). Thus, a good representation is the one that is able to discover and disentangle the underlying factors of variation with very little information loss about the data. It is also assumed to be expressive in a way that captures a huge number of input configurations, promotes feature re-use, and more abstraction at high level [129].

2.4.6 Maximum Mean Discrepancy

Statistical Distances are methods used in measuring the distance between two statistical objects, random variables, two probability distributions, samples etc. Methods used in measuring these objects are mostly not symmetric and hence not a metric. Some of these distances are called statistical divergence [131]. Examples of these distances are Maximum Mean Discrepancy (MMD) [132],[133], and Kullback–Leibler divergence [134]. One of the domain adaptation methods in this thesis use Maximum Mean Discrepancy (MMD) to align the distribution of the treated and the control. MMD measures the distance between the mean embedding of features used for analysing and comparing distributions to determine if they come from different distributions [133]. It is a direct method of measuring the distance between two distributions by avoiding density estimation. Instead, the mean of their embedding suffices [132]. Density estimation is not needed when the problem at hand is

very simple, and is always avoided due to the curse of dimensionality [132]. Given two distributions P and Q, over a set χ, MMD is defined by a transformation or representation map

$$\varphi : \mathcal{X} \to \mathcal{H} \tag{2.33}$$

Where $\mathcal{H}$ is referred to as Reproducing Kernel Hilbert Space (RKHS) [135]. Therefore, the MMD between two probability distributions P and Q is given as :

$$MMD(P, Q) = \|\mathbb{E}_{\mathcal{X}\sim P}[\varphi(\mathcal{X})] - \mathbb{E}_{\mathcal{Y}\sim P}[\varphi(\mathcal{Y})]\|_{\mathcal{H}} = \|\mu_P - \mu_Q\|_{\mathcal{H}} \tag{2.34}$$

where μ_P, and μ_Q are respectively the means of the distributions P, and Q.

2.5 Domain adaptation

Figure 2.20 shows the different categories and sub-categories of learning under non-stationary environment and our work falls under Domain Adaptation. We assume that the source data arrived at time $t1$, and the target data arrived at a subsequent time $t2$ with no further data expected.

2.5.1 Domain Adaptation Framework

The success of a machine learning classifier is measured by it's ability to generalize well on the unseen target set. When the independent and identically distributed (iid) assumption is violated, the model may not generalize well [136],[137]. A large difference between the distribution of the source and target sets would mean the classifier would be trained specific to the source data. This problem arises often in real world. In the machine learning literature, it is referred to as *Data Shift* or *Data Bias* [138].

But under what condition(s) would a model trained on the source domain different from the target generalizes well on the target? [139] propose a theory of learning from different domains. The theory bounds the target error by the sum of the source error and the divergence between the source and the target. However, the theory assumes the existence of a single model that performs well on both domains. The proposed divergence measure is a classifier induced and is denoted by $\mathcal{H}\Delta\mathcal{H}$, which allows the divergence from both domains be estimated from the unlabelled data. Formally, [139] formulate the learning framework as follows:

Let $\mathcal{X}$ be an input space. The function

$$g : \mathcal{X} \longrightarrow [0, 1] \tag{2.35}$$

is called a labelling function. Call a pair consisting of a distribution $\mathcal{M}$ on $\mathcal{X}$, and the labelling function g denoted as $\langle \mathcal{M}, g\rangle$ a domain. Denote $\langle \mathcal{M}_S, g_S\rangle$ and $\langle \mathcal{M}_T, g_T\rangle$ as the source and target domains respectively. Let the function

$$h : \mathcal{X} \longrightarrow \{0, 1\} \tag{2.36}$$

be a hypothesis. The probability of disagreement between the labelling function g and the hypothesis h under the source distribution $\mathcal{M}_S$ is defined as

$$\epsilon_S(h, g) = \mathbb{E}_{\mathcal{X}\sim\mathcal{M}_S}[|h(x) - g(x)|] \tag{2.37}$$

$\epsilon_S(h) = \epsilon_S(h, g_S)$ is the source error of the hypothesis, and $\hat{\epsilon}_S(h)$ the empirical source error. $\epsilon_T(h)$, and $\hat{\epsilon}_T(h)$ are parallel notations for the target domain. For a hypothesis h,

$$\epsilon_T(h) \leq \epsilon_S(h) + d_1(\mathcal{M}_S, \mathcal{M}_T) + min\{\mathbb{E}_{\mathcal{M}_S}[|g_S(x) - g_T(x)|], \mathbb{E}_{\mathcal{M}_T}[|g_S(x) - g_T(x)|]\} \tag{2.38}$$

Where g_S and g_T are the labelling functions for the source and the target domains.

2.5.2 Types of Distribution Shift

A Data shift is said to occur when the joint distribution of the inputs and labels changed between the source and the target domains. For example, in spam filtering application, data at the source would differ from the target data, because subsequent user activity could change the initially trained system [140]. The nature of the shift depends on where it occurs which gives rise to three main shifts namely: *Prior*, *Covariate*, and *Concept*.

Let $P(x, y)$ be a joint probability function, with x as variable describing the object under consideration, and y a class label variable associated with each x. The decomposition of the joint distribution gives $P(x, y) = P(x|y) * P(y)$, $P(x|y)$ a likelihood function, and $P(y)$ the prior probability.
A *Prior Shift* occurs when $P_S(x|y) = P_T(x|y)$, and $P_S(y) \neq P_T(y)$.

Where P_S, and P_T are the probability distributions of the source and target domains respectively. Prior shift could arise in real world in fault detection systems. If a fault detection classifier is trained with previous dataset before an effective maintenance policy is put in place, the prior distribution $P(y)$ of the test data would be different [140] . Similar shifts could arise in oil spills detection systems [141]. *Covariate Shift* (or *Sample Selection bias*), is the most common type of shift, the shift affects the marginal probability distributions. The assumption is that $P_S(y|x) = P_T(y|x)$, and $P_S(x) \neq P_T(x)$ [142], [143]. That is, the posterior distributions are equivalent while the marginals differ. Favouring some events more than others in the form of sample selection bias is one of the causes of this shift [142]. Missing data could also cause covariate shift. This could occur due to subject drop out in a survey, device failure etc. Other cases of covariate shift could be found in object recognition tasks. For instance, in classification of cat or dog where certain species of these classes are omitted in the source domain but present in the target domain. In face recognition tasks for example, such problem could arise if the source data contains predominantly younger faces and the target data has much proportion of the older faces. The domain adaptation methods implemented in this thesis are meant to address the covariate shift problem.

In *Concept Shift* however, the marginal distributions ($P_S(x)$ and $P_T(x)$) are equivalent, but their posterior distributions differ ($P_S(y|x) \neq P_T(y|x)$) [142]. The association between the input and labels change making it the most challenging among other types of shift. Problems of this nature do arise where the concept changes over time, thus the earlier model could not properly predict the new outcome distribution. For instance, in fraud detection, new policy, technology could change the posterior distribution overtime. Spam filtering applications could also be a source. An initial classifier trained for a particular user may fail to filter correctly due to new activities of the user. What was Spam could be not Spam to the user

or vice versa.

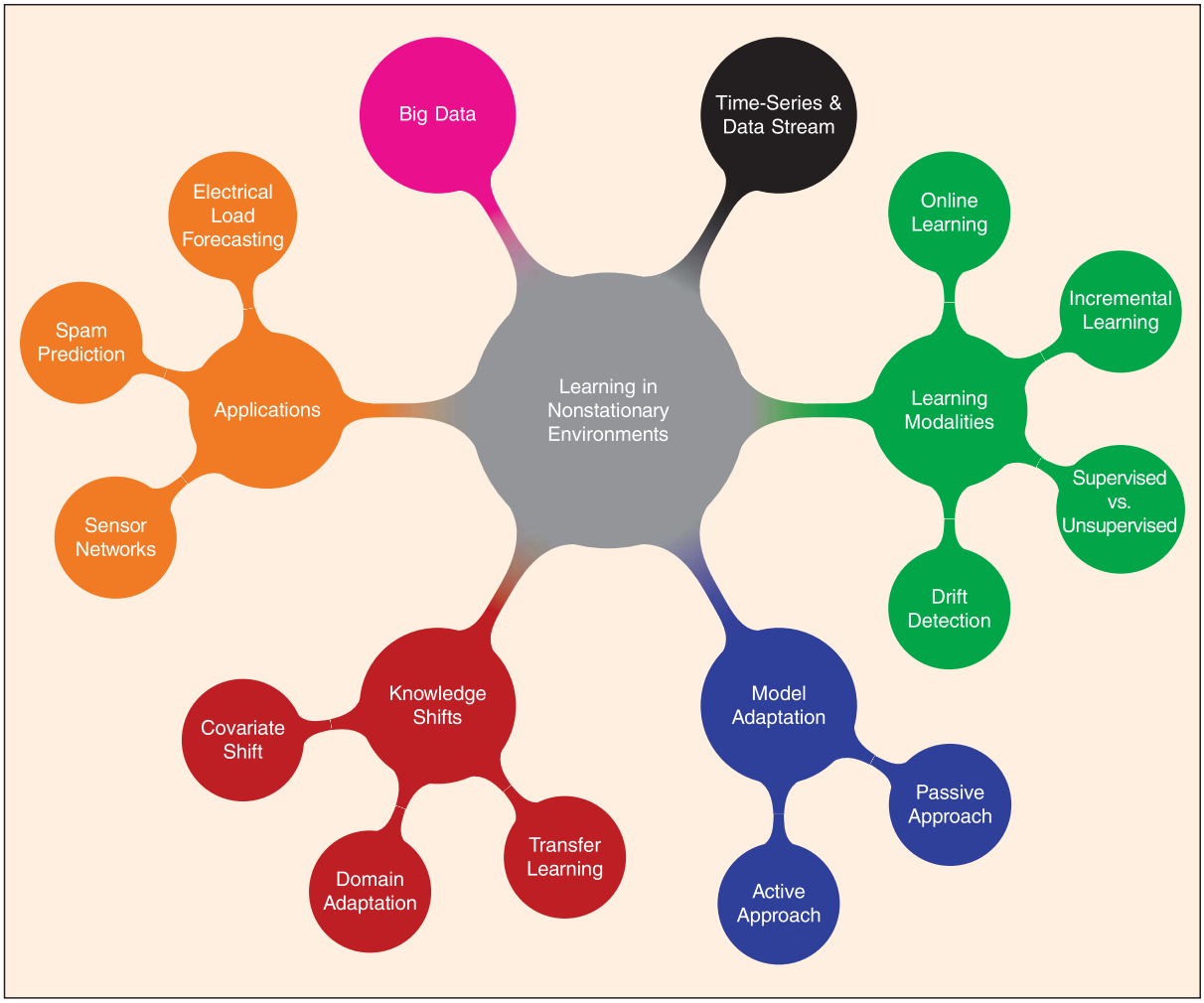


FIGURE 1 Mindmap of concept drift describing the connections the field has with different areas within machine learning and applications where concept drift can be found.

Figure 2.20: Non-Stationary Environment
Source: [144]

2.5.3 Un-supervised Deep Learning Domain Adaptation

Domain Adaptation is a term used to refer to techniques for mitigating data shift. Domain adaptation techniques could be supervised or un-supervised. All our methods in this thesis involve unsupervised domain adaptation and thus, we would restrict our review to relevant unsupervised deep learning methods. An unsupervised domain adaptation method CORrelation ALignment (CORAL) proposed in [145] aligns the second order statistic of the original source and target input features using a linear transformation matrix. The matrix is obtained through whitening and re-coloring transformation of the target input features and a small penalty is added to the elements in the diagonal covariance matrix, explicitly making the matrix a full rank. The matrix then multiplies the original source features by it's inverse square root. This type of transformation method is considered faster and more stable compared to the Singular Value Decomposition (SVD). The method avoids projecting

the input features into a lower dimensional manifold. [53] extends CORAL [145] by proposing an end to end convolutional network with a differentiable CORAL loss for object recognition tasks. Unlike the linear approach reported in [145] where the covariance matrices are computed from the original input features, non-linear activations after the convolutional layers are used in computing the covariances of the source and the target. The joint optimization is performed by minimizing the classification loss and the deep CORAL loss. The distance between the covariance matrices in the deep CORAL loss is computed with Frobenius distance and the loss is added as an additional regularization term. Minimizing the classification loss only would lead to over-fitting, whereas minimizing CORAL loss alone could lead to degenerated features. A hyper-parameter is chosen to trade-off between the two extremes.

[52] propose the use of Riemann metric called Geodesic distance for computing deep CORAL loss reported in [53]. The authors argued that covariance matrices are symmetric positive definite (SPD) matrices which belong to Riemann manifold with non- zero curvature. Therefore, using Euclidean metric in a Riemann manifold would be sub-optimal. The paper presents empirical results against the results in [53] on object recognition task. The authors analyze and draw a connection between correlation alignment and entropy minimization approaches which shows that at it's optimum, correlation alignment attains the minimum of the sum of the cross-entropy on the source and the entropy on the target data.

[54] learn invariant features by bringing together representation learning and domain adaptation. Invariant features are those insensitive to a specific domain. A classifier trained on these features is shown to adapt to the problem of shift between the distribution of the source and target. The architecture embodied a feature extractor together with a label predictor to form a standard feedforward network which is linked to a domain discriminator through a special layer called a Gradient Reversal Layer (GRL). The GRL has no parameters associated with it and acts like an identity transformation during the forward pass by transmitting the features extracted to the domain classifier. During the backward propagation, it takes the gradient from the subsequent level and reverses it (by multiplying it with a -1) before passing to the preceding layer (feature extraction phase). This way, the domain classifier struggles to distinguish between the representations from the source and those from the target. This training is adversarial to the domain classifier. The entire training is minimax optimization that jointly minimizes the label prediction loss and maximizing the domain classifier loss. A hyperparameter is chosen to trade-off between standard label prediction and the degree of the domain adaptation.

Imposing feature in-variance could discard potentially informed features with high discriminating power for the main task [55]. An explicit modeling for the source and the target samples in a separate streams is proposed deep learning architecture [55]. Each stream learn separate representations peculiar to the sample. The weights of the streams are not shared, nonetheless, they are related through weight regularizer, an exponential L2 norm, which ensures the weights from the separate streams are not too far away from each other. The representations for the two streams are align by an additional regularizer based on Maximum Mean discrepancy (MMD) distance. A joint training is performed for the loss functions from the source

and the target streams alongside with the two penalties. Two hyper-parameters are chosen to control the degree of the two regularizations. The approach is evaluated on object recognition task.

[146] assert that Subspace alignment alone fail to align the source and the target distributions, and thus develop a method that not only aligns the basis of the subspaces but also align the data distribution of the subpaces. [147] projects the source and target features into a common spherical distribution (whitening), and in addition, introduces a loss function that combines entropy and consistency loss to avoid tuning many hyper parameters.

2.6 Machine Learning methods for Causal Inference

Different ML techniques are applied in various ways to infer causal parameters. Within tree based methods, [16] propose a similarity search model to compute the Individual Treatment Effect for Amyotrophic lateral scelerosis (ALS) patients, a complex disease in personalized treatment. Causal Forest [148] balances features through tree-based approach, an approach closely related to Random Forest [149]. Instead of Individual Causal Effect, a subgroup level effect at the tree leaf is estimated. It assumes unconfoundedness with an implicit use of propensity score. The features in the training data are splited recursively to achieve leafs that are small enough with very similar features. So that each leaf is pseudo random and are comparable, therefore, yield an-unbiased causal effect. The number of causal trees generated by Causal Forest (ensemble of causal trees) are build from a small random sub-sample of the training data and their outcomes are aggregated to reduce variance in the overall estimation .

[150] is a deep learning architecture for counterfactual prediction similar to TARNet [45] with extended arbitrary number of heads beyond two. It stimulates pseudo randomized data in a mini batch during training by comparing each sample from the mini batch with it's closest counterfactual nearest neighbour having opposite treatment from the training data using propensity scores (the probability of being treated given the features).

A deep learning method for ITE estimation that merge preserving local similarity between subjects in the latent space and balancing of the latent space distributions of the treated and the control groups is proposed in [51]. The method exploit's the advantage of nearest neighbour methods whose key principle solely relies on similarity between neighbourhood. It also incorporates the advantage of balancing the latent distributions of the treated and the control offered by deep representation learning methods. The similarities of subjects is preserved from the feature to the representation/latent space. The local similarity information is preserved using a customized function called Position-Dependent Deep Metric (PDDM). On the other hand, distribution balancing is achieved through a function called Middle-point Distance Minimization (MPDM), both functions are penalties added to the main loss. The PDDM uses 3 hard training example pairs chosen based on propensity scores. One pair lies in the intermediate of the region space where both the treated and the control subjects are mixed. The other 2 pairs are at the 2 extremes that

is, one pair is chosen at a position with predominantly treated subjects and the other dominated by the control group. The MPDM penalty however uses 2 middle points to approximate the centers of the treated and control groups as region where a mixture of the treated and the control representations are. The distance between the 2 middle points is minimized so that the margin area is gradually made closer to the intermediate center region.

[41] propose two balancing methods for counterfactual prediction using observational data by formulating causal inference as domain adaptation problem, precisely a classical case of co-variate shift. The first method called Balancing Linear Regression (BLR) is based on feature selection by re-weighting features that are similar across the treated and the control samples. However, doing this alone would discard dissimilar features that have high discriminating power for the main task. Thus, it uses a sparse re-weighting approach to trade off between similarity/balance features and discriminating power. Features that are most dissimilar are assigned a small weight. Whereas the deep learning method called balancing Neural Network (BNN) [41] is a feedforward deep learning architecture with fully connected layers which learnt representations for counterfactual prediction satisfying a customize objective function. The task is to minimize a mean square error (which minimizes the error between the factual outcome and the predicted observed outcome) together with two balancing penalties. One penalty minimizes the difference between the distribution of the treated and the control using a discrepancy distance proposed by [151]. The second regularizer encourages a low error in the prediction of the counterfactuals using the closest factuals with opposite treatment as it's proxy counterfactual. An improvement to [41] where theoretical analysis for the family of the balanced features and representations methods is reported in [45] . The authors also developed a generalization bound similar to domain adaptation bound of [152], [139] but for counterfactual prediction using observational data. The generalization bound is found to be the sum of the regression/ classification error of that representation and the distance between the distribution of the representations of the treated and the control. The feedforward architecture is modified to have two heads for the treated and control regressors although, the representations used by the two heads come from a single feature generator . The bounds are explicitly modelled for two inverse probability metrics namely: Wassertein and Maximum Mean discrepancy distances. Which subsequently lead to TARNET a model with no balancing penalty, CFR-WASS a model with Wassertain [153] balancing penalty, and CFR-MMD a model with squared linear MMD balancing penalty.

Learning invariant features often reduce the prediction power of the classifier, [154] develop a GAN framework for balancing the representations (between the treated and control) using an encoder network to preserve the mutual information between the original input and the representations in the latent space. [50] deployed Generative Adversarial Networks (GANs) architecture for ITE estimation. The overall architecture has two GANs, one for generating the counterfactuals while the other for the ITE generator. The counterfactual generator works as an imputation method which generates proxy counterfactuals to fill in the missing counterfactuals corresponding factual outcome. Thus, a vector of combined factuals and the generated counterfactuals is passed to the discriminator of the network whose job is to distinguish which part of the vector comes from the generator. The ITE GAN receives the complete dataset from the counterfactual block. The generator of

the ITE block uses the features and some random noise vector to generate potential outcomes, then the discriminator of the ITE generator tries to distinguish outcomes that come from the ITE generator and those from the complete dataset it received from the counterfactual block. The entire learning involve minimizing as well as maximizing some objective functions. A major weakness of this approach is that it's network architecture is complex which requires learning more hyperparameters.

[155] proposed an ITE estimator without asserting the assumption of strong ignorability. Meaning, it assumed the presence of un-measured confounders. The approach discovers the hidden confounders using proxy latent variables by adjusting for the confounders through the back-door criteria [156] (a criteria used to identify variable(s) which when conditioned on it (them) would block any non-causal path from contaminating the causal effect of the variables under investigation) implemented using a Variational AutoEncoder (VAE) network with some modifications. The modifications comprises of an inference network with two heads which takes in features as input and produced compressed representations for the treated and control, while the model network takes the output of the inference network and tries to predict the treated and the control outcomes. Unlike conditioning on the input features as proposed in TARNET, the output is conditioned on the latent representations instead.

A Bayesian non-parametric tree based method [157] approximates an inference function by summing up many trees. The model fit is constrained using a prior penalty to keep the effect of each individual tree small. This weakens the individual trees making each tree only explains a small yet different part of the overall function sum approximated. The function fitting is a Bayesian iterative procedure for the additive models called a Bayesian backfitting Markov Chain Monte Carlo (MCMC) (a random sampling technique for high dimensional probability distribution) on a fixed number of trees providing both point wise and interval estimation.

A method proposed in [49] combines regression method and a propensity score modelled as multi-task problem using deep learning architecture. At the initial layers, common (invariant) features are learnt for the treated and the control groups. The learnt features are subsequently splitted to form a two head streams/path in the architecture, where subsequent layers learn peculiar features for each group from the learnt representations in the previous layers. A separate logistic regression network is used to obtain propensity score (probabilities of being treated given the features) for each training example, and are used as drop-out regularization probabilities for the two streams of the network. Each training example uses a different propensity drop-out. Higher probabilities are assigned to examples with features that belong to a poor treatment overlap in the feature space.

The idea of ITE estimation proposed in [154] involves a Generative Adversarial Networks architecture. The encoder network of the architecture is constrained to learn balance representations for the treated and control while preserving discriminative information for the prediction task. A predictor is specified to estimate and maximize the mutual information between the learnt representations and the features (covariates). The encoder attempts to fool the discriminator network by minimizing the difference between the distribution of the treated and the control representations. A treatment outcome estimator in the network takes in the representations from the constrained encoder and output the treatment outcomes. All the three components are learnt jointly for a robust ITE estimation.

Chapter 3

Causality Methods

As mentioned earlier, developed ideas pioneered in domain adaptation for causality. Our work is motivated by [41] in which they proposed formulating causal inference as domain adaptation problem.

Our choice of these methods was on the account of the diverse approaches to domain adaptation. For instance, both the adversarial and correlation alignment models learn invariant features. Invariant features are features common to both the source and the target domain. However, learning invariant features is only one way of doing domain adaptation. In contrast, the parallel two streams method fall into a different class of orchestrating domain adaptation. At first, we carefully build simple models with no DA regularization which we called our baseline models, and then incorporate the domain adaptation regularizations pioneered in DA into the causality setup. Meaning that we extend the simple baselines to include complex DA components. It is part of our interest to bring as many as possible, concepts and approaches found in domain adaptation methods into causal inference. These methods are modified to suit causal inference setup, which are in-turn evaluated on 4 well known causality benchmarks. Initially, as stated earlier, we built simple NN models as baselines around these models, then we incorporate the domain adaptation components of these models. But we try to keep the core idea of each method to enable a fair discussion relating to the research objectives. The models were developed so that assigning zero to the domain adaptation coefficient would reduce each model to its baseline model. Each of these methods is discussed in a separate subsection.

In this chapter, we describe each of the four(4) methods implemented; each of which falls under one of the following approaches: ***correlation alignment*** [52], [53], ***Adversarial training*** [54], and ***parallel two streams*** architecture [55]. There are two methods under correlation alignment approach; one that employs *Euclidean distance*, and the other which is based on *Non- Euclidean distance*. We would be referring the correlation alignment method with no DA (Neither Euclidean nor Geodesic) as Counterfactual Regression (CR). Both the Euclidean and the geodesic correlation alignment methods collapsed to CR model when DA component is set to zero. The CR model is the baseline model for the two correlation alignment methods.

3.1 Problem Formulation

We follow the causal problem setting of Neyman and Rubin known as the Potential Outcome framework [6]. Under this set up, we assume a binary treatment indicator $t \in \{0, 1\}$. The treatment indicator t is a feature that tells whether an individual received a treatment or not. If an individual i received a treatment of interest, then $t_i = 1$, and $t_i = 0$ if i received no treatment (or a placebo). To be consistent with our terminology, from now henceforth, we would stick to using the term treatment to refer to other similar terms like intervention, policy etc . Let $x \in \chi$ represent pre-treatment features describing (other than the treatment feature) an individual in a given context. A context could be hospital, and in such case, an individual would be a patient. x_i would be a vector of characteristics describing patient i. In a business context, such an individual would be akin to a customer. There are two possible outcomes called the potential outcomes that would be observed depending on the treatment an individual received. We denote $Y_1(x)$, and $Y_0(x)$ to be the potential outcomes respectively for an individual characterized by x having being treated or not treated (control). At a particular point in time, an individual could only receive one of the treatment options but not both. Therefore, we could only observe one of the two outcomes unless off-course one decides to simulate both outcomes. In reality, both outcomes could not be obtained at a time. This is a challenge, and it is prominently known as the fundamental problem of causal inference [104]. However, if SUTVA [105] see section 2.3.4 and Strong Ignorability [24] are satisfied, the potential outcome framework [6] posits that the Individualized Treatment Effect (ITE) could be computed by comparing these two potential outcomes as

$$ITE(x) = Y_1(x) - Y_0(x) \tag{3.1}$$

In addition, the Average Treatment Effect defined by

$$ATE = \mathbb{E}[Y_1(x_i) - Y_0(x_i)] \tag{3.2}$$

could be estimated. Where the $\mathbb{E}[.|.]$ denotes the mathematical expectation conditioned on the features of each individual i .

Asserting these assumptions is a standard practice, and it is not uncommon for empirical researchers to use the Potential Outcome framework in causal inference. Lack of access to one of the two outcomes portrays causal inference with Potential Outcomes as a missing value problem knowing that one of the two quantities required for computing ITE would always be unknown. This means we can not directly estimate ITE from data. If an individual i received a treatment ($t_i = 1$) with an outcome ($Y_1(x_i)$) observed, then our dataset contains only one of the two outcome pair needed to compute it's $ITE(x_i)$. For this reason, the treatment effect estimation using the Potential Outcome framework boils down to estimating the unknown (missing) outcome pair called the counterfactual outcome. There should be some ways of approximating this missing counterfactual pair for an alternative treatment $t_i = 0$. An attempt is often made using regressors to estimate $\hat{y(x)}_1$ and $\hat{y(x)}_0$ as an approximations to $Y_1(x)$, and $Y_0(x)$ respectively for all t and x. This is exactly what our models attempt to do. Estimating the missing outcome intends to answer the counterfactual question: *what would be the outcome of the individual i if he had received an alternative treatment $t_i = 0$?* Our models are counterfactual

models as they attempt to predict counterfactual outcomes, and thus, the Potential Outcome framework is also known as the Counterfactual framework. To train the neural network models, we use the factual data D^F at our disposal. Each individual i in D^F is a 3-tuple $(x_i, t_i, Y_{t_i}(x_i))$. x_i describes the characteristics of an individual before receiving a treatment, t_i is it's treatment indicator (whether i received a treatment or not), and $Y(x_i)$ is the actual outcome observed under t_i. All our models are built to estimate reliably the missing counterfactual pairs. Once they are obtained, computing causal effect is reduced to a comparison between these two outcomes. That is, we subtract the difference between the factual outcome and the predicted counterfactual outcome of each individual under the two treatment regimes.
Predicting the counterfactuals means we would be making inference over a counterfactual data set D^{CF}, a 2-tuple $(x_i, 1 - t_i)$ for each individual i.

We assume that the empirical factual distribution $\hat{D}^F$ approximates D^F (population factual distribution) and the empirical counterfactual distribution $\hat{D^{CF}}$ approximates D^{CF} (the population counterfactual distribution). It is often the case that $\hat{D^{CF}}$ is not the same as $\hat{D}^F$. This means, we are making inference over a potentially different distribution than the factual distribution. In machine learning literature, learning from one distribution, and making inference over a different distribution is a data shift problem [142]. A problem found in many real world applications. These problems are well tackled using domain adaptation techniques [136].

From a design perspective, our generic neural network architecture is as shown in Figure 3.1. We initially trained simple models called the baseline for each model. These baseline have no domain adaptation components, which means, the domain discrepancy component in figure 3.1 was not involved in each case. The models were optimized and evaluated on all the four benchmarks. Later, the models were trained with the domain discrepancy component for each DA method.

Each of the four methods to be discussed in subsequent sections encompasses three components : feature extraction, regression, and the domain discrepancy phases. The domain discrepancy phase is in the form of domain adaptation. All the three components are learnt jointly in a single training loop. Architecture specific to each model is presented and discussed in detail in the next subsections.

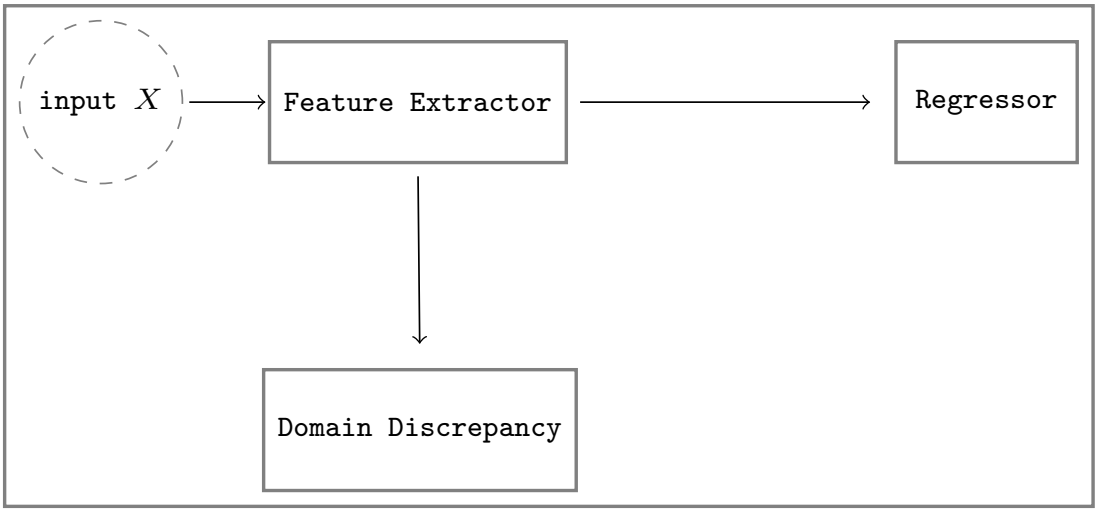


Figure 3.1: The General Counterfactual Architecture

3.2 Correlation Alignment Methods

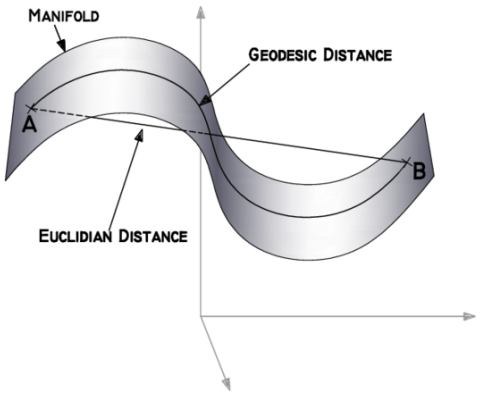


Figure 3.2: Distinction between Geodesic and Euclidean distance
Source: [158]

We introduce Euclidean and Log Euclidean (Geodesic) correlation alignment methods pioneered in DA for causal inference. The correlation alignment methods minimize the distance between the covariance matrices of the treated and the control representations produced by the feature generator of the respective architectures. The distance between the covariance matrices of the representations of the treated and control is minimized in an effort to balance the bias between the two treatment groups. The NN architecture for each method has components that work together to produce the final outcome. The feature generator component of the architecture learnt common representations which are used in computing the covariance matrices and eventually measure the distance between the two covariance matrices. The estimated distance is the correlation alignment distance which is added to the mean square error of the regression task as a regularizer. Both the mse, the correlation alignment regularization, and the weight regularizers were learnt jointly in a single training loop. Each of the two correlation alignment method is an end to end deep feed forward neural network with back-propagation. Their difference as shown in Figure 3.2 depends on how the correlation alignment distance is computed. The novelty of our correlation alignment methods is that we designed NN architecture for counterfactual regression using domain adaptation alignments proposed in [52], [53]. These alignment methods were initially explored in classical domain adaptation but were never been used for balancing the treated and the control group within the causality context. We propose these methods to understand if the alignment methods could offer improvement to a standard, yet peculiar NN architecture for counterfactual prediction.

3.2.1 Counterfactual Euclidean Correlation Alignment ($CeCA$)

We leverage on the correlation alignment loss introduced in [53] for domain adaptation. The architecture in Figure 3.3 has 3 main components (the 3 grey dashed inner rectangles) which correspond to the general architecture depicted in Figure 3.1.

Comparing the general architecture shown in Figure 3.1 with the specific $CeCA$ architecture in Figure 3.2.1, the feature extractor in Figure 3.1 corresponds to the feature generator (first dashed grey rectangle with an arrow directed towards it from the input layer) in Figure 3.2.1. The Euclidean correlation alignment component in the $CeCA$ architecture is equivalent to the domain discrepancy part shown in the general architecture in Figure 3.1. The third component in Figure 3.1 is the regressor which is equivalent to the regressor (the third dashed rectangle with three layers) component in the $CeCA$ architecture.

First component in the architecture shown in Figure 3.2.1 has 2 dense layers. The first layer has N neurons while the second has a 2 neurons fixed. The choice of both N and the fixed 2 neurons was made after several runs of experiments with different number of neurons N. The neurons configurations at the two layers which give the best predictions on the test data amongst all experiments were retained. The feature generator component of the architecture receives the input X comprising of both the treated and the control from the input layer and learn a non-linear representations of the input X. The representations obtained as the output of the generator are then passed to the other two components of the architecture namely: the domain adaptation function and the regressor. A copy of the output from the generator which are the representations is used by the correlation alignment function to compute the covariance matrices of the treated and the control representations first and then calculates the Frobenius norm. Frobenius norm is a Euclidean distance for the two matrices. The Euclidean distance is the square root of the sum of all squares of its elements. The distance score obtained is used by the regression component of the network as an additional regularization.

Another copy of the feature generator output goes to the concatenating layer which appends the treatment feature and then passed to the regressor. The reason for not including the treatment feature in the representation learning phase is that the feature generator could easily learn based on the treatment feature and therefore compromise the main aim– which is balancing the pre-treatment features. The regressor has two dense layers before the prediction layer, both of which have N number of neurons. Again, the reason for the choice of these Ns is based on the quality of the predictions. At the regression phase, the combined treatment feature and the representations produced by the feature generator is passed to the regressor as an input, which pass through additional layers (non-linear transformations) before the prediction layer. The regressor minimizes a custom loss function which is the sum of the mean square error, together with the standard weight regularizer in this case l1 (mean absolute error), and the correlation distance loss computed from the correlation alignment function in the architecture defined in equation 3.3. The treated and the control are jointly learnt, thus, the error in predicting the counterfactuals ($\hat{y}_1$,$\hat{y}_0$) at the regressor component of the network is minimized by promoting features that are indistinguishable (invariant) between the treated and the control. Both the alignment and the mean square error lossses are minimized jointly during the training. The order of the execution is such that the feature generator is first computed, which supply its output to the domain adaptation, and the regression components of the architecture respectively. The domain adaptation is then computed in the form of Euclidean correlation alignment defined in equation 3.3. The regression component is computed last using the distance score obtained from the domain adaptation.

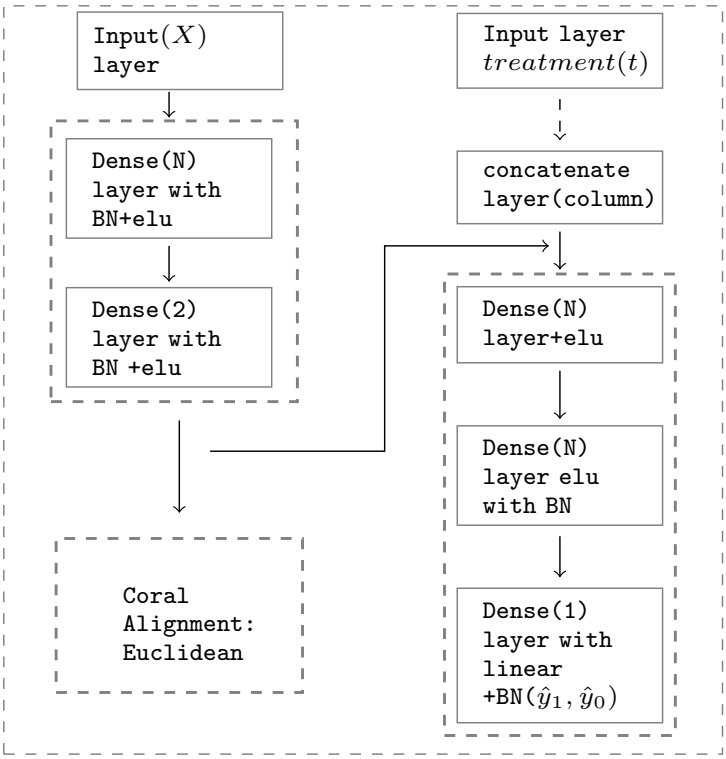


Figure 3.3: Counterfactual Euclidean Correlation Alignment (CeCA) Architecture

We are interested in the correlation alignment loss defined by

$$\mathcal{L}_{coral} = \frac{1}{4 * d^2} * \|\mathcal{C}_{\mathbf{1}} - \mathcal{C}_{\mathbf{0}}\|_F^2 \tag{3.3}$$

$\mathcal{C}_1$ and $\mathcal{C}_0$ are the co-variance matrices of the respective representations $\mathcal{R}_1^l$, and $\mathcal{R}_0^l$. Given that $\mathcal{R}^l$ represents the internal state or the output representations produced by the feature generator component of the network, $\mathcal{R}_1^l$, and $\mathcal{R}_0^l$ denote the representations for the treated and control groups respectively. $\mathcal{R}_1^l$ and $\mathcal{R}_0^l$ are such that $\mathcal{R}_0^l \cup \mathcal{R}_1^l = \mathcal{R}^l$, and $\mathcal{R}_0^l \cap \mathcal{R}_1^l = \varnothing$. $\|.\|_F^2$ denotes the squared matrix Frobenius norm, and d is the d-dimensional of the activation layers. $1/d^2$ is a normalization term which accounts for the sum of the d^2 term in the Frobenius equation and induces independence of the loss from the feature size.

Breaking down the losses according to each constituent of the network, the regressor component minimizes the mean square error (mse) and is used for all the benchmarks. The feature learning component depicted in figure 3.3 minimizes a cross-entropy loss. The third component of the network is the domain discrepancy which computes a loss by measuring the discrepancy between the treated and the untreated (control) groups using Euclidean correlation alignment defined in Equation 3.3. Note that both the regressor and the domain discrepancy components rely on the output from the feature learning parameterized by the weight $\mathcal{W}_1$ as their input. This is to say we are minimizing

$$\min_{\mathcal{W}_1, \mathcal{W}_2} \left[\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\mathcal{W}_1^{(i)}, \mathcal{W}_2^{(i)}) + \lambda \mathcal{L}_{coral}(\mathcal{W}_1^{(i)}) \right] \tag{3.4}$$

$\mathcal{W}_1$ and $\mathcal{W}_2$ are the weight matrices of the feature generator, and the regressor phase of the network respectively. $\mathcal{L}_{coral}$ is the domain adaptation regularization defined in equation 3.3. The hyperparameter λ determines the degree of the adaptation. A higher λ value would force the network towards learning a model less discriminating while focusing more towards aligning the treated and the control. A small λ on the other hand, would produce a model incapable of reliably predicting the counterfactuals, being that the systematic bias between the treated and control could not be aligned.

3.2.2 Counterfactual Geodesic Correlation Alignment (CgCA)

The CgCA model adopts the use of domain adaptation loss proposed in [52] as an alternative regularization to the Euclidean loss in equation 3.3. The loss is an extension of [53]. [52] argues that covariance matrices are Symmetric Positive Definite (SPD) with non-positive curvature, and thus, Euclidean distance used in [53] does not take into account the structure of the manifold, hence, renders the correlation alignment sub-optimal. To address this, a Log Euclidean distance suitable for capturing the manifold structure is proposed. Geodesic metric could be expressed as

$$\mathcal{G} = \|\log(C_1) - \log(C_0)\|_F \tag{3.5}$$

C_1, C_0 are the covariance matrices of the treated and the control representations respectively as defined earlier under CeCA model. From Spectral Decomposition theorem which relates the structure of matrix with its eigenvalues and eigenvectors, any matrix A of $d \times d$ dimension could be written as

$$A = U\Lambda U^T = \sum_{j=1}^{d} \mu_j \gamma_j \gamma_j^T \tag{3.6}$$

where $\Lambda = diag(\mu_1, ..., \mu_d)$, $U = (\gamma_1, ..., \gamma_d)$. γ_j , and μ_j are the eigenvectors and eigenvalues respectively of the matrix A. Applying equation 3.5 and 3.6 in 3.3 we obtain

$$\mathcal{L}_{Logcoral} = \frac{1}{4 * d^2} * \|\mathcal{U}\, diag(\log(\mu_{\mathbf{1}}), ..., \log(\mu_{\mathbf{d}}))\mathcal{U}^{\mathbf{T}} - \mathcal{V}\, diag(\log(\nu_{\mathbf{1}}), ..., \log(\nu_{\mathbf{d}}))\mathcal{V}^{\mathbf{T}}\|_F^2 \tag{3.7}$$

d is the dimension of the activation layers, U, and V are the matrices which diagonalize the respective eigenvalues $\mu_1, ..., \mu_d$ of the treated and control $\nu_1, ..., \nu_d$ for $\mathcal{R}_1^l$ and $\mathcal{R}_0^l$ activations. The explanations provided pertaining the model architecture under $CeCA$ in Figure 3.3 applies to the architecture of $CgCA$ in Figure 3.4 except for the distance metric used in the covariance alignment function. While $CeCA$ used Euclidean distance in computing the covariance alignment, $CgCA$ used geodesic distance metric defined in equation 3.7. Similar to $CeCA$ architecture discussed earlier, the $CgCA$ architecture shown in Figure 3.4 also has three components when compared with the general architecture in Figure 3.1. Basically, both the $CeCA$ and $CgCA$ models have the same network architecture except for the geodesic correlation alignment component shown in Figure 3.4 and defined in equation 3.7. The feature generator first receive the input features (X) from the input layer, which are then non-linearly transformed to an output called representations. These output representations are passed to the correlation alignment function which compute the

two covariance matrices for the treated and the control representations. After which a geodesic distance is used in computing the distance between the two matrices as defined in equation 3.7. The distance score computed is used at the last stage of the network which is the regression component. At the regression component, an additional treatment feature is appended to the output of the feature generator and is passed to the regression layers as shown in Figure 3.2.2. The root mean square error is minimized along with the domain adaptation loss computed from the correlation alignment function defined in equation 3.7. Counterfactual outcomes $(\hat{y_1}, \hat{y_0})$ are estimated by the regressor.

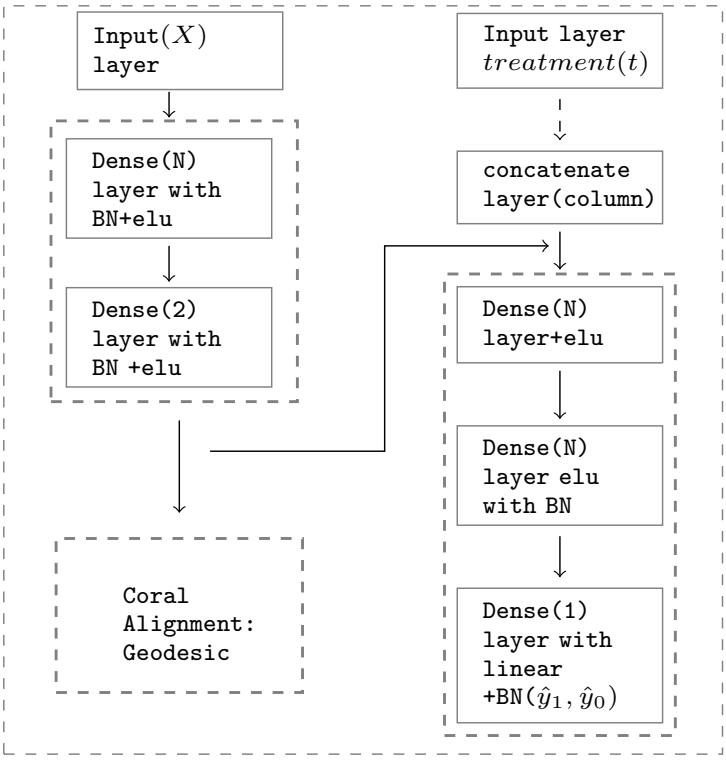


Figure 3.4: Counterfactual Geodesic Correlation Alignment Architecture

The model minimizes the following objective:

$$\min_{\mathcal{W}_1, \mathcal{W}_2} \left[\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\mathcal{W}_1^{(i)}, W_2^{(i)}) + \lambda \mathcal{L}_{Logcoral}(\mathcal{W}_1^{(i)}) \right] \tag{3.8}$$

Note that $\mathcal{L}_{coral}$ in Equation 3.3 is replaced with $\mathcal{L}_{Logcoral}$ defined in Equation 3.7. The datasets used have only the factual outcomes, which are the outcomes for each individual who has received one of the two treatments (treated or control but not both). Our goal for using MSE is to minimize the error in estimating the other outcome under the opposite treatment called the counterfactual outcomes for each individual. The outcome that would be observed if the individual had received the opposite treatment from the factual dataset. Once the counterfactual outcomes are estimated, then for each individual, its factual and counterfactual outcomes are used to estimate its ITE, being that we are equipped with both the two outcomes under the two binary treatment regimes.

3.3 Counterfactual Domain Adversarial Training of Neural Networks (CDANN)

The idea of using domain adversarial training for solving domain adaptation problems was first reported in [54]. The approach learns representations for adapting to data shift in which the source data at training differ substantially with the target data at testing. Like many other domain adaptation methods, this approach is also inspired by the theory of domain adaptation which states that generalization is well achieved if a model is trained with features that are indistinguishable between the source and the target domains [152],[139]. The approach was evaluated for classification problems, precisely: Sentiment analysis and image classification. In causality, researchers have in recent times recognised aligning the distributions of the treated and the control as an effective technique for predicting counterfactuals [51],[41] . We build on this by proposing an adversarial training neural networks for counterfactual prediction. Our adversarial approach is the same with the work proposed in [159] except that [159] only provides a conceptual framework without providing any training details and empirical evaluation on any causality benchmark. This chapter reports the details of our contributions with the adversarial training, which incorporates part of the work published in [159] (that discusses the plausibility of adversarial training of neural networks framework for causality), and in addition, provides results and analysis obtained from evaluating the adversarial training framework on the four most widely used causality benchmarks. Contrary to the use of adversarial training for domain adaptation (aligning the source and the target) reported in [54], here, the adversarial training proposed learn representations for counterfactual predictions. The neural network is trained on the labelled factual data D^F. The objective is to minimize the loss on the regression while maximizing the loss on the domain discrepancy in an adversarial manner. Unlike coral alignment methods (CeCA, and CgCA) introduced earlier, the domain discrepancy is a domain classifier which attempts to classify which individuals in D^F come from the treated and which amongst them come from the untreated (control). We implement this idea with a feed-forward network with standard layers as shown in figure 3.5. The reversed arrow indicates a direction for back-propagation where the weights are multiplied by a -1 causing the domain classifier to struggle distinguishing between members of the groups (treated and control). Implicitly, the inability of the discriminator to discriminate between the treated and the control subjects means the representations of the treated and the control are very much a like which solves one of the key challenges of causal inference with observational study– which is, "correcting" the bias that exist between the treated and the control pre-treatment features. As the training progresses, the regressor learns by promoting features discriminative of the main task (prediction) and invariant of the difference between the treated and the control examples. The treatment feature in the factual data at our disposal has a binary outcome. Subjects with the value of the treatment feature as "1" received a treatment whereas those with a "0" were not treated (control). For the discriminator, we utilize the knowledge of the treatment feature, and use the treatment values as akin to labelled classes in a binary classification setting. Such that treated subject is in a class 1, and the control in a class 0. The task before the discriminator is to use the representations of all the subjects (treated and the control) received from the feature extractor and build discriminator that distinguishes between subjects

that come from the treated and the control.

The use of adversarial training is not new in areas like domain adaptation, however, the novelty of $CDANN$ is deploying the adversarial training within the causality framework to balance the treated and the control groups. The training jointly optimizes :

$$\min_{\mathcal{W}_1,\mathcal{W}_2,\mathcal{W}_3} \left[\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\mathcal{W}_1, W_2) - \lambda \mathcal{L}_d(\mathcal{W}_1, \mathcal{W}_3) \right] \tag{3.9}$$

While the weight matrices $\mathcal{W}_1$ and $\mathcal{W}_2$ remain the learnable weights at the feature and regression phases respectively, $\mathcal{W}_3$ is the weight matrix for the domain discriminator. $\mathcal{L}_d$ is the domain classifier loss and λ plays the same role as described in subsection 3.2.1. The negative sign reverses the minimization to maximization of the domain classifier error. The process is adversarial as we minimize a loss in some components and maximize a loss in another component. This way, features that are invariant to the difference between the treated and the control are learnt, allowing the model trained on the balanced features predict counterfactuals. Adding a domain discrimination loss and jointly minimizing them (MSE, domain losses) is a way of "transferring" knowledge during the training to prevent the regressor from minimizing only MSE on the factual data which in turn generalizes well to the counterfactual distribution.

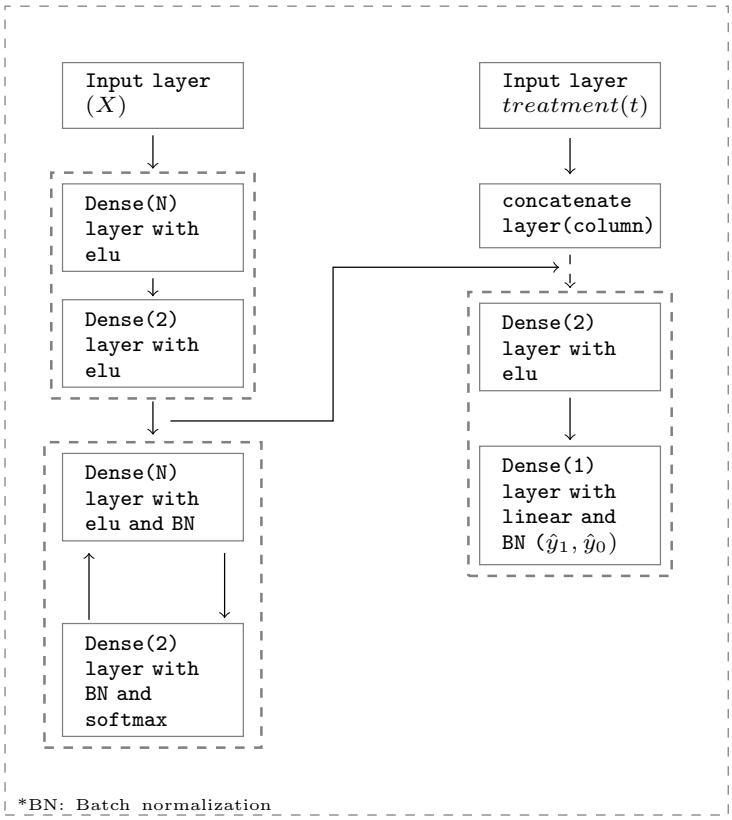


Figure 3.5: Causal Domain Adversarial architecture[1]

[1]N is the number of neurons which depends on the model and dataset. BN is shorten for Batch Normalization.

The forward arrow at the domain adaptation phase acts as an identity transformation during the forward pass; it passes the representations from the feature extraction phase to the domain adaptation phase, while the reverse arrow multiplies a negative one (-1) to the gradients during backpropagation (backward pass) before they are passed back to the feature generation phase.
The description of the architecture pertaining the choice of N and the fixed 2 for neurons remains the same as the architectures previously discussed in figure 3.3, and 3.4. Moreover, the adversarial architecture in figure 3.5 reflects the general architecture of our models shown in figure 3.1. Precisely, we trained a causal model for predicting counterfactuals using domain adversarial training with factual data at our disposal.

3.4 Counterfactual Two Streams (CTS)

Many deep learning methods for domain adaptation learn invariant features by sharing weights between the source and the target distributions [54],[140], [53], [52]. It is however shown that learning invariant features reduces the discriminating power of the classifier [55]. On this premise, two separate streams model architecture pioneered in domain adaptation [55] is developed for causal inference. Our proposed Counterfactual Two Streams (CTS) method is a parallel two streams deep neural network architecture, one stream each for the treated and control as depicted in figure 3.6. The two parallel streams are related through regularization. One regularizer is based on the parallel weights from the feature extractor, and the other on the activations of each stream just before the prediction layer. Penalizing the weights prevents them from being too different from each stream. What this method uniquely contribute is the idea of learning parallel streams for the treated and the control and then relating them through two regularizers. Details on the two regularizers is discussed later in the section

The data $\mathcal{D}^F$ is divided into two; all treated $\mathcal{D}_t^F$ and all the untreated(control) individuals $\mathcal{D}_c^F$ so that $\mathcal{D}_c^F \cup \mathcal{D}_t^F = \mathcal{D}^F$ and $\mathcal{D}_t^F \cap \mathcal{D}_c^F = \varnothing$
. From the architecture, we define the loss along the treated and the control streams as

$$\mathcal{L}^t = \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{D}_t^F} (Y_1(x_i) - \hat{y}_1(x_i))^2 \tag{3.10}$$

$$\mathcal{L}^c = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{D}_c^F} (Y_0(x_i) - \hat{y}_0(x_i))^2 \tag{3.11}$$

$\mathcal{W}_t$ and $\mathcal{W}_c$ are their respective weights. $\mathcal{L}^t$ and $\mathcal{L}^c$ are the standard mean square error losses for the two splits, see section 2.4 for more on loss functions. $|\mathcal{T}|$, and $|\mathcal{C}|$ are the number of treated and control individuals in the samples respectively. Each separate stream minimizes two regularization losses $\mathcal{L}_w$ defined by

$$\mathcal{L}_w = \lambda_w \sum_{i \in \Omega} r_w(\mathcal{W}_t^{(i)}, \mathcal{W}_c^{(i)}) \tag{3.12}$$

and $\mathcal{L}_m$ defined by

$$\mathcal{L}_m = \lambda_m \sum_{i \in \Omega} r_m(\mathcal{W}_t^{(i)}, \mathcal{W}_c^{(i)} | \mathcal{D}_t^F, \mathcal{D}_c^F) \tag{3.13}$$

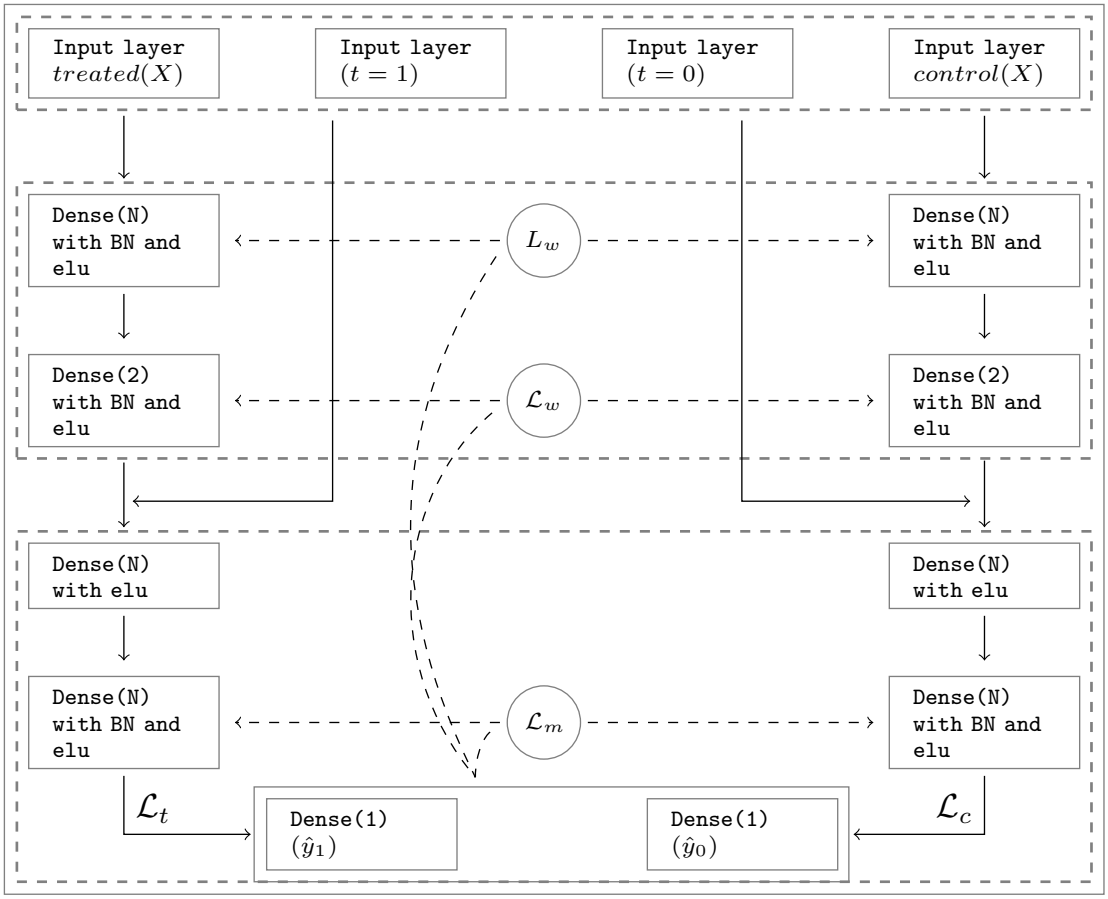


Figure 3.6: Counterfactual Two Stream (CTS) architecture

in addition to the losses in equation 3.10 and equation 3.11, these regularizers relate the treated and the control streams. The regularizer r_w in equation 3.12 measures the weight distance between the two streams and is define by

$$r_w(\mathcal{W}_t^{(i)}, \mathcal{W}_c^{(i)}) = \exp\Big(\|a_i \mathcal{W}_c^{(i)} + b_i - \mathcal{W}_t^{(i)}\|^2\Big) - 1 \tag{3.14}$$

In equation 3.14, a_i and b_i are different scalar parameters learnt together with other network parameters during training for each layer i. The idea of using r_w is that although our intention is to avoid weight sharing, the two streams should be related for the model to be robust to over-fitting [55]. We equally posit that the plausibility of this assumption exists in the case of counterfactual prediction.

The second loss $\mathcal{L}_m$ is based on the additional un-supervised regularizer r_m which uses activations just before the prediction layer to compute the Maximum Mean Discrepancy (MMD) [132],[133] between the treated and the control.

λ_w, and λ_m are regularization coefficients for the weight, and the un-supervised encoder regularizers respectively.

The entire training minimizes

$$\mathcal{L} = \mathcal{L}_t + \mathcal{L}_c + \lambda_w * \mathcal{L}_w + \lambda_m * \mathcal{L}_m \tag{3.15}$$

so that

$$\mathcal{L}(\mathcal{D}^F, \mathcal{W}_t, \mathcal{W}_c) = \frac{1}{|\mathcal{C}|}\sum_{i\in\mathcal{C}}(Y_0(x_i) - \hat{y}_0(x_i))^2 + \frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}}(Y_1(x_i) - \hat{y}_1(x_i))^2 \tag{3.16}$$

3.5 Summary

In this chapter, the four methods deployed for causality with their associated baselines are discussed. As stated earlier, although the domain adaptation components of these methods were pioneered in domain adaptation, however, none of them was ever used for causality. At first, we formulate the counterfactual prediction under the potential outcome framework [6]. Subsequently, methods based on correlation alignment with Euclidean and geodesic distances were discussed. The adversarial approach was later introduced as another method for counterfactual prediction. Unlike the correlation alignment methods, the method used an adversarial training in balancing the treated and control. The last part of this chapter focused on the parallel two stream architecture. Where the treated and the control have separate streams each, however, the two streams were related through regularizations. All the models developed follow the causality formulation described in section 3.1.

Chapter 4

Datasets, Evaluation Metrics, and Training

This chapter comprises 3 sections namely : Datasets, Evaluation metrics, and Training. The Datasets section explains all the four benchmarks used, which includes the nature of the dataset, the task it intends to solve etc. Evaluating models is an important part of a model building, we therefore use Evaluation metrics section to describe all the metrics deployed for the evaluation of our models . The final section in this chapter provides the entire training procedure for the models, highlighting areas of commonalities and also where their training differ.

4.1 Datasets

The 4 datasets reported have variations regarding the number of replications of each dataset, except for JOBS and TWINS which has 10 replications each. Infant Health Development Program (IHDP), and NEWS datasets each has 1000, and 50 replications respectively. These datasets were introduced into causality by different authors with this exact number of replications. Subsequently, most authors in the causality retain these number of replications for fair comparison. Similarly, the train, validation, and test splits ratio for each dataset is also according to the original authors of the datasets. We equally maintain this ratio in our work to give more basis for discussion.

4.1.1 IHDP dataset

The Infant Health development Program (IHDP) is a randomized control experiment conducted in 1985 [160] to measure the cognitive test score between children that receive intervention (treated) and those that received no intervention (the control group). The experiment targeted low-birth weight premature infants which were selected and offered a treatment intervention. The treated infants were those who received the intensive superb quality child care and home visits from a professional health expert. The study collected data on pre-treatment measurements from the child, which include: birth weight, head circumference, weeks born preterm, birth order, whether it is first born, neonatal health index, sex, twin status. The pre-treatment features collected under behaviours during pregnancy include: whether the mother smoked cigarette, drank alcohol, took drugs. At the time of birth, the

age of the mother, marital status, educational attainment (whether she attended high school, completed college etc.), whether the mother was working during pregnancy, and whether she received pre-natal care. The addresses with which the family resided during the entire period of the study is collected. There are 6 continuous, and 19 binary covariates (features) totalling 25 pre-treatment features. Using experimental data provides a base with which observational study can be created, and ensuring that overlap is satisfied. Hill [47] uses the 25 covariate features from the experimental data to simulate the outcomes by generating response surfaces based on the features. All the 25 features are used however, ethnicity is excluded in generating 2 different response surfaces. These response surfaces are known and thus, ignorability assumption could easily be estimated by conditioning on the confounding features used in generating them [47]. The IHDP benchmark used in this thesis is the semi-simulated version introduced by Hill [47]. While preparing the dataset for use in observational study, bias was artificially introduced by removing non white mothers from the treated group bringing the total treated subjects to 139, with 608 control subjects. The task on this benchmark is a regression problem. Each of our models was trained using this benchmark with a view to predict the cognitive test score (real value). Given the pre-treatment features, the treatment feature, and the factual simulated outcomes, our models learn to predict the counterfactuals. The predicted counterfactuals are the cognitive test score of a child under a different treatment (or intervention) than the one the child actually received and the results are evaluated with 1000 replications. The mean of these replications is reported for each metric. The training, validation, and testing splits are in the ratio 60%, 30%, and 10% respectively. More details can be found in [47, 41, 45].

4.1.2 NEWS dataset

The NEWS benchmark introduced in [41] simulates the opinions of a media consumer when presented with multiple news items. Each news item is consumed either via mobile device or on a desktop. Different news item units are presented as a word count , and the outcomes are the reader's experience of the new item. A word count is the number of times a word appears in a single document. The two interventions represent viewing a device on a desktop with $t = 0$ or on a mobile $t = 1$. It is assumed that the customer has preference for reading certain topics on a mobile device. To prepare this dataset, [41] trained a topic model with large text documents from the NewYork Times corpus. A topic model in machine learning is a natural language processing technique used in discovering hidden semantics structure in a text in a collection of documents often called corpus [161]. The data available to these algorithms are the raw word counts, from a vocabulary of 3477 words, selected as union of the most 100 probable words in each topic. There are hundreds of thousands of topics if not millions of them in NewYork Times corpus such as medical, sports, entertainments, religion, finance etc. The NEWS dataset is simulated for regression task and our models aimed at predicting the reader's experience after reading the news item on an alternative treatment. For example, if a reader received a news item on a mobile device ($t = 1$), and his experience (a real value) outcome is observed, what would be the experience of the reader if he had received the news item on a desktop device ($t = 0$)? There are 50 different files of the dataset called realizations, with each file standing as a complete dataset generated with some ran-

dom noise that vary from file to file. In each file, there are 100 rows representing 100 topics, and 3477 columns for the word count. We report the average over all of them. For each realization, the training, validation, and testing sets are in the ratio 60% , 30% , and 10% respectively.

4.1.3 JOBS dataset

The JOBS dataset [162] is a combination of Lalonde's experimental data from the National Supported Work Program (NSWP) [163] and the Observational study from the Panel Study of Income Dynamics(PSID) [71]. The aim of the NSWP is to provide job training (as intervention) with which an increase in outcome (earning) or change in employment status of the participants after the training is expected. The experimental subset of the dataset comprises 297 participants who received the training and 425 control [163]. While the PSID observational comparison group contains 2490 control individuals.

We use the feature set of [71] which is a set of 8 features comprising among others education, previous income earning, and age [71]. The number of features are the same in both the experimental study and the PSID. Because all the treated subjects come from the original experimental study sample, it enables us to estimate the "true" average treatment effect on the treated (ATT). The dataset is constructed for classification task to predict whether a subject is employed or not. The employment status after training for each subject is known and hence are not simulated. We use 56%, 24%, and 20% for training, validation, and test splits respectively on each of the ten (10) replications. The result reported is the average over these replications.

4.1.4 TWINS dataset

This benchmark introduced in [155] is based on the 1989 to 1991 US twin births register [164]. Of interest are twin pairs of same sex both weighing less than 2kg at birth. Under this setting, being heavier amongst the twin pair has $t = 1$ and the lighter twin is assigned $t = 0$. The outcome is a binary which corresponds to the mortality of each twin in their first year of life. A 1 indicates the twin pair died in its first year of life and a 0 means the twin pair lived beyond its first year of life. These outcomes are not simulated being that for each twin pair, there are records from the birth register, pertaining the mortality of both twins (heavier and the lighter twins) in their first year of life. This means we have access to both the outcome when $t = 0$ (lighter twin) and when $t = 1$ (heavier twin). Knowing the outcomes for $t = 0$, and $t = 1$ could be seen as having access to both the potential outcomes. There are 11,984 twin pairs in the dataset with an average treatment effect of -2.5%. The mortality rate for the heavier and the lighter twins are 16.4%, and 18.9% respectively. A bias is induced in the dataset by selectively hiding one of the twins for each twin pair to simulate an observational study.

For each twin pair, 46 features relating to the parents, birth, and pregnancy were collected. These include, parents education; pregnancy risk factors such as herpes, diabetes, smoking, alcohol use; marital status, number of previous births, race; residence, number of gestation weeks before birth (GESTAT10), etc.

We report the average of 10 replications, with percentage splits for training, validation, and testing as 56% , 24% , and 20 % respectively.

4.2 Evaluation Metrics

Access to the counterfactual outcome is impossible in real world. This is one of the challenges in causal inference with the potential outcome framework. Benchmark datasets with (semi) simulated outcomes are used instead. The results provide an indication of the quality of our models. We evaluate our models with 1000 replications of the IHDP datasets, 50 replications for the NEWS dataset, and 10 replications each for JOBS and TWINS dataset respectively. We report the following metrics: Root Mean Square Error (RMSE), Absolute Error for ATT, Precision in Estimating Heterogeneous Treatment Effect (PEHE), Absolute Error for ATE, and Policy Risk. However, we treat Area Under Curve (AUC) as a generic ML metric and is thus defined under loss functions in section 2.

Root Mean Square Error (RMSE):—

The RMSE for ITE denoted as ITE_ε is defined by

$$ITE_\varepsilon = \sqrt{\frac{1}{m}\sum_{i=1}^{m}([\hat{ITE}(x_i)] - [Y_1(x_i) - Y_0(x_i)])^2} \tag{4.1}$$

$\hat{ITE}(x_i)$ is the estimated ITE defined by

$$\hat{ITE}(x_i) = \begin{cases} Y_1(x_i) - \hat{y}_0(x_i), & t_i = 1 \\ \hat{y}_1(x_i) - Y_0(x_i), & t_i = 0 \end{cases} \tag{4.2}$$

$\hat{y}_0$, and $\hat{y}_1$ are the estimated counterfactual outcomes for Y_1, and Y_0 respectively.

Absolute Error for ATE:—

The absolute error for ATE denoted by ATE_ε is defined as

$$ATE_\varepsilon = \left|\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_1(x_i) - \hat{y}_0(x_i)) - \frac{1}{m}\sum_{i=1}^{m}(Y_1(x_i) - Y_0(x_i))\right| \tag{4.3}$$

Precision in Estimating Heterogeneous Treatment Effect (PEHE):—

We report PEHE [47] defined by

$$PEHE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}((\hat{y}_1(x_i) - \hat{y}_0(x_i)) - (Y_1(x_i) - Y_0(x_i)))^2} \tag{4.4}$$

PEHE measures not only the quality of the counterfactual predictions of our model, in addition, it captures the ability of our model to reproduce the ground truth [47]. $\hat{y}_1$ and $\hat{y}_0$ are the model estimates for the treated and control potential outcomes Y_1 and Y_0 respectively.

Absolute Error for ATT : —

The absolute error in estimating the ATT denoted by ATT_ε is the absolute difference between the "true" ATT and the estimated ATT, and it is defined by

$$ATT_\varepsilon = \left| ATT - |\mathcal{T}|^{-1} \sum_{i \in \mathcal{T}} \hat{y}_1(x_i) - \hat{y}_0(x_i)) \right| \tag{4.5}$$

$\mathcal{T}$ are the treated individuals from the experimental group in the JOBS dataset. ATT is defined by

$$\text{ATT} = \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} Y_1(x_i) - \frac{1}{|\mathcal{C} \cap \mathcal{T}|} \sum_{i \in C \cap \mathcal{T}} Y_0(x_i) \tag{4.6}$$

Policy Risk :—

The absence of ground truth for the JOBS dataset prevents us from using PEHE as an evaluation metric. Instead, we use *Policy Risk* proposed in [41]. This quantity measures the average loss in value when a treatment is carried out according to the policy implied by an ITE estimator. Let $\Omega_{\hat{y}}(x) = 1$ be the treated policy if $\hat{y}_1(x) - \hat{y}_0(x) > \beta$, and $\Omega_{\hat{y}}(x) = 0$ otherwise. The risk is computed from the randomized subset of the data for $\beta = 0$. The policy risk is defined as

$$\begin{aligned} P_{risk}(\Omega_y) = 1 - (\mathbb{E}[Y_1 | \Omega_y(x) = 1] * p(\Omega_y(x) = 1) + \\ \mathbb{E}[Y_0 | \Omega_y(x) = 0] * p(\Omega_y(x) = 0)) \end{aligned} \tag{4.7}$$

4.3 Training

In this section, experimental protocols are discussed for each model and its variants. What is common to each model is that before choosing the number of neurons N, for a model, the domain adaptation part of the model was removed by setting $\lambda = 0$ and the model is trained for different number of N ranging from N= $2^0, 2^1, ..., 2^9$. The N that gives the best performance for a metric was chosen as the "right" N for the model. Subsequently, all other variants of each model trained were obtained using the same "best" N. There were cases where the best metrics were achieved with different N, that is, no single N produces the best score across all the metrics. In such case(s), different variants of the model were trained with the selected Ns. Once the Ns were chosen, then each model variant (with a fixed N) was trained for different values of λ. The values of λ ranges from $\lambda = 0, 10^1, ..., 10^5$. Again, we chose the value of λ for which the model variants performed best. The varying λ was intended to understand the effectiveness of the domain adaptation (regularization) and if it is, what value of λ gives the best metrics. Similar to N, there were cases in which best metrics are obtained under different λ. That is, for example, $\lambda = 10^1$ may give the best ATE_ε, and $\lambda = 0$ may produce the best ITE_ε for the same model. We use a binary scale along x-axis while plotting the sensitivity of N to different metrics. While log10 scale is used along the x-axis for sensitivity of λ. As stated earlier, we covered $\lambda = 0, 1, 10, 10^2.., 10^5$, to avoid un-defined outcome when $\lambda = 0$,

we replace $\lambda = 0$ with $\lambda = 10^{-1}$ for plotting convenience, however, it's corresponding value along y-axis is actually that of $\lambda = 0$.

4.3.1 Training : Correlation Alignment

CeCA and CgCA were trained using a total of 1024 samples. Both the treated and control have batch size of 512 each sampled with replacement to make up the total. Where the sample is less than the batch size, all the sample is used. The training was performed for 10,000 iterations. For parameter search and update, the Adam optimizer was used initialized with 0.001 learning rate . The Mean Square Error was used as a loss function on all the four(4) datasets regardless of the dataset problem (regression or classification). The models and their variants were trained with domain adaptation coefficient λ in equations 3.8 and 3.4 set to 1 except otherwise stated. This is done irrespective of the dataset evaluated. That means both CeCA and CgCA models were trained with a weight loss in the ratio 1:1. Only the output of the IHDP and NEWS datasets were centred and scaled with sample mean and variance. There was no pre-processing on input and output of the TWINS dataset. JOBS's input is scaled and centered with the sample variance and mean. The feature extraction phase in figure 3.4 has two dense layers with the first layer having N number of neurons. The number N is varied according to the dataset evaluated. It is found that both CeCA and CgCA models perform better on IHDP and NEWS with N = 256, and the observation holds true for all their variants. For the same performance reason, N = 1 and N= 4 are used for the JOBS and TWINS datasets respectively. The first dense layer in the feature extraction phase is regularized with 0.001 coefficient, also, a dropout layer is used with 0.5 coefficient. The regularization and dropout are repeated on the first layer in the regression phase. Note that setting $\lambda = 0$ collapses CeCA and CgCA to the same model called Counterfactual Regression denoted as CR. Linear variants of CeCA and CgCA denoted by CeCA-L, and CgCA-L respectively were trained by removing all non-linear activations elu from CeCA, and CgCA.

4.3.2 Training: CDANN

The CDANN architecture shown in figure 3.5 is used on all the 4 benchmarks. Both regression and domain adaptation phases indicated in dashed squares receive their input from the feature extractor. The number of neurons N is varied according to the dataset. In the architecture, a dense layer with 2 neurons was fixed, and does not change with the choice of N regardless of the dataset used. Fixing N=2 in some layers is done based on the quality of the regression observed. N is the only parameter we varied in the architecture and its choice was based on the empirical results. We have found this model to work well with N=1 on NEWS and IHDP datasets. On JOBS dataset, N= 2 was chosen while N = 32 was used for TWINS. We trained the model using sampling with replacement; batch size of 512 each for the treated and control adding up to 1024. Where the sample is less than the batch size, the entire sample was used. Throughout the experiment, training was done for 10,000 epoch for all the datasets. To optimize the parameter $\mathcal{W}_2$, we chose Adam and set the initial learning rate to 0.001. While $\mathcal{W}_1$, and $\mathcal{W}3$ were optimized using SGD with the Keras default settings. The outputs of IHDP and NEWS were scaled

and centered with the variance and mean of the training sample but their inputs were not pre-processed. For the JOBS dataset, only the input was scaled and centered, neither the input nor the output of the TWINS dataset was pre-processed. Both layers in the feature extraction stage were L1 regularized with coefficients 0.001, and 0.01 respectively. The layer at the regression stage just before the prediction was also L1 regularized with 0.001 coefficient. We chose λ for the domain adaptation loss (cross entropy loss) in equation 3.9 to be 0.2. The choice of λ was informed by the improved quality of the empirical results. The Mean square error alongside with the domain loss in equation 3.9 were jointly optimized in the ratio 1:0.2. We indicate a specific value of λ as well as the number of neurons N in bracket example, for CDANN, CDANN ($\lambda = 0, N = 1$) means CDANN was trained with $\lambda = 0$, that is, with no domain adaptation and N=1. Its linear variant is written as CDANN-L ($\lambda = 0, N = 1$) . CDANN-L is obtained by removing all non-linear elu activations in figure 3.5.

4.3.3 Training: Counterfactual Two Stream

The Counterfactual Two Stream (CTS) in figure 3.6 has two parallel streams, each of which is an exact replica of the other. The feature extraction phase has two dense layers with the first having N neurons which depends on the dataset used. The second dense layer has 2 neurons fixed, and is independent of the choice of N. The two subsequent layers in the regression phase were batch normalized, and have N neurons each with elu activation. Only the first layer in the feature extraction and regression stages were L1-regularized with a coefficient of 0.001.

This model and all its variants were trained with N=4 on TWINS and N=8 on JOBS. While N = 256 is used for both IHDP, and NEWS. These choices (of N) were based on empirical performance on the datasets. Only the outcomes of IHDP and NEWS are standardized with variance and mean of the training sample. No pre-processing is done on the inputs and outputs of TWINS and JOBS datasets. The model was trained for 10,000 epoch and a batch size of 512 each for the treated and the control totaling 1024 training examples. In the event the sample size is less than the batch size, all the samples were used. Parameter optimization is done using Adam initialized with 0.001 learning rate. λ_m and λ_w were chosen as 0.002. The value of σ for Radial Basis Function (RBF) was set at 0.3. The choice of λ_w, λ_m, and σ were based on performance improvement. The variant of CTS with no domain adaptation and N=1 is written as $CTS(\lambda_w = \lambda_m = 0$, N=1), indicating that both λ_m and λ_w were set to zero. Which means, CTS without $\mathcal{L}_m$, and $\mathcal{L}_w$ losses in equation 3.15. CTS has a linear version denoted by $CTS - L$, which was obtained by removing all the elu activations in all the layers of the architecture.

Chapter 5

Result and Analysis

This chapter discusses the performance of CTS, CDANN, CeCA, and CgCA and their variants on all the four benchmark datasets. These results are accompanied with analysis on how each of these models and their variants fare overall.

The chapter is designed to discuss, analyze, as well as compare the results for each benchmark. Initially, we focus on the performance of each model and its variants. Consequently, four tables are presented, one for each benchmark which include results from all the models and their variants. Each results table provides model estimates on all the metric scores. In addition, graphs showing the sensitivity of the models to the number of neurons (N) and to changes in λ are also provided.

A comprehensive and more elaborate comparison of our results against some baselines and other state of the art approaches are discussed in the following subsection. The discussion is aided by a table containing all results produced by these models. Additional graphs and tables are included in the appendix.

5.1 Evaluation on IHDP

Model	Benchmark: IHDP		
	ITE_ε	ATE_ε	$PEHE$
CTS (N=128,$\lambda_m = \lambda_w = 0$)	1.544 ±0.0392	0.372 ± 0.019	2.291 ±0.098
CTS (N=128,$\lambda_m = \lambda_w = 100$)	1.545 ±0.039	0.344 ± 0.016	2.289± 0.098
$CTS - L$ (N=128,$\lambda_m = \lambda_w = 0$)	1.543 ±0.040	0.368 ± 0.0169	2.296 ± 0.098
$CTS - L$ (N=128,$\lambda_m = \lambda_w = 100$)	1.543 ±0.039	0.364 ± 0.019	2.286 ± 0.097
$CDANN - 0$ $(N = 1, \lambda = 0)$	1.875 ± 0.066	**0.148 ± 0.005**	**0.524 ±0.021**
$CDANN - L$ $(N = 1, \lambda = 0)$	4.172 ± 0.167	0.843 ±0.054	5.720 ±0.247
CR $(N = 128, \lambda = 0)$	1.318 ±0.014	0.435 ±0.002	1.216 ± 0.034
$CgCA$ $(N = 128, \lambda = 1)$	**1.271 ± 0.021**	0.390 ±0.021	1.221 ± 0.047
$CgCA$ $(N = 128, \lambda = 10)$	1.387 ±0.028	0.370 ±0.017	1.660 ± 0.067
$CR - L$ $(N = 128, \lambda = 0)$	3.536 ± 0.138	0.846± 0.055	5.720 ± 0.247
$CgCA - L$ $(N = 128, \lambda = 1)$	3.367 ±0.131	0.824 ± 0.054	5.716 ± 0.247
$CgCA - L$ $(N = 128, \lambda = 10)$	3.370 ±0.130	0.833 ± 0.054	5.717 ± 0.246

Table 5.1: Results of all models and their variants on IHDP

At the initial stage, our models were evaluated on IHDP with different number of neurons (N) and a fixed lambda value set to zero as discussed in chapter 4. The value of λ was fixed to zero to enable us understand the effect of N without any influence of λ. Figures 5.1, 5.2, and 5.3 show the responses of all the models trained with respect to N. CTS is shown to effectively estimates ITE_ε, ATE_ε, and PEHE with N = 128 neurons. The same N = 128 best estimates these metrics for CR (Correlation Alignment model with no domain adaptation). While the most effective neuron for $CDANN$ is N =1. There is a general pattern that is consistent for each metric, which is an improvement on the metrics as the number of N increases except for $CDANN$, which shows slightly better performance with small N.

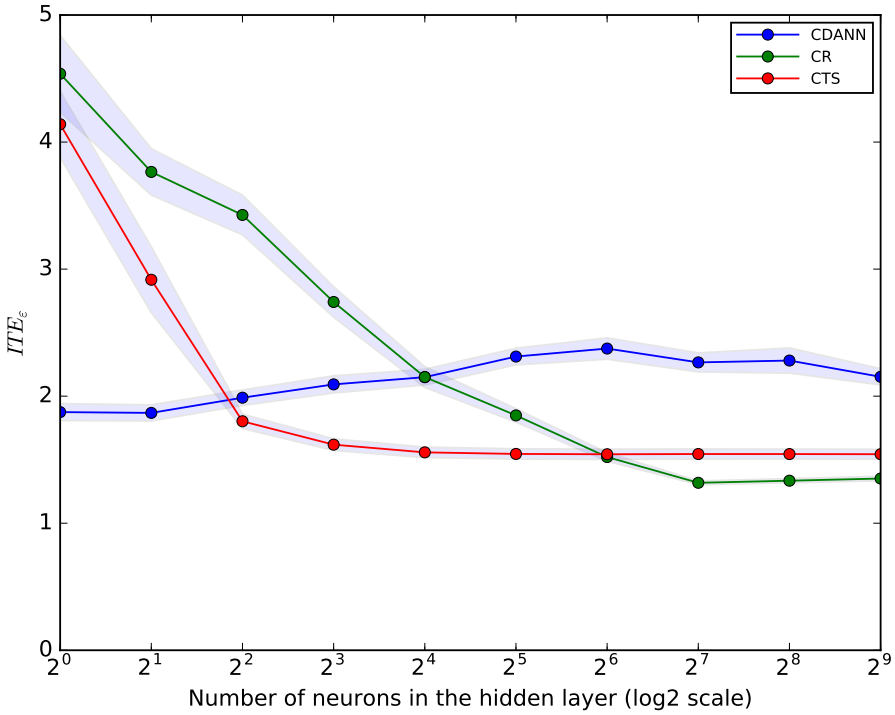


Figure 5.1: Sensitivity of N to ITE for all Models on IHDP

Except for slight fluctuations on ATE_ε, CTS, $CeCA$, and $CgCA$ showed little or no change as λ is varied across all the three metrics, while the performance of $CDANN$ worsens as λ increases, see Figures 5.4, 5.5, and 5.6. From Table 5.1, CTS with $\lambda_w = \lambda_m = 100$ slightly improves ATE_ε and $PEHE$ whereas, its ITE_ε is just as good as CTS with $\lambda_w = \lambda_m = 0$. $CDANN$ with no domain adaptation produced the best ATE_ε and $PEHE$ amongst its variants. While $CgCA$ with $\lambda = 1$ estimated a slightly better ITE_ε and ATE_ε scores, although, CR produced a marginally better $PEHE$. The scores of other variants of $CgCA$ are very similar. Results of other variants of $CeCA$, and $CgCA$ whose scores are not as good as CR are in Appendix A.1.

With regards to non-linearity, both $CTS(\lambda_w = \lambda_m = 100, N = 128)$ and $CTS(\lambda_w = \lambda_m = 0, N = 128)$ showed no difference between them and their linear versions, see Table 5.1.

Out of the three metrics, $CDANN$ produced the best ATE_ε, and $PEHE$ scores,

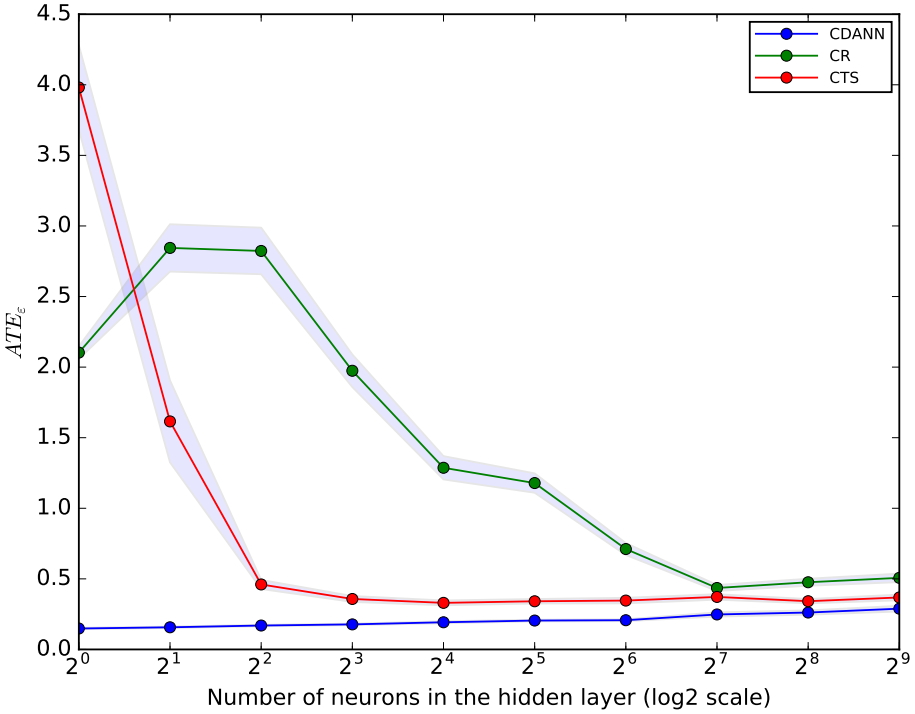


Figure 5.2: Sensitivity of N to ATE for all Models on IHDP

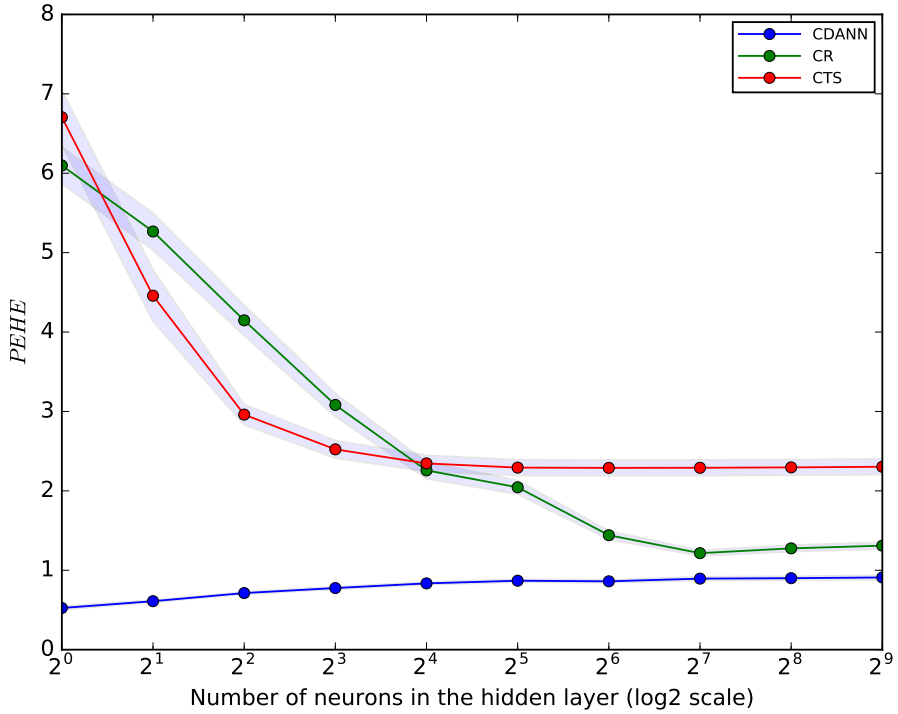


Figure 5.3: Sensitivity of N to PEHE for all Models on IHDP

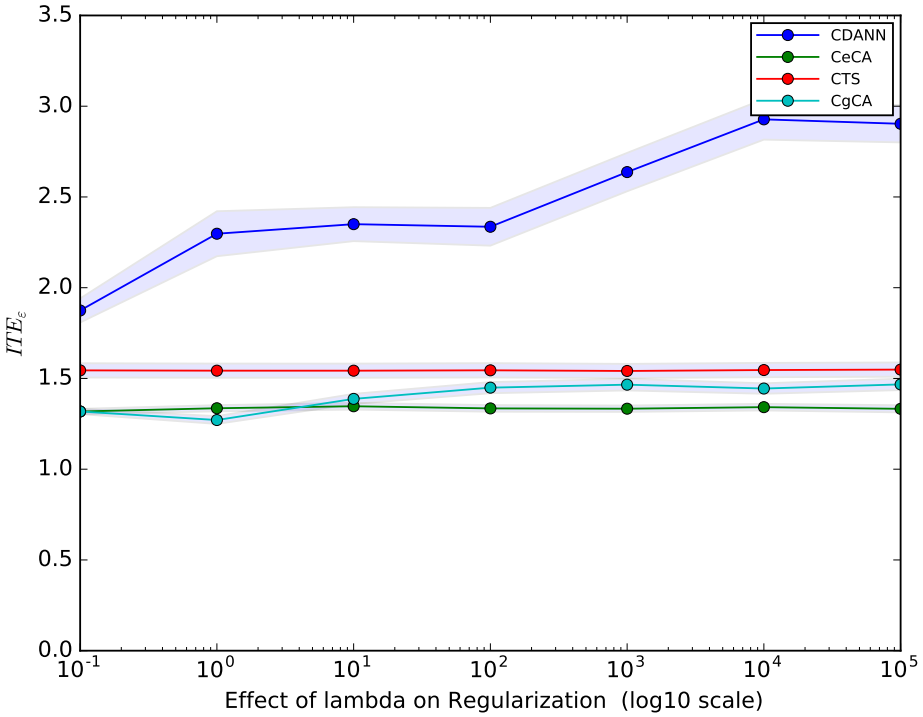


Figure 5.4: Sensitivity of λ to ITE for all Models on IHDP

showing consistently that a single neuron is enough to effectively estimate these metrics on IHDP. It was observed that removing all non-linearity in $CDANN$ plummets ITE_ε from 1.875 to 4.172, ATE_ε declined to 0.843 from 0.148, and PEHE worsened to 5.720 from 0.524 as seen in Table 5.1. This empirically established the importance of non-linearity in estimating these metrics. That is, $CDANN$ requires non-linearity to effectively estimate the scores on IHDP. In contrast to $CDANN$, across all the 3 metrics, CTS remains unperturbed when non-linearity was removed (see Table 5.1). There is significant drop in the quality of these metrics by both $CeCA$, and $CgCA$ compared to their corresponding linear versions when non-linearity was removed as shown in Table 5.1.

From the evaluation of all the four models on IHDP, what was observed is although $CDANN$ used only a single neuron to achieve the overall best scores on the ATE_ε and the $PEHE$ against $CeCA$, $CgCA$, and CTS yet, the model is far less robust to changes in λ (i.e, it is affected the most by changes in λ) possibly due to the single neuron. Similarly, $CDANN$ showed the least changes when N was varied. It was also observed that the three models ($CeCA$, $CgCA$, and $CDANN$) that shared weights (learnt invariant features) were more sensitive to non-linearity than the CTS – a model that relates the two streams but does not share weights. For more results on IHDP, see Appendix A.

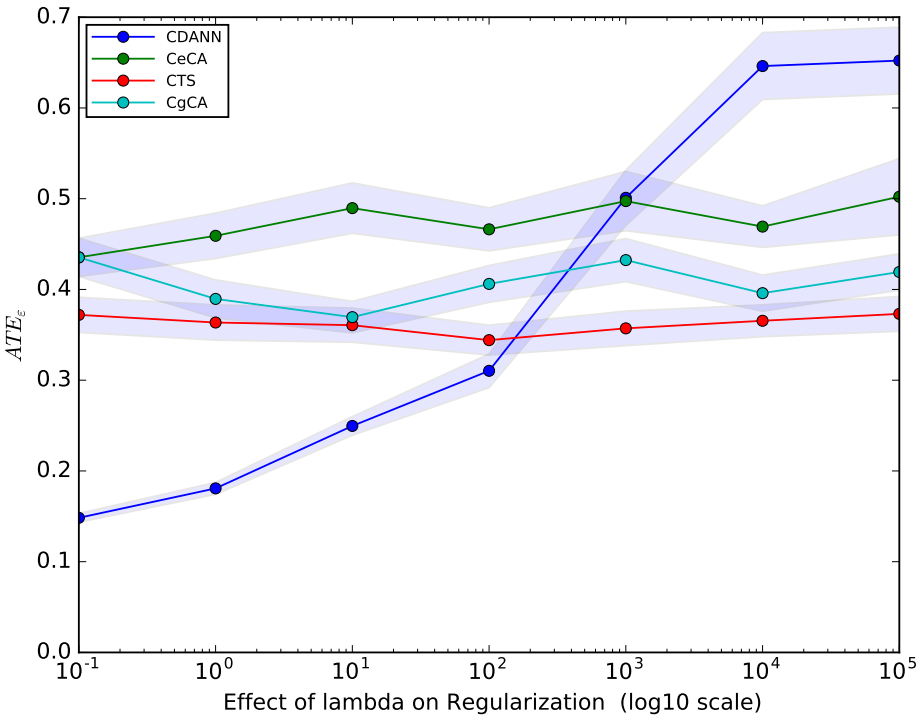


Figure 5.5: Sensitivity of λ to ATE for all Models on IHDP

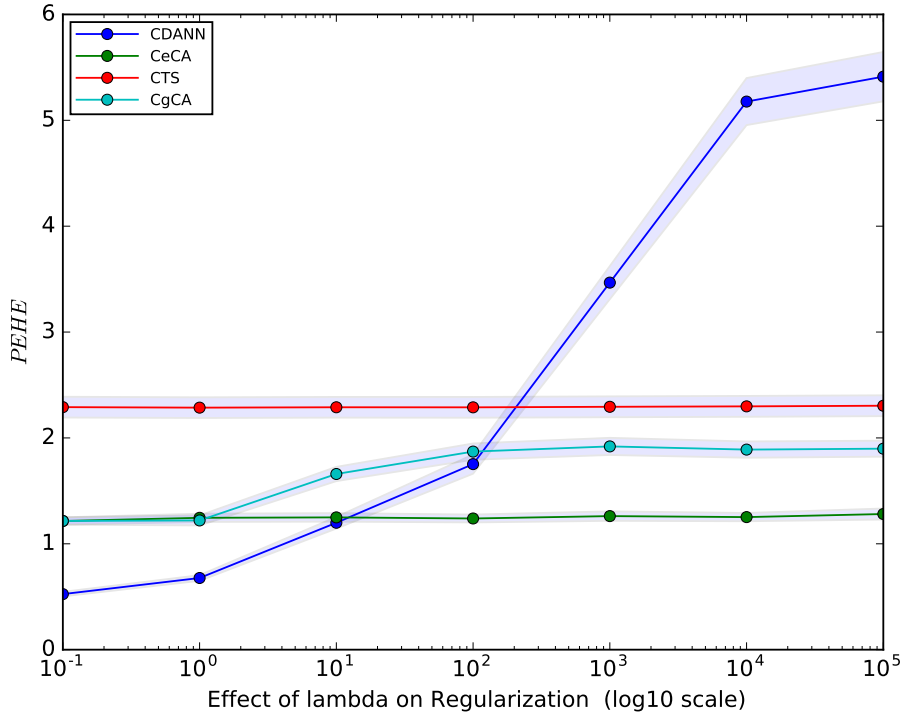


Figure 5.6: Sensitivity of λ to PEHE for all Models on IHDP

5.2 Evaluation on NEWS

Model	Benchmark: NEWS		
	ITE_ε	ATE_ε	$PEHE$
CTS (N=256,$\lambda_m = \lambda_w = 0$)	1.578 ±0.020	0.207 ± 0.024	1.803 ±0.049
CTS (N=256,$\lambda_m = \lambda_w = 100$)	**1.574 ±0.021**	**0.159 ± 0.017**	1.795±0.051
$CTS - L$ (N=256,$\lambda_m = \lambda_w = 0$)	2.406 ±0.084	0.260 ± 0.025	3.200 ± 0.148
$CTS - L$ (N=256,$\lambda_m = \lambda_w = 100$)	2.415 ±0.081	0.229 ± 0.023	3.198 ± 0.139
$CDANN$ ($N = 16, \lambda = 0$)	3.160 ± 0.114	0.327 ± 0.040	2.986 ±0.112
$CDANN$ ($N = 16, \lambda = 100$)	2.383 ±0.157	0.775 ±0.090	2.686 ±0.176
$CDANN - L$ ($N = 16, \lambda = 0$)	5.812 ± 0.330	0.276 ±0.0377	3.373 ±0.182
$CDANN - L$ ($N = 16, \lambda = 100$)	3.663 ± 0.248	0.427 ±0.0695	3.410 ±0.184
CR ($N = 256, \lambda = 0$)	1.670 ±0.068	0.203 ± 0.025	1.70 ± 0.089
$CeCA$ ($N = 256, \lambda = 100000$)	1.732±0.039	0.164 ±0.019	1.883 ± 0.076
$CgCA$ ($N = 256, \lambda = 10000$)	1.692 ±0.055	0.200 ±0.0256	1.733 ± 0.0784
$CgCA$ ($N = 256, \lambda = 1000$)	1.665 ±0.075	0.183 ±0.027	1.705 ± 0.092
$CR - L$ ($N = 256, \lambda = 0$)	3.150 ± 0.178	0.236± 0.035	3.370 ± 0.182
$CeCA - L$ ($N = 256, \lambda = 100000$)	3.342 ± 0.194	0.239± 0.031	3.370 ± 0.181
$CgCA$ ($N = 256, \lambda = 100000$)	1.574 ±0.029	0.307±0.036	**1.651 ± 0.074**
$CgCA - L$ ($N = 256, \lambda = 10000$)	2.851 ± 0.162	0.167± 0.020	3.364 ± 0.180
$CgCA - L$ ($N = 256, \lambda = 1000$)	3.116 ± 0.180	0.223± 0.031	3.368 ± 0.182

Table 5.2: Results of all models and their variants on NEWS

All the models were trained on NEWS dataset with a fixed lambda which was set to zero. While varying N, $CDANN$, and CR were trained with $\lambda = 0$, on the other hand, CTS was evaluated with $\lambda_w = \lambda_m = 0$. For the same reason stated in section 5.1, we want to observe the effect of N under no influence of any λ. It was found that the best N for CTS and CR models is N = 256, while N = 16 produced the best metrics for $CDANN$ model see Figures 5.7, 5.8, and 5.9. A general decline trend (improved metrics) was observed as the number of neurons N increases except for PEHE produced by $CDANN$ which shows stability after N=8 as shown in Figures 5.7, 5.8, and 5.9. Figures 5.10, 5.11, and 5.12 show the sensitivity of domain adaptation regularization across different λ for the four models. For $CeCA$, and $CgCA$, ITE_ε, ATE_ε , and $PEHE$ remain unchanged as λ increases. CTS demonstrates similar pattern except for a little fluctuation as shown in the figures. On the contrary, $CDANN$ reacts to changes in λ, although, it doesn't show a clear trend as λ increases.

Removing non-linearity has the most effect on ITE_ε and PEHE for the CTS, $CDANN$, $CgCA$, and $CeCA$ models. In contrast, ATE_ε remains largely the same across all models, but $CDANN$ which surprisingly shows improvement on the score. It is important to note that the ITE_ε, and ATE_ε results produced when $\lambda = 0$ are at least as good for all the models except for $CDANN$ which shows significant improvement on ITE_ε and PEHE when $\lambda = 100$ (see Table 5.2). Also, CTS and $CeCA$ have shown improved ATE_ε with $\lambda_w = \lambda_m = 10^5$ and $\lambda = 100$ respectively.

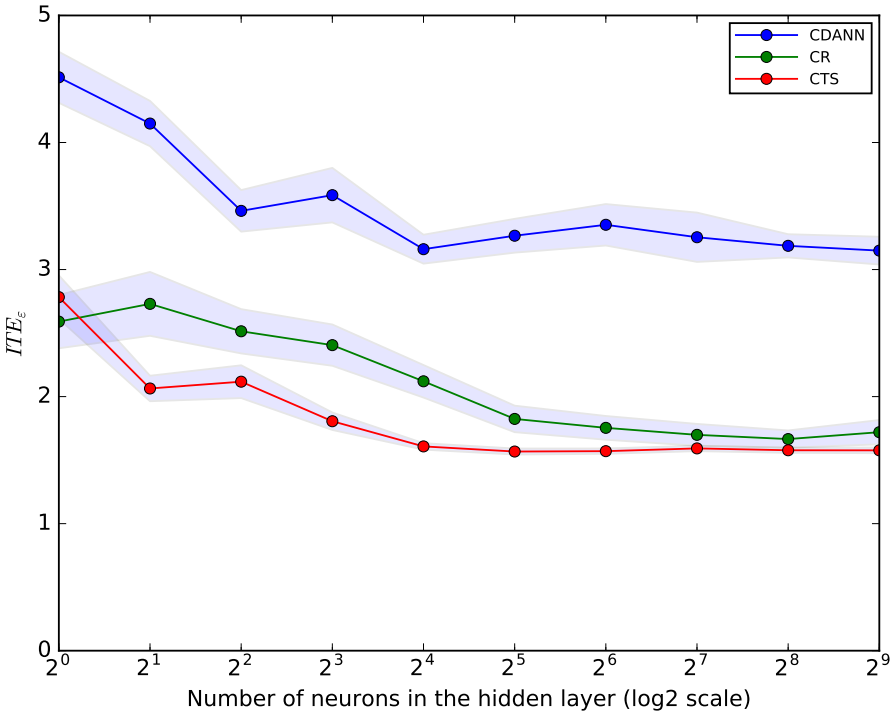


Figure 5.7: Sensitivity of N to ITE_ε for all Models on NEWS

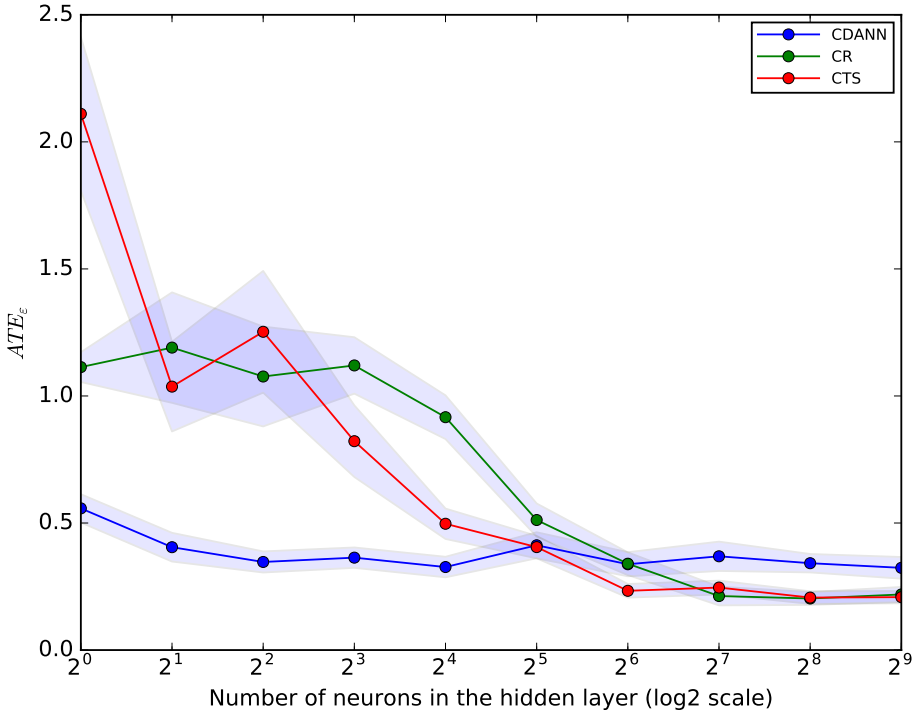


Figure 5.8: Sensitivity of N to ATE_ε for all Models on NEWS

Evaluation on NEWS revealed that $CDANN$ has the smallest number of neurons

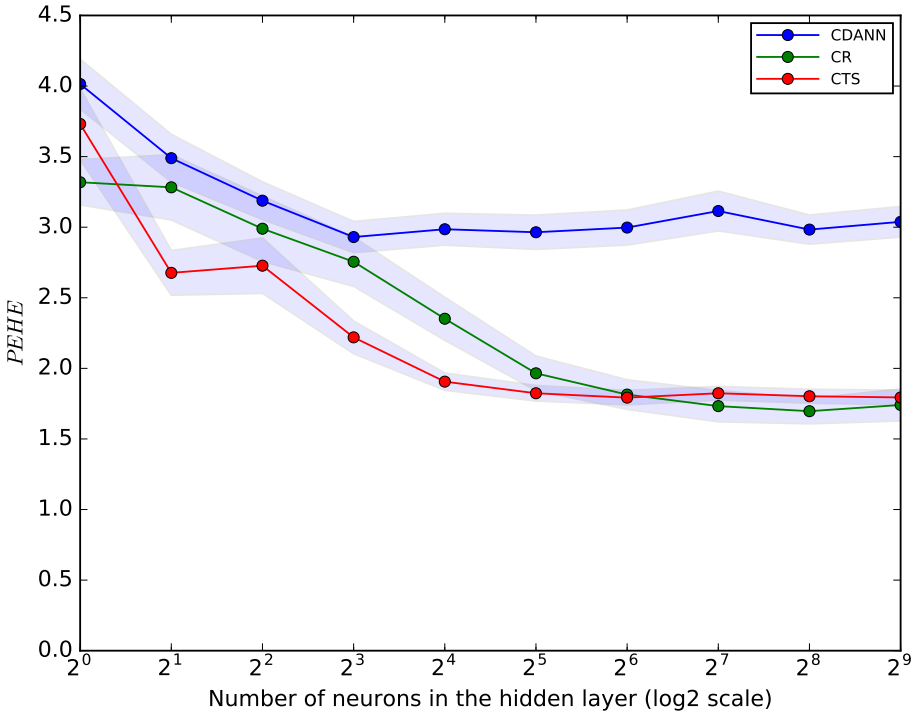


Figure 5.9: Sensitivity of N to PEHE for all Models on NEWS

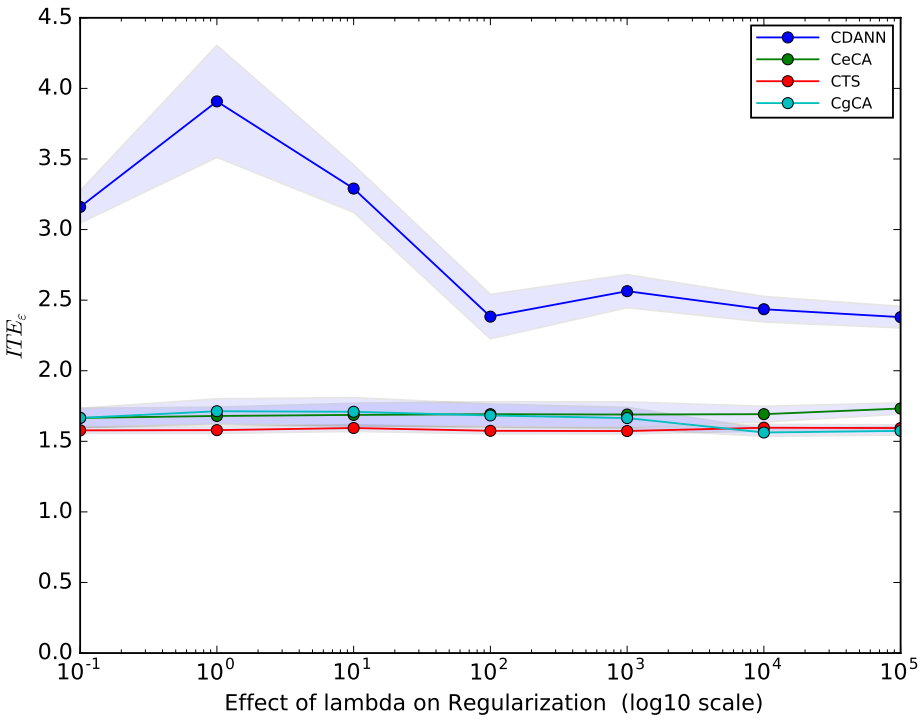


Figure 5.10: Sensitivity of λ to ITE_ϵ for all Models on NEWS

when compared to both CR, and CTS and is observed to be fairly less sensitive

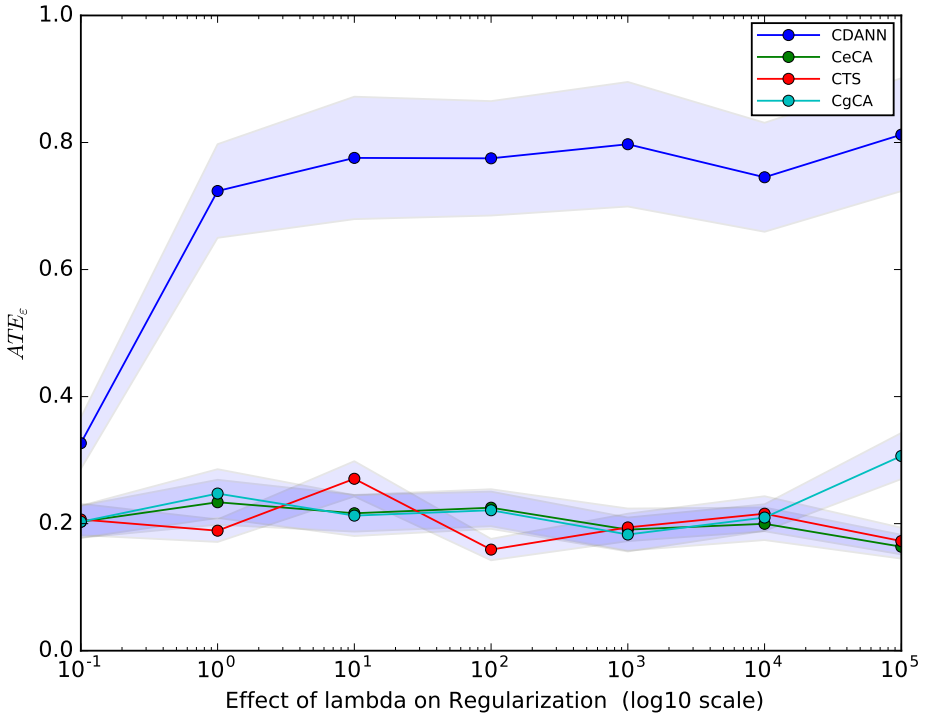


Figure 5.11: Sensitivity of λ to ATE_ε for all Models on NEWS

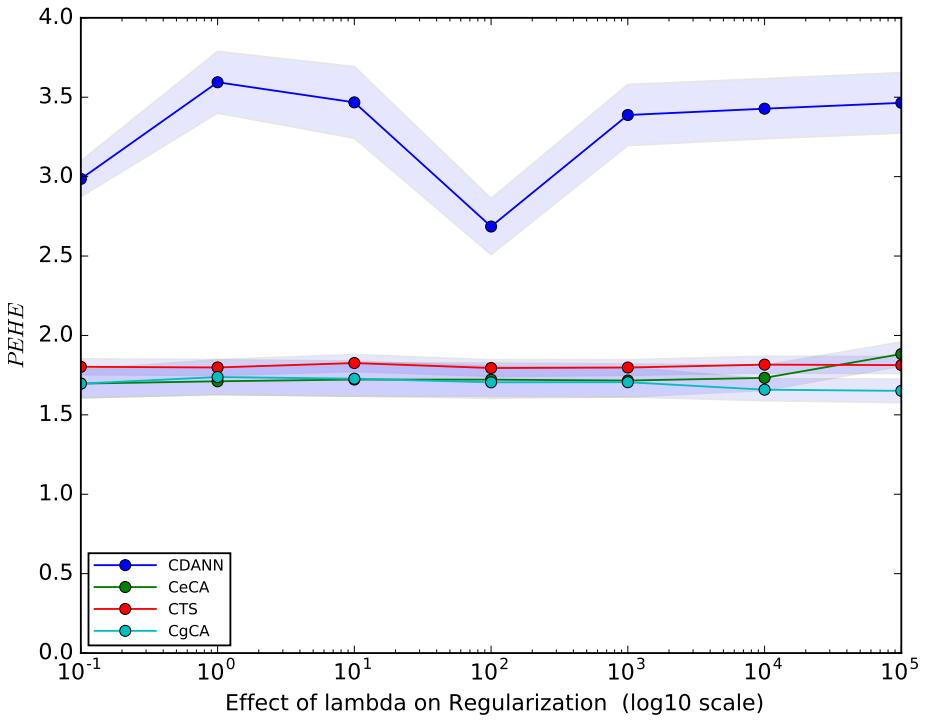


Figure 5.12: Sensitivity of λ to $PEHE$ for all Models on NEWS

to variations in the number of neurons (N). The correlation alignment methods

($CeCA$, and $CgCA$) have shown to be largely insensitive to changes in λ across all the metrics, while the CTS model is also relatively insensitive to λ across all the metrics. On the contrary, $CDANN$ was very responsive to λ variations but with no clear trend. In addition, the performance $CDANN$ on NEWS was the least amongst other models. However, with regards to non-linearity, all the models were responsive to it especially on ITE_{ε}, and $PEHE$ metrics. Appendix B provides additional results of the models on NEWS.

5.3 Evaluation on JOBS

The results obtained from the four models on JOBS are presented in Table 5.3. Primarily, the scores on this dataset are Policy Risk (P_{risk}) and ATT_{ε}.

Model	Benchmark: JOBS	
	ATT_{ε}	P_{Risk}
CTS ($\lambda_m = \lambda_w = 0$,N=2)	0.046±0.009	0.263 ± 0.021
CTS ($\lambda_m = \lambda_w = 0$,N=256)	0.1731 ±0.034	**0.182 ± 0.037**
CTS ($\lambda_m = \lambda_w = 10^4$,N=2)	0.0218±0.0057	0.328 ± 0.016
$CTS-L$ ($\lambda_m = \lambda_w = 0$,N=256)	0.109 ±0.012	0.256± 0.009
$CTS-L$ ($\lambda_m = \lambda_w = 10^4$,N=2)	0.046±0.011	0.303± 0.023
$CDANN$ ($\lambda = 0, N = 8$)	0.069 ± 0.010	0.311 ± 0.077
$CDANN$ ($\lambda = 0, N = 32$)	0.0657 ± 0.011	0.416 ± 0.0977
$CDANN$ ($\lambda = 10, N = 8$)	0.027±0.004	0.292 ±0.010
$CDANN$ ($\lambda = 1, N = 32$)	0.067±0.011	0.237 ±0.0286
$CDANN-L$ ($\lambda = 10, N = 8$)	0.037 ± 0.009	0.314 ±0.000
$CDANN-L$ ($\lambda = 1, N = 32$)	0.077 ± 0.008	0.931 ±0.069
CR ($N = 1, \lambda = 0$)	0.012 ±0.002	0.314 ± 0.000
CR ($N = 32, \lambda = 0$)	0.101 ±0.021	0.228 ± 0.021
$CeCA$ ($N = 1, \lambda = 1000$)	**0.0048± 0.001**	0.314 ±0.000
$CgCA$ ($N = 1, \lambda = 100$)	0.006 ± 0.001	0.314 ±0.000
$CgCA$ ($N = 32, \lambda = 100$)	0.077 ±0.011	0.262 ±0.008
$CR-L$ ($N = 32, \lambda = 0$)	0.072 ±0.001	1.00 ± 0.00
$CeCA-L$ ($N = 1, \lambda = 1000$)	0.0068 ± 0.001	0.314 ± 0.000
$CgCA-L$ ($N = 1, \lambda = 100$)	0.005 ±0.002	0.314 ± 0.000
$CgCA-L$ ($N = 32, \lambda = 100$)	0.046 ±0.002	1.00 ± 0.000

Table 5.3: Results of all models and their variants on JOBS

The best N that effectively estimates ATT_{ε} for CR, CTS, and $CDANN$ trained on JOBS with $\lambda = 0$ are: N=1, N =2, and N =8 respectively. Like with the previous evaluations on $IHDP$ and $NEWS$ benchmarks, a fixed $\lambda = 0$ helps to observe the effect of N on the metrics without any domain adaptation. Both the CR and the $CDANN$ models have N =32 as the best N for Policy risk. While N = 256 is the most effective N for CTS on the same metric (P_{risk}) as shown in Figure 5.13, and 5.14 respectively. Interestingly, in this case, each model has a different N for each metric. There is no clear general pattern as to whether the two metrics increase or decrease with increase in N. The two closely related models $CeCA$ and $CgCA$ exhibit similar behaviour in response to λ on ATT_{ε} as shown in Figures 5.15, and 5.16. The ATT_{ε} score for the two models are at best when $\lambda = 10^3$ and $\lambda = 10^2$ respectively. Using domain adaptation improved slightly the ATT_{ε}, although, it worsened the P_{risk} from 0.228 to 0.262 without DA. $CDANN$ had its best ATT_{ε} when $\lambda = 10$. CR and CTS achieved the lowest (best) policy when $\lambda = 0$, whereas $CDANN$ produced the least (best) Policy with $\lambda = 1$. It is difficult to draw

inference on whether increasing λ increases or decreases the quality of both metrics from the figure. The effect of removing non-linearity while estimating these metrics on JOBS for all the models is shown in Table 5.3. On the one hand, CTS showed robustness when non-linearity was removed. $CDANN$, $CeCA$, and $CgCA$ on the other hand need non-linearity the most in estimating P_{risk}. Removing non-linearity affects P_{risk} more compared to ATT_{ε}. For instance, $CDANN(N = 32, \lambda = 1)$ with non-linearity whose Policy value is 0.237 suddenly dropped to 0.931 with linearity. Similarly, $CR(\lambda = 0, N = 32)$ declined to 1.00 from 0.228 without non-linearity as shown in the Table 5.3.

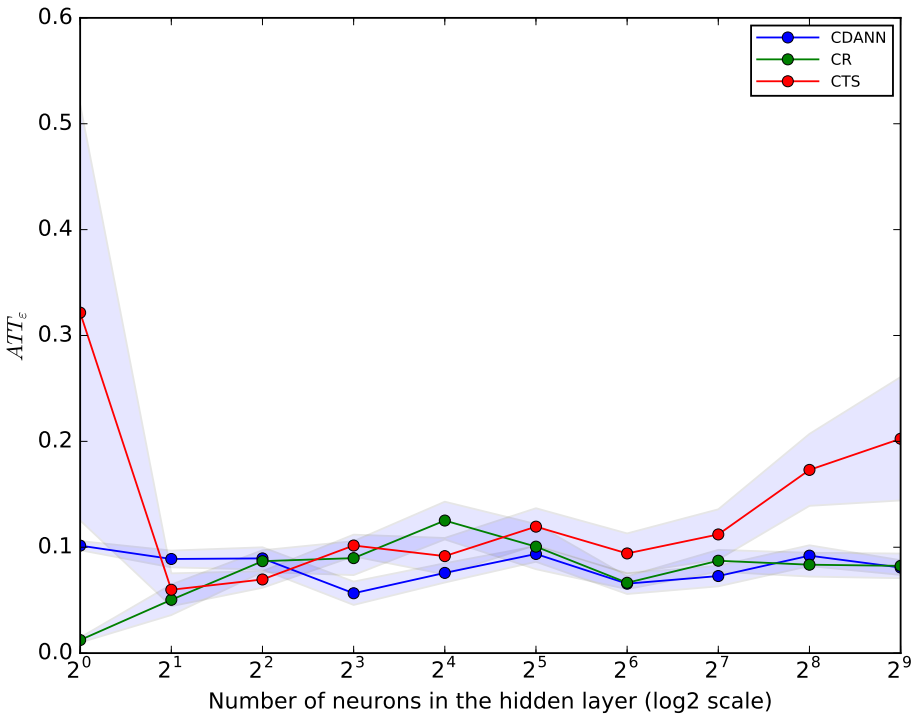


Figure 5.13: Sensitivity of N to ATT_{ε} for all models on JOBS

It was observed from the JOBS evaluation that no model amongst all the models effectively estimates both ATT_{ε} and P_{risk} with one neuron (i.e, with the same neuron). That is, each model produced its best metric score using a different N. For both P_{risk}, and ATT_{ε}, there is no clear pattern (increase or decrease) as N increases. The correlation alignment methods behave similarly and appeared less sensitive to λ, whereas both CTS, and $CDANN$ showed sharp fluctuations in response to λ particularly on the ATT_{ε}. The CTS model has shown robustness when non-linearity was removed, however, correlation alignment models and $CDANN$ both are sensitive to removing non-linearity. More results from the evaluation of the models on JOBS could be found in Appendix C.

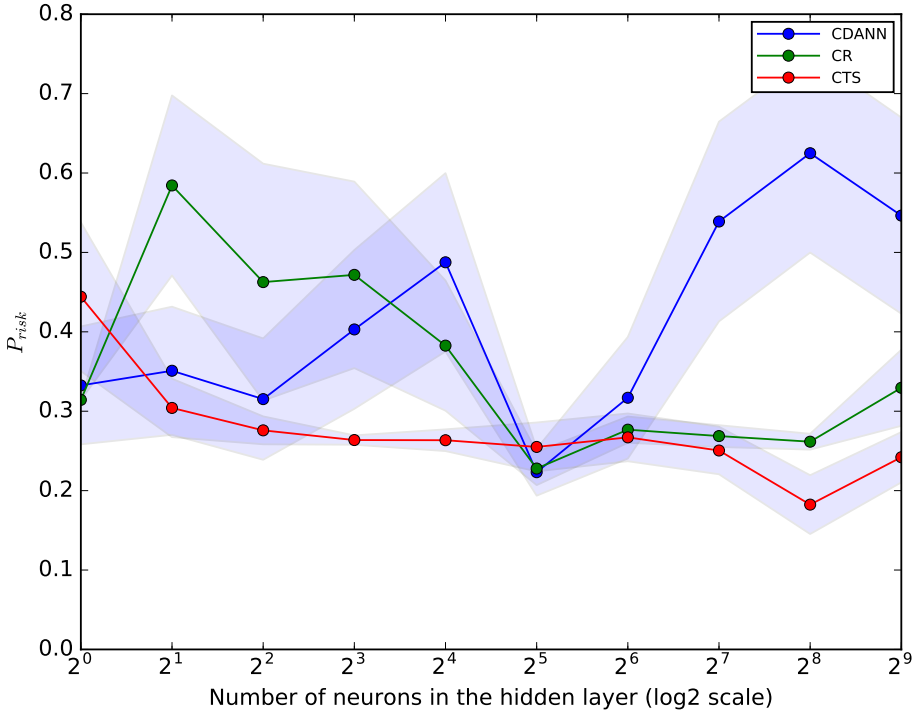


Figure 5.14: Sensitivity of N to P_{risk} for all models on JOBS

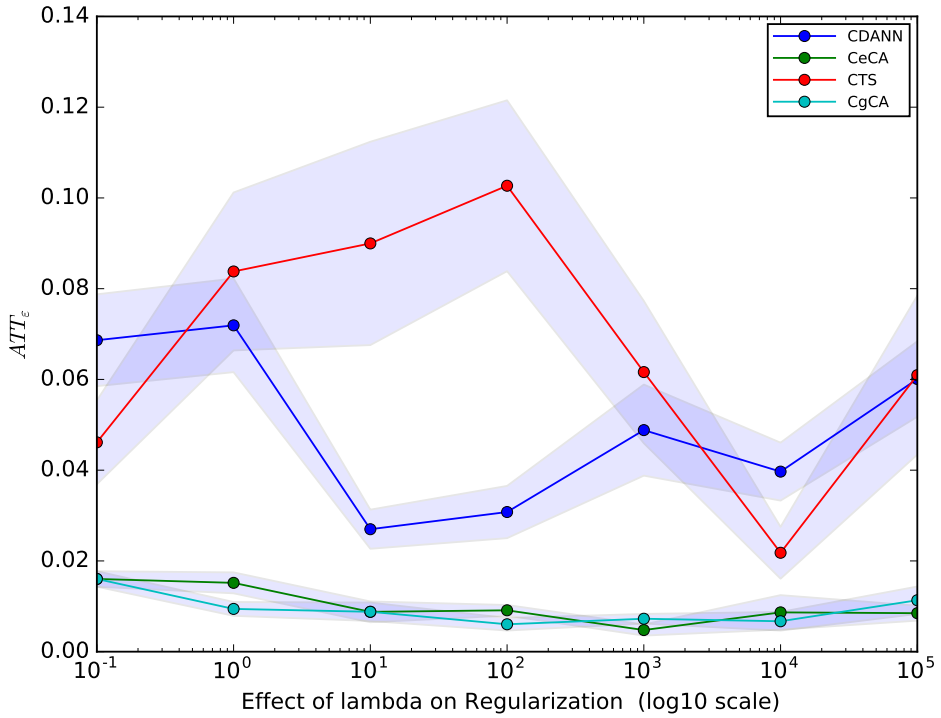


Figure 5.15: Sensitivity of λ to ATT_ε for all models on JOBS

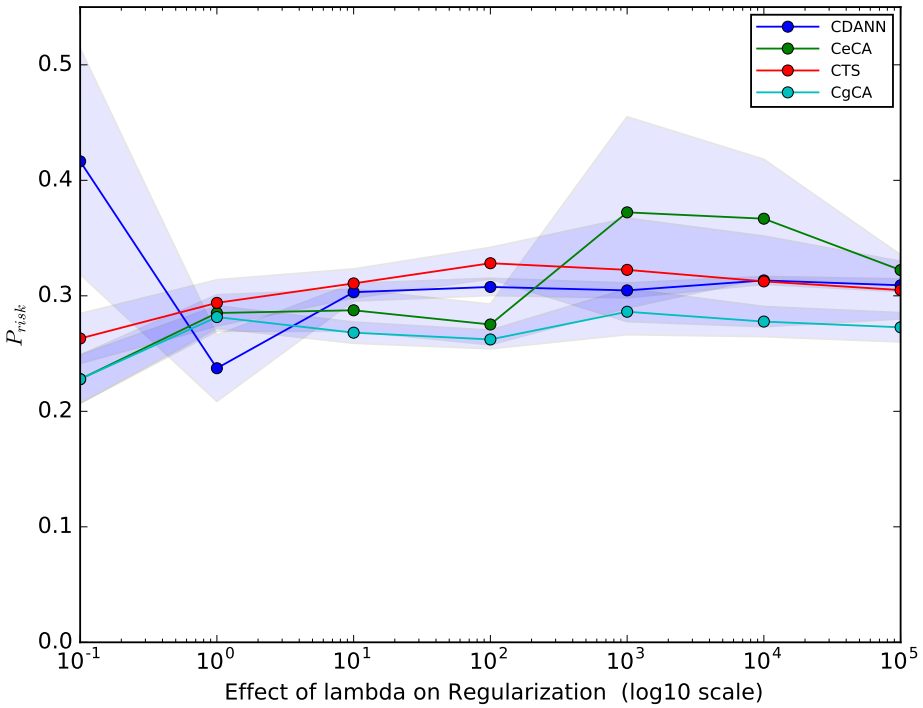


Figure 5.16: Sensitivity of λ to P_{risk} for all models on JOBS

5.4 Evaluation on TWINS

Model	Benchmark: TWINS		
	ATE_ε	$PEHE$	AUC
CTS (N=4,$\lambda_m = \lambda_w = 0$)	0.060 ±0.010	0.374 ± 0.010	0.712 ±0.008
CTS (N=8,$\lambda_m = \lambda_w = 0$)	0.0322 ±0.008	0.363 ± 0.008	0.705 ±0.004
CTS (N=256,$\lambda_m = \lambda_w = 0$)	0.023 ±0.005	0.402 ± 0.006	0.680 ±0.005
CTS (N=4,$\lambda_m = \lambda_w = 1$)	0.053 ±0.005	0.357 ±0.006	0.724± 0.006
CTS (N=8,$\lambda_m = \lambda_w = 1$)	0.042 ±0.008	0.374 ±0.009	0.708 ± 0.005
CTS (N=256,$\lambda_m = \lambda_w = 1$)	**0.022 ±0.005**	0.396 ±0.007	0.674 ± 0.004
$CTS - L$ (N=4,$\lambda_m = \lambda_w = 1$)	0.057 ±0.003	0.349 ± 0.002	0.710 ± 0.003
$CTS - L$ (N=8,$\lambda_m = \lambda_w = 1$)	0.051 ±0.003	0.345 ± 0.002	0.732 ± 0.002
$CTS - L$ (N=256,$\lambda_m = \lambda_w = 1$)	0.040 ±0.005	0.340 ± 0.002	0.740 ± 0.002
$CDANN$ ($\lambda = 0, N = 1$)	0.0432 ± 0.003	0.333 ± 0.002	**0.753 ±0.001**
$CDANN$ ($N = 4, \lambda = 0$)	0.043 ± 0.003	0.334 ±0.002	0.751 ±0.001
$CDANN - L$ ($N = 1, \lambda = 0$)	0.054±0.003	0.338 ±0.002	0.751 ±0.001
CR ($N = 1, \lambda = 0$)	0.032 ±0.000	**0.325 ± 0.000**	0.725 ± 0.003
CR ($N = 4, \lambda = 0$)	0.035 ±0.002	0.330± 0.001	0.74 ± 0.001
CR ($N = 8, \lambda = 0$)	0.0315 ±0.002	0.331 ± 0.001	0.74 ± 0.006
$CeCA$ ($N = 4, \lambda = 100$)	0.031±0.002	0.326 ±0.001	0.75 ± 0.001
$CeCA$ ($N = 8, \lambda = 100$)	0.0286 ±0.002	0.329 ±0.002	0.732 ± 0.005
$CgCA$ ($N = 4, \lambda = 1$)	0.038 ±0.002	0.331 ±0.001	0.746 ± 0.001
$CgCA$ ($N = 8, \lambda = 1$)	0.0286 ±0.008	0.328 ±0.001	0.737 ± 0.005
$CeCA - L$ ($N = 4, \lambda = 100$)	0.044 ± 0.001	0.331 ± 0.001	0.745 ± 0.001
$CgCA - L$ ($N = 4, \lambda = 1$)	0.042 ±0.001	0.330 ± 0.001	0.748 ± 0.001
$CgCA - L$ ($N = 8, \lambda = 1$)	0.046 ±0.00	0.332 ±0.00	0.751 ± 0.005
$CeCA - L$ ($N = 8, \lambda = 100$)	0.045 ±0.001	0.331 ±0.00	0.750 ± 0.001

Table 5.4: Results of all models and their variants on TWINS

Similar to the three previous datasets, we also evaluated CR and $CDANN$ when $\lambda = 0$, while CTS was evaluated with $\lambda_w = \lambda_m = 0$ for different values of N in the range $2^0, 2^1, .., 2^9$. It was aimed at uncovering the sensitivity of N to the metrics in the absence of any influence from λ. The response of all the models when N varies remain similar for the ATE_ε score as shown in Figures 5.17, 5.18, and 5.19, except for CTS which produced some sharp fluctuations when N =2, and 4. The neurons that best estimate ATE_ε score for CTS, $CDANN$, and CR models are: N=256, N=4, and N =8 respectively. The responses of all the models on PEHE show insensitivity to N except for the CTS model. The best ATE_ε for $CDANN$ occurred when $\lambda = 0$ (no domain adaptation). The ATE_ε score for CTS and $CgCA$ improved slightly when $\lambda = 1$. Figure 5.20 shows that on ATE_ε, $CeCA$ is more robust to changes in λ than $CgCA$. λ does not seems to improve the AUC score as shown in Figure 5.21. For instance, $CDANN$ has better AUC score when $\lambda = 0$. Whereas $CeCA$ shows insensitivity to changes in λ as it remained almost unchanged

throughout. The quality of the AUC score for $CgCA$ and CTS deteriorated with increasing λ, while $\lambda = 1$ gives a better AUC for CTS as could be seen in the figure. The $PEHE$ estimates for $CgCA$ and $CeCA$ are not affected by λ whatsoever. The behaviours of CTS and $CDANN$ did not suggest significant effect of λ on their $PEHE$ estimates as shown in figure 5.22. ATE_ε shows a little effect when non-linearity was removed. This observation happened across all models. The same was observed for the $PEHE$ score as shown in Table 5.4. AUC is also not different when linearity was used.

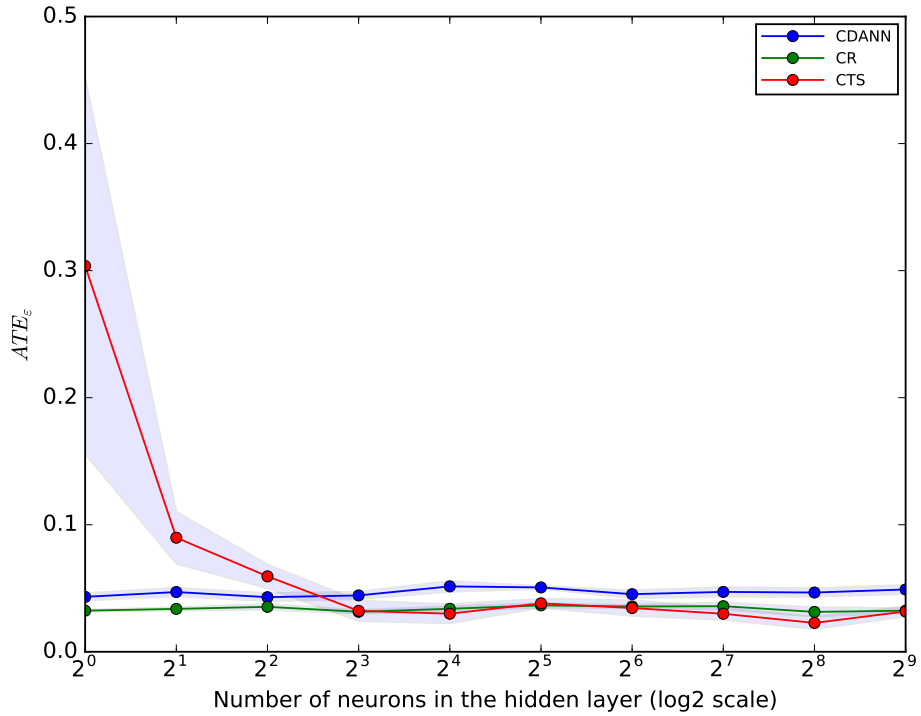


Figure 5.17: Sensitivity of N to ATE for all models on TWINS

Analysing the outcomes of all the methods on TWINS showed that all the models behave fairly similarly across all the metrics as N varies except for a sharp fluctuations occasioned by CTS at $N = 2$, and then it gets better thereafter. It was also observed that $CDANN$ achieved a very good AUC score with only a single neuron. Removing non-linearity does not seem to affect the models much on TWINS. Additional results on TWINS dataset could be found in Appendix D.

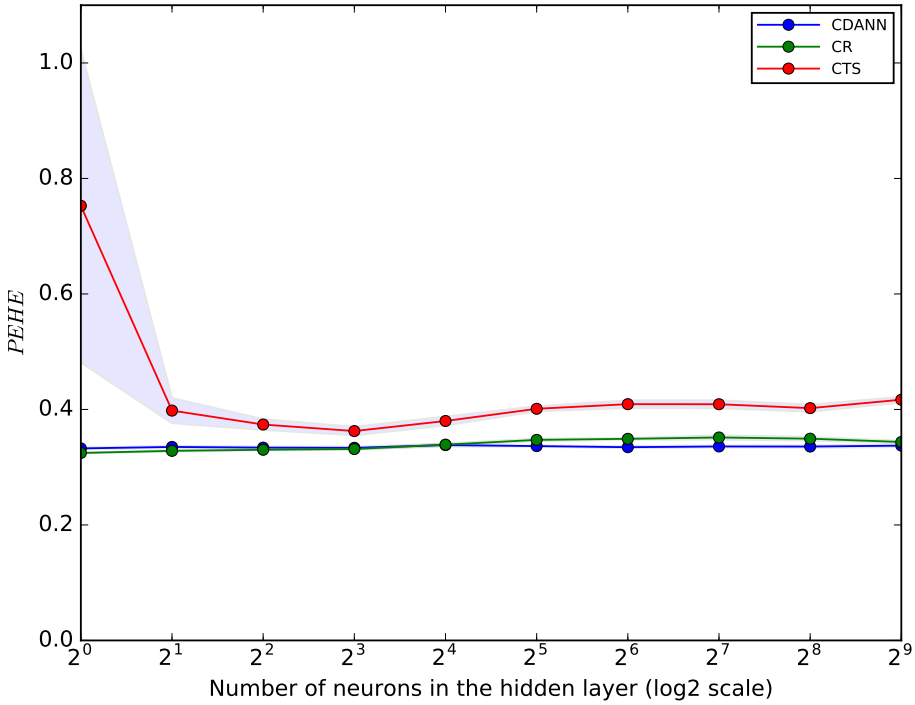


Figure 5.18: Sensitivity of N to $PEHE$ for all models on TWINS

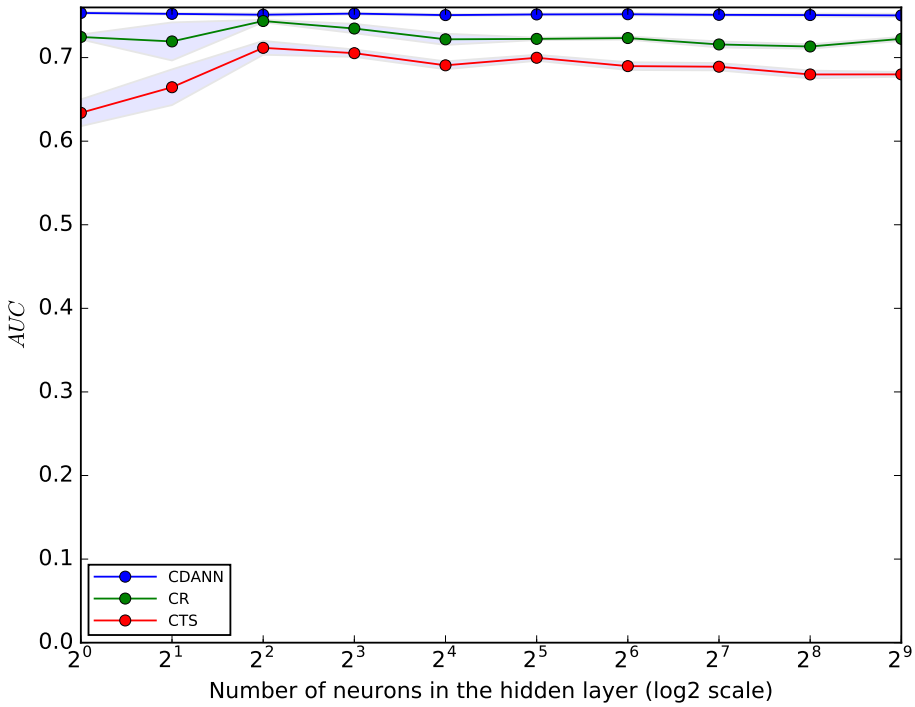


Figure 5.19: Sensitivity of N to $AUC$for all models on TWINS

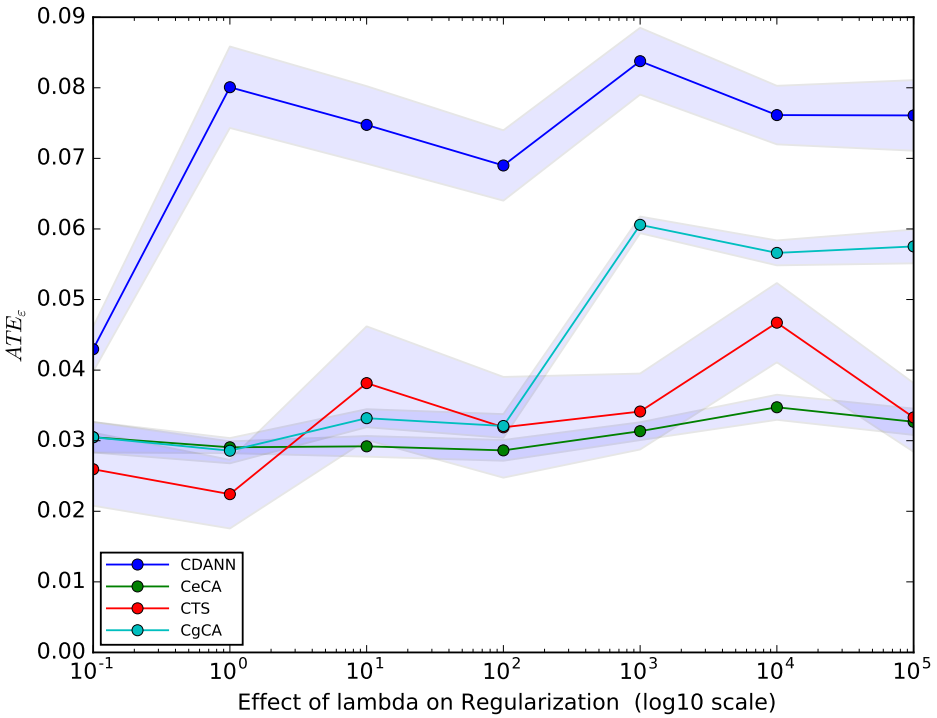


Figure 5.20: Sensitivity of λ to ATE for all models on TWINS

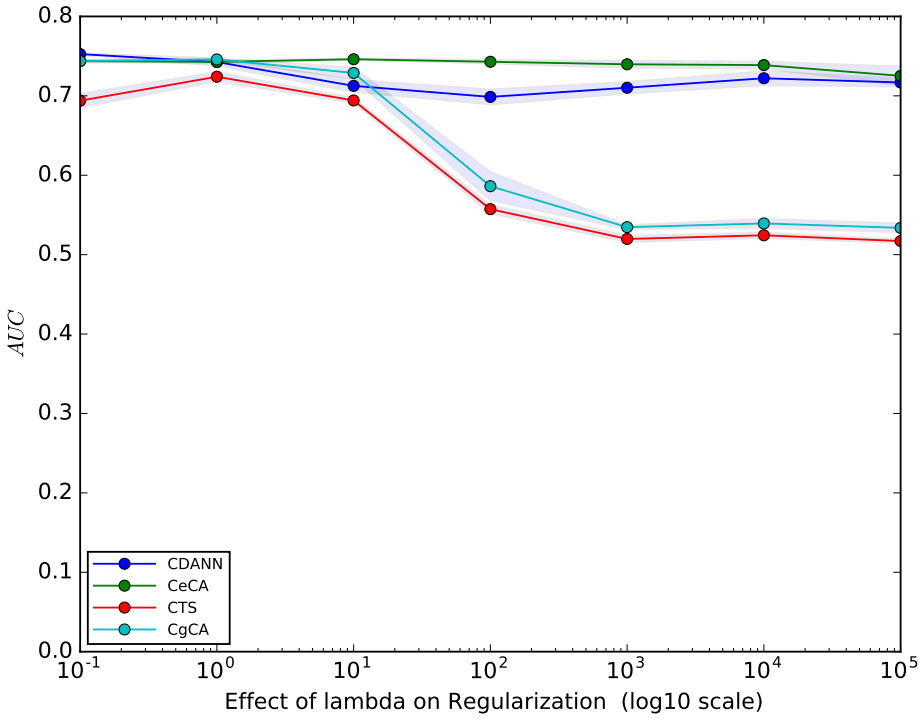


Figure 5.21: Sensitivity of λ to AUC for all models on TWINS

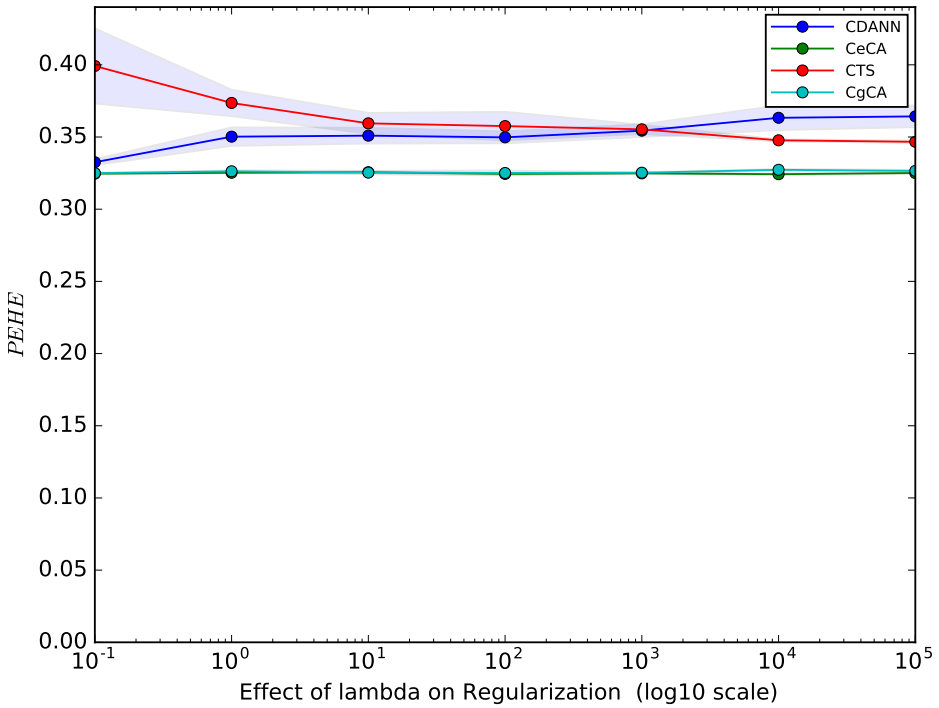


Figure 5.22: Sensitivity of λ to $PEHE$ for all models on TWINS

5.5 Comparison among models

In this section, we compare the performances of our models against each other for each benchmark. It is shown that on the IHDP dataset, the $CDANN$ model is robust to changes in N, when compared to CR and CTS models across all the 3 metrics (ITE_ε, ATE_ε, and $PEHE$) see Tables 5.1, 5.2, and 5.3. Variation in λ is shown to affect the $CDANN$ model the most relative to other models, this observation cut across all the three metrics (ITE_ε, ATE_ε, and $PEHE$). The sensitivity of $CDANN$ to λ could be associated with its number of neurons N=1 in this case. On the same dataset, $CgCA$ with $\lambda = 1$ produced the best ITE_ε score amongst all our models. CR with no domain adaptation yet produced the second best ITE_ε. $CDANN$ produced the least ITE_ε score, however, its estimates for ATE_ε and $PEHE$ were the best amongst all the models. The results under IHDP imply that all the models that produced the best metrics on IHDP learnt invariant features, whereas the CTS model which learnt weights separately underperforms in this case.

Observing Figures 5.7, 5.8, and 5.9 for NEWS dataset, a fairly general decline trend as N gets larger across all models on all the 3 metrics is observed. It was observed that on this dataset (NEWS), the $CDANN$ model is outperformed by other models (CTS, and CR). Similar to what was observed in IHDP, $CDANN$ is shown to be very sensitive to λ compared to the other models ($CeCA$, $CgCA$, and CTS). The least ITE_ε and ATE_ε on NEWS were produced by $CTS(\lambda_m = \lambda_w = 100, N = 256)$. These errors were found to be very close to those produced by $CTS(\lambda_m = \lambda_w = 0, N = 256)$ (i.e, its variant with no domain adaptation). CR estimated the best $PEHE$ of 1.7 on NEWS. The performance of $CDANN$ and its variants across all the metrics on NEWS is the least when compared to the rest of the models.

On JOBS dataset, unlike IHDP and NEWS dataset, there was no clear trend (increase or decrease) relating to changes in either N (See Figures 5.13, and 5.14) or λ (See Figures 5.15, and 5.16) for all the models across all the metrics. The $CTS(\lambda_m = \lambda_w = 0, N = 256)$ model had the least P_{risk} score of 0.182 amongst all the models evaluated. A 0.005 score produced by $CeCA(\lambda = 1000, N = 1)$ is the least ATT_ε error on JOBS. $CeCA(\lambda = 1000, N = 1)$ produced almost similar ATT_ε score. Again, $CDANN$ produced higher errors for both ATT_ε and P_{risk}.

Regarding the TWINS dataset, all our models are less sensitive to changes in N except CTS. This observation holds across all the 3 metrics, as shown in Figures 5.17, 5.18, and 5.19. Similarly variation in λ showed no clear trend across all the models on ATE_ε as shown in figure 5.22. The models appeared less sensitive to changes in λ on $PEHE$. The effect of λ on $CeCA$, and $CDANN$ remained largely unchanged for AUC, whereas, the quality of the AUC score dropped significantly after $\lambda = 10$ as shown in figure 5.19. The lowest error in estimating ATE_ε on TWINS was achieved by $CTS(\lambda_m = \lambda_w = 1, N = 256)$. CTS and its variants produced the smallest error of 0.022 for ATE_ε amongst all other models and produced few other ATT_ε scores that were almost as good as the best score (See Table 5.4). Meanwhile, CR had the least $PEHE$ estimate of 0.325 on TWINS. It is worth

noting that the variation between $PEHE$ scores produced by all the models is very small. The $CDANN$ consistently estimated higher AUC scores. Both $CeCA$ and $CgCA$ estimated better AUC scores than CTS.

Recall that all the results discussed so far under the four benchmarks were based on the interaction of neurons with λ when $\lambda = 0$. The choice targeted the best N under no domain adaptation. In order to test for other interactions of neurons (N) when $\lambda > 0$, with respect to the metric scores, experiments were run for different λ ($\lambda = 1$, $\lambda = 10$, $\lambda = 100$,.,.,$\lambda = 10^5$) on JOBS dataset. We checked if finding the best N for $\lambda > 0$ could in anyway be beneficial or improve the metrics found. That is, whether a better metric could be found when N is chosen under the interaction of different Ns (2^0, 2^1,.,.,.,.,.,.,.,2^9) and some $\lambda > 0$. We only test for the other interactions of Ns with $\lambda > 0$ on $JOBS$ dataset. The best results obtained for the different interactions between different Ns and the different values of $\lambda > 0$ are shown in tables 5.5, and 5.6.

Benchmark: JOBS

lambda (λ)	$Model$	ATT_{ε}
$\lambda = 1$	$CgCA(N = 1)$	0.008 $\pm$0.001
$\lambda = 10$	$CgCA(N = 1)$	0.0106 $\pm$0.002
$\lambda = 100$	$CgCA(N = 1)$	0.0076 $\pm$0.001
$\lambda = 1000$	$CeCA(N = 1)$	0.0083 $\pm$0.002
$\lambda = 10000$	$CgCA(N = 1)$	0.0067 $\pm$0.002
$\lambda = 100000$	$CgCA(N = 1)$	0.0053 $\pm$0.001

Table 5.5: Best ATT_{ϵ} scores for different $\lambda > 0$ on JOBS

Benchmark: JOBS

lambda (λ)	$Model$	P_{risk}
$\lambda = 1$	$CTS(N = 64)$	0.247 $\pm$0.018
$\lambda = 10$	$CTS(N = 1)$	0.238 $\pm$0.0295
$\lambda = 100$	$CDANN(N = 4)$	0.222 $\pm$0.025
$\lambda = 1000$	$CTS(N = 256)$	0.264 $\pm$0.033
$\lambda = 10000$	$CDANN(N = 16)$	0.249 $\pm$0.011
$\lambda = 100000$	$CDANN(N = 1)$	0.251 $\pm$0.006

Table 5.6: Best P_{risk} scores for different $\lambda > 0$ on JOBS

There is no improvement/ benefit observed on the two metrics in all the outcomes as shown in the table. However, correlation alignment methods ($CgCA$, $CeCA$) have produced very good results close to the best results obtained when neurons were chose with a fixed $\lambda = 0$. Some graphs of these interactions (with $\lambda > 0$) on JOBS are in Appendices E and F.

5.6 Results and Statistical Significance

Models	alpha (α) = 0.05		
	Statistic		P-value
$CgCA(N = 128, \lambda = 1)$ vs $CR(N = 128, \lambda = 0)$	$ITE\varepsilon$	0.29	0.00
	ATE_ε	—	—
	$PEHE$	—	—
$CDANN(N = 1, \lambda = 0)$ vs $CDANN(N = 1, \lambda = 1)$	$ITE\varepsilon$	—	—
	ATE_ε	0.085	0.0013
	$PEHE$	0.125	0.00

Table 5.7: Statistical significance for Models that produced the best scores on IHDP

Models	alpha (α) = 0.05		
	Statistic		P-value
$CTS(N = 256, \lambda_m = \lambda_w = 100)$ vs $CTS(N = 256, \lambda_m = \lambda_w = 0)$	$ITE\varepsilon$	0.12	0.841
	ATE_ε	0.24	0.095
	$PEHE$	—	—
$CgCA(N = 256, \lambda = 100000)$ vs $CR(N = 256, \lambda = 0)$	$ITE\varepsilon$	—	—
	ATE_ε	—	—
	$PEHE$	0.18	0.358

Table 5.8: Statistical significance for Models that produced the best scores on NEWS

Models	alpha (α) = 0.05		
	Statistic		P-value
$CTS(N = 256, \lambda_m = \lambda_w = 0)$ vs $CTS(N = 256, \lambda_m = \lambda_w = 1)$	$ATT\varepsilon$	—	—
	P_{risk}	0.3	0.675
$CeCA(N = 1, \lambda = 0)$ vs $CeCA(N = 1, \lambda = 1000)$	$ATT\varepsilon$	0.8	0.0012
	P_{risk}	—	—

Table 5.9: Statistical significance for Models that produced the best scores on JOBS

The significance test for all the models that produced the best results is presented in tables 5.7, 5.8, 5.9, and 5.10. Earlier in sections 5.1, 5.2, 5.3, and 5.4, the best metric scores under each benchmark were shown in bold. Here, we select the models that produced the best scores in bold and test for the significant difference between the sample of scores the model produced with domain adaptation ($\lambda > 0$) and the sample it produced without domain adaptation ($\lambda = 0$). If the model produced the best score with $\lambda > 0$, we want to understand if indeed domain adaptation is responsible for the improvement or not. In a case where a model produced a best metric score with $\lambda = 0$, then the significance test is performed between the model and its twin copy with $\lambda = 1$. The choice of $\lambda = 1$ is arbitrary, it could be any $\lambda > 0$ to justify the significance test conducted is between a model with domain adaptation ($\lambda > 0$) and the same model without domain adaptation ($\lambda = 0$). Before the test, $\alpha = 0.05$ (significance level) was chosen as the alpha score. Which is the probability

Models	alpha (α) = 0.05		
	Statistic		P-value
$CTS(N = 256, \lambda_m = \lambda_w = 100)$ vs $CTS(N = 256, \lambda_m = \lambda_w = 0)$	ATE_ε	0.20	0.975
	$PEHE$	—	—
	AUC	—	—
$CeCA(N = 1, \lambda = 10000)$ vs $CR(N = 1, \lambda = 0)$	$ATE\varepsilon$	—	—
	$PEHE$	0.5	0.11
	AUC	—	—
$CgCA(N = 1, \lambda = 10)$ vs $CR(N = 1, \lambda = 0)$	$ATE\varepsilon$	—	—
	$PEHE$	0.4	0.31
	AUC	—	—
$CDANN(N = 1, \lambda = 0)$ vs $CDANN(N = 1, \lambda = 1)$	$ATE\varepsilon$	—	—
	$PEHE$	—	—
	AUC	0.6	0.031

Table 5.10: Statistical significance for Models that produced the best scores on TWINS

of rejecting the null hypothesis when it is true [165], [166]. It indicates that we have taken a 5% risk of concluding that the difference we observed between a model with $\lambda > 0$ (with domain adaptation) and $\lambda = 0$ (no domain adaptation) is significant when there was actually no difference or the difference is not significant. We define the null hypothesis as follows : *There is no significant difference between the values of the metrics obtained when domain adaptation is not used ($\lambda = 0$) and when domain adaptation is used ($\lambda > 0$).* We test each of the models that produced at least one of the best metric scores on any of the benchmark. We accept the null hypothesis if the p-value calculated from our samples is lower than the chosen $\alpha = 0.05$. The role of p-value in statistical hypothesis test is to help in deciding whether to reject the null hypothesis [167]. The p-value describe how likely a researcher is to have found a particular set of observations if the null hypothesis were true [167]. The value tells how likely it is that our results (data) could have occurred under the null hypothesis. That is, how often a researcher would expect to see an extreme test statistic than the one calculated by the statistical test if the null is true. In our case, the null hypothesis states that there is no difference among groups (groups of results produced by a model with $\lambda = 0$ and similar model with $\lambda > 0$). The alternative hypothesis is that *there is difference in the results produced by the two models* [168]. We chose the *Kolmogorov Smirnov* statistical test [169], a non-parametric test for continuous data.

Table 5.1 in section 5.1 shows the best results obtained on IHDP benchmark, we therefore test for the significance test on each model that produced any of the best metric scores on IHDP. Here, table 5.7 has all the models which produced the three best metrics. At first, we test whether the difference between the ITE_ϵ scores produced by $CgCA$ with domain adaptation($\lambda = 1$), and the ITE_ϵ score produced by the same model with no domain adaptation ($\lambda = 0$) is statistically significant or not. Recall that there are 1000 replications of the IHDP dataset, from table 5.7, $CgCA(N = 128, \lambda = 1)$, and $CgCA(N = 128, \lambda = 0)$ produced 3 metrics (ITE_ϵ, ATE_ϵ,$PEHE$) each on the dataset. For each replic-

ation, a model estimates 3 metrics on IHDP, meaning that the two variants of $CgCA$ have 1000 ITE_ϵ estimates each. These samples of 1000 observations each are used in computing the p-values. We use the numpy package implementation of the Kolmogorov-Smirnov statistic test on 2 samples [170] with the following setting: $scipy.stats.ks_2samp(data1, data2, alternative =' two-sided', mode =' auto')$, where the $data1$, and $data2$ are the two samples of 1000 ITE_ϵ estimates each from the two $CgCA$ model variants. We ignore the ATE_ϵ, and $PEHE$ scores produced by the $CgCA$ model being that they were not the best scores on $IHDP$. The significance test on the ITE_ϵ metric samples showed a p-value of 0.00, which is smaller than the chosen $\alpha = 0.05$ and hence, we conclude that the difference observed between the two models percisely $CgCA(N = 128, \lambda = 0)$ and $CgCA(N = 128, \lambda = 1)$ is statistically significant with a mean statistic of 0.29. Therefore, we reject the null hypothesis and conclude that there is difference between the values of the ITE_ϵ scores produced by the two $CgCA$ models. For the best ATE_ϵ and $PEHE$ scores on $IHDP$, both of which were produced by $CDANN$ model, the test also showed a p-values of 0.00 (which is less than $\alpha = 0.05$) each. We therefore reject the null hypothesis and conclude that the difference between the results produced by the two CDANN variants is significant. Under the $NEWS$ dataset, there are also 3 metrics just like the $IHDP$. We have seen in section 5.2 that CTS produced the best ITE_ϵ, and ATE_ϵ on NEWS. The significance test is thus run between the $CTS(N = 256, \lambda_m = \lambda_w = 100)$ and $CTS(N = 256, \lambda_m = \lambda_w = 0)$ as shown in table 5.8. The p-values obtained for the ITE_ϵ and the ATE_ϵ are respectively 0.841 and 0.095. Statistically, we cannot reject the null hypothesis in each case having that each of the p-values is larger than the chosen $\alpha = 0.05$. That is, there is no difference between the values of ITE_ϵ and ATE_ϵ produced by the two variant models. Similarly, with a p-value of 0.358 for the $PEHE$ score, we also conclude that there is no difference between the $PEHE$ scores produced by the two variants of $CgCA$ model as shown in the table (table 5.8). The samples used in computing the 3 p-values (one for each metric) have 50 observations (scores), which corresponds to the number of replications in the $NEWS$ dataset.

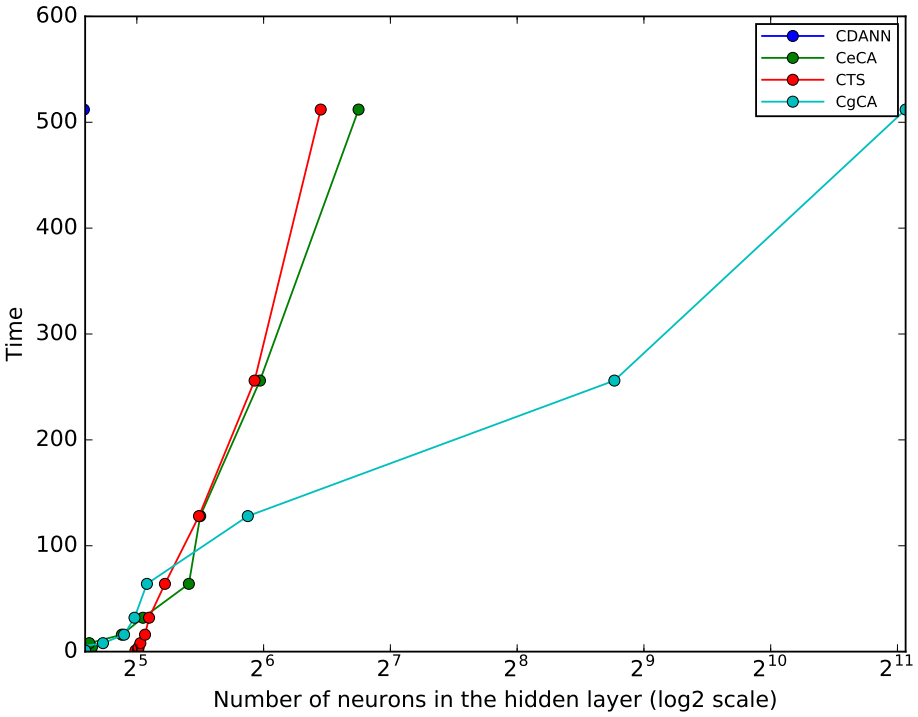
In table 5.9, the significance test for the two metrics on $JOBS$ is shown. The CTS which produced the best P_{risk} has p-value of 0.675 when the statistical test was run between its two variants (($CTS(N = 256, \lambda_m = \lambda_w = 1)$ and $CTS(N = 256, \lambda_m = \lambda_w = 0)$). Recall that $JOBS$ dataset has 10 replications, which means each of the model variant has 10 P_{risk} scores in its sample used in computing the p-value. A larger p-value than the $\alpha = 0.05$ chosen implies we cannot reject the null hypothesis, hence, the P_{risk} scores produced by the two variant models are statistically the same. Contrary to the P_risk score on $JOBS$, the ATT_ϵ estimates from the two variants of $CeCA$ ($CeCA(N = 1, \lambda = 1000)$, and $CR(N = 1, \lambda = 0)$) are different statistically with a p-value of 0.00 (which is lower than the chosen $\alpha = 0.05$). Therefore, we reject the null hypothesis that there is no difference between the ATT_ϵ estimates from $CeCA(N = 1, \lambda = 1000)$ (model with domain adaptation) and $CR(N = 1, \lambda = 0)$ (model variant without domain adaptation). The test of significance for the best models on $TIWNS$ has shown that with p-value of 0.975 which is higher than $\alpha = 0.05$, we cannot reject the null hypothesis, we thus conclude that the ATE_ϵ scores produced by the variants of CTS as shown in table 5.10 are statistically the same (no difference between them). For the $PEHE$ metric, the p-values: 0.11, and 0.31 respectively computed from the samples of the $CeCA$

and $CgCA$ models variants as shown in the table, we conclude that $PEHE$ scores produced by the variants of $CeCA$ and $CgCA$ are statistically similar. In each case, the p-value is larger than the $\alpha = 0.05$ score and hence we cannot reject the null hypothesis. For the AUC score however, the p-value is 0.03 which is smaller than the chosen $\alpha = 0.05$, in this case, we reject the null hypothesis and conclude that there is difference between the AUC scores produced by the two model variants of $CDANN$. There are 10 AUC scores in each sample of the models used in computing the statistical test, which reflects the number of replications for the $TWINS$ dataset. Overall, we observed that out of the 10 test of significance presented in tables 5.7, 5.8, 5.9, and 5.10, only five amongst them are significant. That is, the difference between between two model variants is significant (p-value less than α). And only two cases amongst this five had the best metric scores obtained with $\lambda > 0$. These two cases in which the tests were significant and produced the best metric scores with $\lambda > 0$ are: ITE_ϵ on $IHDP$ produced by $CgCA(N = 128, \lambda = 1)$ and ATT_ϵ on $JOBS$ produced by $CeCA(N = 1, \lambda = 1)$. While the other three cases of the five significant cases produced their best scores with $\lambda = 0$ (no domain adaptation).

5.7 Models and Computational Time

Understanding the most efficient model with respect to time is very relevant, thus, the performance of the four models is evaluated on JOBS dataset. Each model is evaluated with $\lambda = 10^5$, an extreme value of λ and a varying number of neurons N as shown in tables 5.13, 5.14, 5.11, and 5.12 . As expected, each model has its maximum execution time when $N = 512$ from the range of Ns evaluated as shown in the tables. The $CDANN$ is the most time efficient model, followed by the CTS model. On the other hand, $CgCA$ has the highest execution time in this setting amongst all other models. Individual execution times for each model are shown in tables 5.13, 5.14, 5.11, and 5.12. Their corresponding figures for each table are in Appendix G.

Figure 5.23 presents the *neurons* against *time* plots for all the models. Notice that $CDANN$ being the model with the least execution time is almost dominated by other models.



Ta ons
on

Figure 5.23: N against time for all models on JOBS

Benchmark: JOBS

$Neurons$	$Model$	$Time(sec)$
1	$CDANN(N = 1, \lambda = 10^5)$	13.17
2	$CDANN(N = 2, \lambda = 10^5)$	12.62
4	$CDANN(N = 4, \lambda = 10^5)$	14.68
8	$CDANN(N = 8, \lambda = 10^5)$	13.82
16	$CDANN(N = 16, \lambda = 10^5)$	14.68
32	$CDANN(N = 32, \lambda = 10^5)$	15.47
64	$CDANN(N = 64, \lambda = 10^5)$	17.15
128	$CDANN(N = 128, \lambda = 10^5)$	19.36
256	$CDANN(N = 256, \lambda = 10^5)$	21.97
512	$CDANN(N = 512, \lambda = 10^5)$	23.96

Table 5.12: Computational time for $CDANN$ model with different Number of Neurons on JOBS

Benchmark: JOBS

$Neurons$	$Model$	$Time(sec)$
1	$CeCA(N = 1, \lambda = 10^5)$	22.60
2	$CeCA(N = 2, \lambda = 10^5)$	22.32
4	$CeCA(N = 4, \lambda = 10^5)$	25.03
8	$CeCA(N = 8, \lambda = 10^5)$	24.67
16	$CeCA(N = 16, \lambda = 10^5)$	29.48
32	$CeCA(N = 32, \lambda = 10^5)$	33.09
64	$CeCA(N = 64, \lambda = 10^5)$	42.53
128	$CeCA(N = 128, \lambda = 10^5)$	45.25
256	$CeCA(N = 256, \lambda = 10^5)$	62.74
512	$CeCA(N = 512, \lambda = 10^5)$	107.56

Table 5.13: Computational time for $CeCA$ model with different Number of Neurons on JOBS

Benchmark: JOBS

$Neurons$	$Model$	$Time(sec)$
1	$CgCA(N = 1, \lambda = 10^5)$	24.20
2	$CgCA(N = 2, \lambda = 10^5)$	22.28
4	$CgCA(N = 4, \lambda = 10^5)$	22.91
8	$CgCA(N = 8, \lambda = 10^5)$	26.59
16	$CgCA(N = 16, \lambda = 10^5)$	29.84
32	$CgCA(N = 32, \lambda = 10^5)$	31.59
64	$CgCA(N = 64, \lambda = 10^5)$	33.81
128	$CgCA(N = 128, \lambda = 10^5)$	58.70
256	$CgCA(N = 256, \lambda = 10^5)$	435.65
512	$CgCA(N = 512, \lambda = 10^5)$	2,141.51

Table 5.14: Computational time for $CgCA$ model with different Number of Neurons on JOBS

5.8 Result Discussion

At first, one would expect the performance of the CTS model — a model that relates the treated and control streams but did not share weights– to have performed better than the other models that shared weights ($CDANN$, $CeCA$, and $CgCA$) as reported in [55] from domain adaptation. However, based on the results, the models with shared weights outperformed CTS on all metrics on IHDP, and JOBS datasets. Except for the NEWS dataset where CTS outperformed them ($CDANN$, $CeCA$, and $CgCA$) on two metrics (ITE_ε and ATE_ε) and one other metric (ATE_ε) on the TWINS dataset. Deep learning models are known to be robust to feature dimensions of a datasets, however, one possible explanation regarding the difference in performance between the shared and the unshared methods as observed from the results is the feature dimensions. The CTS model appeared to have performed better on datasets with relatively high dimensions. For example, out of the four datasets, the NEWS dataset has the highest dimensions (3,477), then followed by the TWINS dataset with 46 features. The number of features are higher when compared to IHDP dataset which has 25 features, and JOBS with only 8 features. Another important reason that could potentially be associated with relatively worse performance of CTS (the unshared weight method) overall is the sample size. Very often, in causal inference, the total number of treated subjects are far less than the number of control. Splitting the dataset into treated and the control for the two streams without direct interaction between the weights could hinder an effective learning particularly on the treated stream. More data would provide more insight during training which would affect the quality of the learning model. In addition, the two regularizations were possibly not sufficient to account for the lack of direct interaction between the treated and the control weights.
Therefore, it is safer to say that methods which learnt invariant features performs better relative to CTS model which learnt features separately in this case. One could possibly argue that only one method based on the unshared weights was considered and thus our findings require a careful generalization which is plausible. Such argument is valid and is considered as part of the limitations of our work to be discussed

in chapter 6. It was also found that models based on weights sharing estimated causal parameters quite well with a single neuron in virtually all the benchmarks across all the metrics. This observation was more pronounced on TWINS and JOBS datasets. With a single neuron, we observed however that CTS performed terribly bad in all benchmarks across all the metrics. We are once more of the belief that it has to do with the architecture, which has two streams, one for encoding each group, therefore, using only a single neuron with no interaction (shared weights) was not enough. As evidenced, the model sharply improved when the number of neurons increased to more than one (that is, at least two neurons, in which case, each stream has at least a neuron that encodes it).
We have realized very good estimations of some causal parameters from the different models deployed for causality. Of course there are cases where empirically, metrics improved with domain adaptation as shown in the results tables in the previous sections. Some of the scores superseded the performance of the state-of-the-art. However, we have equally observed that amongst the best scores from our models, some of them were achieved with no domain adaptation i.e, when $\lambda = 0$. There are also cases where using domain adaptation worsened the scores. But the fact that there are cases where models with no domain adaptation outperformed the state of the art has made us not to conclude that those performances were occasioned by the domain adaptation. For example, the CR model with no domain adaptation produced the best ITE_{ε} on $NEWS$ as shown in table 5.2. The CTS estimated the best P_{risk} with $\lambda = 0$ on $JOBS$ see table 5.3. Moreover, the best ATE_{ε}, and $PEHE$ on IHDP dataset were estimated by $CDANN$ with no domain adaptation as shown in table 5.1. Therefore, our results do not support to completely associate the performances of these models on the specialized regularizations, training procedure etc used in aligning the treated and the control proposed in [52], [53], [54], and [55]. If the alignment mechanisms reported in [52], [53], [54], and [55] from domain adaptation are effective for causality, we should have observed consistently a decline in performance when no domain adaptation is used ($\lambda = 0$) and a significant improvement with domain adaptation (i.e, showing that the balancing mechanism is effective). It is important to note that out of all the significant test conducted in section 5.6, only two of the improvements could be associated statistically with domain adaptation regularization. These are: the ITE_{ϵ} score estimated by the $CgCA$ on IHDP which is produced with $\lambda = 1$, and the ATT_{ϵ} score on JOBS estimated by $CeCA$ with $\lambda = 1$. In the three other cases, the test conducted have shown significance but the best scores are produced with $\lambda = 0$ suggesting that using DA regularization could have affected (negatively by reducing) the quality of the estimates in this case. Whereas, the other five cases showed no difference between the scores produced by models with domain adaptation and without domain adaptation. Alluding that in 8 out of the 10 tests in section 5.6, the baseline models (models with no DA) have produced better scores than or at least as good as models with DA. Overall, what the results suggest is that deep learning models could fairly be able learn counterfactual prediction without the need for many specialized often complicated models. Throughout the evaluation, the Euclidean and the non-Euclidean (geodesic) have shown similar behaviour and performances on all the benchmarks. In few occasions, one fluctuates slightly differently, or sometimes offered a slightly better performance over the other. Nevertheless, the results support a fairly similar performance and behaviour between them overall. Again, this is in contrast to the existing findings

in domain adaptation where [52] showed geodesic distance offered improvement over Euclidean distance [53] for domain adaptation. Based on the results, it would be difficult to say one of the methods is better than the other. In our opinion, it could be that the spaces in which the datasets lie are not very complex and thus, renders using non-Euclidean distance less relevant. That is, the Euclidean distance is enough to effectively compute the correlation alignment. Another possible reason could be, the CR model is robust to the regularizations of the two distances.

5.9 Comparison against the State-of-the- art

We compare our results with baselines which include OLS, K-Nearest Neighbour (KNN) [171]. Others are Doubly Robust (D ROBUST) [31], Bayesian Additive Regression Tree (BART) [157], Random Forest(R.FOREST) [149], and Causal Forest (C.FOREST) [172]. The state-of-the-art compared with include: Balancing Neural Network (BNN-2-2) [41], Treatment Agnostic Representation Network (TARNET) [45], Counterfactual Regression with Wassertein distance (CFR-WASS) [45], Multitask Gaussian Process (CMGP), [49], Generative Adversarial nets for Inference of Individualized Treatment Effect (GANITE) [50], and local Similarity Preserved Individual Treatment Effect (SITE), [51], and Perfect Match (PM) [150]. We present our results together with baselines and the state-of-the-art in Table 5.15. On IHDP, the results of R.Forest, C.Forest, and BART came from [50], while that of D ROBUST, and KNN are as reported in [45]. The results reported on JOBS for C.Forest, R.Forest, and BART are from [41]. Whereas that of KNN is reported in [45]. On NEWS, the results of R.Forest, C.Forest, and KNN, are as reported [150], while D ROBUST, and BART are those reported in [45]. The AUC score of KNN on TWINS is from [51], while its $PEHE$ score along with OLS, BART R.FOREST, C.FOREST, BNN-2-2, TARNET, CFR-WASS, GANITE, and CMPG are as reported in [50]. The results of BNN-2-2 are from [41];CFR-WASS from [45] except for NEWS which were obtained from [150]. Results from TARNET are as reported in [45] but for NEWS which are obtained from [150]. GANITE comes from [50] except for the NEWS which is reported in [150]. SITE is as published in [51] while CEVAE is from [155]. All other results come directly from the original authors.

Method	IHDP			JOBS		NEWS			TWINS		
	ITE_ε	ATE_ε	$PEHE$	ATT	P_{risk}	ITE_ε	ATE_ε	$PEHE$	ATE_ε	$PEHE$	AUC
OLS	–	–	–	–	–	3.1±0.2	0.2 ± 0.0	3.3 ± 0.2	–	0.318 ± 0.007	–
D ROBUST	3.0 ±0.1	0.2 ± 0.0	5.7 ± 0.3	–	–	3.1±0.2	0.2 ± 0.0	3.3 ± 0.2	–	–	–
KNN	–	0.79 ± 0.05	4.1 ± 0.2	0.13 ± 0.05	0.26 ± 0.0	–	–	18.14 ± 1.64	–	0.345 ± 0.07	0.49 ± 0.012
BART	–	0.34±0.02	2.3 ± 0.1	0.08 ± 0.03	0.25 ± 0.0	5.8±0.2	0.2 ± 0.0	3.2 ± 0.2	–	0.338 ± 0.016	–
R FOREST	–	0.96±0.06	6.6 ± 0.3	0.09 ± 0.04	0.28 ± 0.0	–	–	17.39 ± 1.24	–	0.321± 0.005	–
C FOREST	–	0.40 ±0.03	3.8 ± 0.2	0.07 ± 0.03	0.20 ± 0.02	–	–	17.59 ± 1.63	–	0.316± 0.011	–
BNN-2-2	1.7±0.0	0.3 ± 0.0	1.6 ± 0.1	–	0.24 ± 0.02	2.0±0.0	0.3 ± 0.0	2.0 ± 0.1	–	0.321± 0.018	–
TARNET	–	0.28 ± 0.01	0.95 ± 0.0	–	0.21 ± 0.01	–	–	17.17 ± 1.25	–	0.315 ± 0.03	–
CFR-WASS	–	0.27± 0.01	0.76 ± 0.0	–	0.21 ± 0.01	–	–	16.93 ± 1.12	–	0.313± 0.008	–
SITE	–	–	0.656 ± 0.108	–	0.219 ± 0.009	–	–	–	–	–	**0.85±0.006***
GANITE	–	–	2.4 ± 0.4	–	**0.14 ± 0.01***	–	–	18.2 ± 1.66	–	**0.297 ± 0.016***	–
CEVAE	–	0.46±0.02	2.6 ± 0.1	0.03 ± 0.01	0.26±0.0	–	–	–	–	–	0.82 ± 0.0
CMPG	–	–	0.76 ± 0.001	–	–	–	0.319± 0.08	–			
PM	–	–	0.84 ± 0.61	–	–	–	–	–			
CR	–	0.44 ± 0.00	1.22 ± 0.03	0.02 ± 0.00	0.23 ± 0.02	–	0.2 ± 0.03	**1.70 ± 0.09**	0.03 ± 0.00	–	0.74 ± 0.00
	–	($N = 128$)	($N = 1$)	($N = 32$)	($N = 256$)	($N = 256$)	($N = 1$)	($N = 32$)	($N = 256$)	($N = 1$)	($N = 32$)
CR-L	–	0.85 ± 0.06	5.72 ± 0.25	0.02 ± 0.00	1.00 ± 0.00	–	0.24 ± 0.04	3.37 ± 0.18	–	–	–
	–	($N = 128$)	($N = 1$)	($N = 32$)	($N = 256$)	($N = 256$)	–	($N = 32$)	($N = 256$)	($N = 256$)	
CeCA	1.318 ±0.014	0.44 ±0.00	1.22 ±0.03	**0.01 ± 0.00**	0.228 ± 0.021	1.67 ±0.07	**0.16 ± 0.02**	1.88 ± 0.08	0.03 ±0.002	0.33 ± 0.00	0.75 ± 0.001
	($N = 128, \lambda = 0$)	($N = 128, \lambda = 0$)	($N = 128, \lambda = 0$)	($N = 1, \lambda = 10^3$)	($N = 32, \lambda = 0$)	($N = 256, \lambda = 0$)	($N = 256, \lambda = 10^5$)	($N = 256, \lambda = 10^5$)	($N = 8, \lambda = 100$)	($N = 1, \lambda = 0$)	($N = 4, \lambda = 100$)
CgCA	**1.27 ±0.02**	0.37 ± 0.02	1.22 ± 0.05	**0.01 ± 0.00**	0.23 ± 0.02	1.67 ±0.08	0.183 ± 0.03	1.71± 0.09	0.029 ± 0.01	0.33 ±0.00	0.75 ± 0.001
	($N = 128, \lambda = 1$)	($N = 128, \lambda = 10$)	($N = 128, \lambda = 1$)	($N = 1, \lambda = 100$)	($N = 32, \lambda = 0$)	($N = 256, \lambda = 10^3$)	($N = 256, \lambda = 10^3$)	($N = 256, \lambda = 10^3$)	($N = 8, \lambda = 1$)	($N = 1, \lambda = 0$)	($N = 4, \lambda = 1$)
CTS	1.54 ±0.04	0.34 ± 0.02	2.29 ± 0.10	0.02 ± 0.01	0.18 ± 0.04	**1.57 ±0.02**	**0.16 ± 0.02**	1.80 ± 0.05	**0.02 ± 0.01**	0.36 ±0.01	0.74 ± 0.00
	($N = 128, \lambda_w = \lambda_m = 0$)	($N = 128, \lambda_w = \lambda_m = 100$)	($N = 128, \lambda_w = \lambda_m = 100$)	($N = 2, \lambda_w = \lambda_m = 10^4$)	($N = 256, \lambda_w = \lambda_m = 0$)	($N = 256, \lambda_w = \lambda_m = 100$)	($N = 256, \lambda_w = \lambda_m = 100$)	($N = 256, \lambda_w = \lambda_m = 100$)	($N = 256, \lambda_w = \lambda_m = 1$)	($N = 4, \lambda_w = \lambda_m = 1$)	($N = 256, \lambda_w = \lambda_m = 1$)
CDANN	1.88 ±0.07	**0.148 ± 0.01**	**0.52 ± 0.02**	0.03 ± 0.00	0.24 ± 0.03	2.38 ±0.16	0.28 ± 0.04	2.69 ± 0.18	0.04 ± 0.00	0.33±0.00	0.73 ± 0.001
	($N = 1, \lambda = 0$)	($N = 1, \lambda = 0$)	($N = 1, \lambda = 0$)	($N = 8, \lambda = 10$)	($N = 32, \lambda = 1$)	($N = 16, \lambda = 100$)	($N = 16, \lambda = 0$)	($N = 16, \lambda = 100$)	($N = 4, \lambda = 0$)	($N = 1, \lambda = 0$)	($N = 1, \lambda = 0$)

Table 5.15: State of the art results for each dataset, and results from $CeCA$, $CgCA$, CTS, and $CDANN$ alongside with their CR and linear variants. All values are reported as mean ± standard deviation

The IHDP results show that the best ITE_ε (1.27±0.02) is estimated by $CgCA$, and it outperformed all other state-of-the-art. Our model $CDANN$ produced the best ATE_ε (0.48 ± 0.01) among all other models as seen in Table 5.15. The same $CDANN$ produced the best $PEHE$ score of 0.52±0.02 that outperformed even the most recent state-of-the-art scores. This score is particularly important, being that it is one very popular metric often reported in causality, moreover, $CDANN$ achieved this score with no domain adaptation.

On JOBS, the two correlation alignment methods $CgCA$ and $CeCA$ produced an ATT_ε score of 0.01 ±0.00 each, which is the lowest hence the best ATT_ε score reported. While the best P_{risk} was achieved by CEVAE–0.14±0.01 as shown in Table 5.15. CTS model however produced the second best P_{risk} score.

The CTS model on the NEWS benchmark produced respectively 0.16±0.02 and 1.52±0.02, as the best ATE_ε and ITE_ε estimates amongst both the baselines and the state of the art as shown in Table 5.15. Moreover, our CR model produced a $PEHE$ score of 1.70 which outperformed all scores in the Table. $CgCA$ also produced 1.71, a score similar to the one estimated by CR.

The best (lowest) ATE_ε score on TWINS is 0.02 ±0.01 (See Table 5.15) and was estimated by CTS. Whereas the best PEHE on this benchmark was produced by $CeCA$, $CgCA$, and $CDANN$. Whereas, CEVAE produced the best AUC score of 0.82 ±0.0 and thus outperformed all other models as shown in the Table.

Chapter 6

Conclusion

Government policy provides intervention to its citizens as a social investment program, businesses offer incentives to boost sales, social researchers intervene to study topical questions, and medical experts administer new drugs to patients. These are causal not correlation problems, and estimating the effect of these interventions has been a major challenge. An ideal approach would require conducting experiments which are often time consuming, unethical, extremely expensive, or even impossible. If we are to rely only on experiments, many interesting research and business questions would remain unanswered. Nowadays, automated systems generate massive amounts of data called observational data. Such data could be found in health electronic records, retail, government databases among many others. Using observational data with no need for experiments becomes a feasible alternative to answer many questions that would not be possible with experiments resulting in observational studies. However, an observational study contains a systematic bias, posing a great challenge for causal inference task. The systematic bias between the treated and the control subjects has to be "corrected" through some balancing techniques. To address this problem, in the statistical literature, methods based on *Regression Adjustment* [68], *Propensity Score* [21], [22] and *Instrumental Variables* [34], [35] are proposed. Other methods include Structural Equation Models (SEMs) [173], [174].

Researchers in recent times have suggested that aligning (balancing) the distributions of the treated and the control is an effective technique for predicting counterfactuals [51], [41]. For that reason, ML/Deep Learning models have recently been deployed for Causality with different approaches such as those proposed in [41], [42], [43], [44], [45], [46], [51], [50], and [175]. Many ML and Deep Learning models deployed for causality are specialized and are intended to mitigate the bias in the data through balancing the treated and the control during training. But the existing Deep Learning models often involve complicated architectures.

6.1 Contributions

Our work exploits the idea proposed in [41] which connects causality and domain adaptation. Knowing that on the one hand, the main challenge in causality with observational study is the bias between the distributions of the treated and the control, and on the other hand, domain adaptation methods align different distributions for effective prediction. It sounds a feasible idea to bring some DA methods as distri-

bution alignment mechanisms for aligning the treated and the control in causality. Going by this line of research, in this thesis, we brought four deep learning methods pioneered in DA into Causality namely: methods based on correlation alignment [52], [53], adversarial training [54], and parallel two streams [55], reflecting diverse (methods based on shared weights, and the other based on unshared weights) techniques by modifying them to fit within the causality framework. But we initially built around these methods simple baseline NN models in each case which are optimized and then we incorporate their DA components as an additional loss to the baseline models. Following the potential outcome framework [6], assuming ignorability, independence [24], and SUTVA [105] are satisfied, we formulate each one of the methods in DA as a causal inference problem. We were interested in understanding how effective the simple baseline models with no any form of domain adaptation (often complicated architectures, customize regularizations) are for causality. And what improvement does incorporating domain adaptation brings in causal parameter estimation. We also seek to understand if there is any superiority in performance between these classes (shared and unshared weights) or methods for this task. Each of the four models were evaluated using four most widely used causality benchmarks name : IHDP, NEWS, JOBS, and TWINS datasets.

Results produced by our models outperformed the state of the art on some metrics. For example, on IHDP dataset, our ITE_ε, ATE_ε, and $PEHE$ estimates were the best (see Table 5.15). Moreover, our models produced the best ATT_ε score on JOBS. On the NEWS dataset, we had the best ITE_ε, ATE_ε, and $PEHE$ scores amongst all other models (other baselines and state of the art). Whereas on TWINS, one of our models produced the best ATE_ε (See Table 5.15). As discussed in chapter 5 and from the test of significance in section 5.6, there were only two cases out of the ten cases (best results) presented in section 5.6 where models with domain adaptation regularization produced better results over the same models with no DA. And their performance could statistically be associated with DA regularization. Whereas, in three of the ten cases, using DA regularization worsened the metric scores as seen in Tables 5.1, 5.3, 5.2, and 5.4. In fact, for the three cases, the results obtained with no DA regularization outperformed all state-of-the-art. For instance, the CR model with no domain adaptation estimated the best $PEHE$ on $NEWS$ dataset (see Table 5.15). The CTS model estimated the best P_{risk} with $\lambda = 0$ on $JOBS$ as shown in Table 5.3. In addition, the best ATE_ε, and $PEHE$ on IHDP dataset were estimated by $CDANN$ with no domain adaptation 5.1. In some other cases, the results produced are indifferent or insensitive to the regularization. These observations cut across all models and benchmarks. For example, the best three scores under NEWS dataset outperformed state-of-the-art, yet, all the models that produced them showed no difference between a model with DA and without DA (See test of significance in section 5.6). The fact that we were able to achieve results outperforming the state-of- the-art with no DA, and the many cases where DA worsened the quality of the parameters indicates deep learning models could estimate causal parameters fairly well without the need for DA. This suggests that with hyper parameter tuning, one could build a deep learning model that is capable of effectively predicting counterfactuals without necessarily using domain adaptation regularization or specialized training procedure which often have complex architectures. This observation is further confirmed in section 5.6 where out of

the 10 best metrics produced by our models, we run a significant test between the models that produced these scores but only two cases are statistically significant. The remaining eight showed no difference between the outcomes of the models with and without DA. Inferring that the effect of domain adaptation is insignificant.

It is also evident from the results that models with shared weights (CR,$CeCA$, $CgCA$, and $CDANN$) produced fairly better results on most of the datasets compared to the performance of the CTS model (model with unshared weights). We could therefore say that methods with shared weights have an edge over unshared method evaluated in this thesis. In addition, with a single neuron, models based on weights sharing produced relatively better causal estimates overall compared to the CTS model across all benchmarks and metrics.

Contrary to [52] from classical DA which reported that for covariance alignments methods, using geodesic distance instead of Euclidean distance improves performance, our results could not support such a claim. What our results showed was that both models behaved fairly very similarly overall. There were few cases where one method fluctuates slightly differently or marginally performed better than the other, even with that however, overall, we cannot conclude one is better than the other. The results of these models are very close to assert one regularization is superior over the other. The similarity across the covariance alignment models suggests some robustness to distance measures. Our work also reveals that less complicated models with linearity alone are sufficient to recover some causal parameters but on the population level.

6.2 Limitations and future work

The scope of our research is limited as it only considers four domain adaptation methods [52], [53], [54], and [55] from two classes of DA (methods with shared weights and unshared weights). This has limited the generalizability of our findings. More insight could be uncovered if additional domain adaptation techniques from diverse classes are deployed. For instance, the conclusion on which class of domain adaptation (shared approaches, and the unshared) performs better for causality would have been more general if more methods from unshared weights were considered. Four benchmarks were used for evaluation, even though they are amongst the most popularly used for causality, there is a need to evaluate our models on more datasets. Doing so would provide a strong basis to support or contradict some researchers claim (in recent times) that aligning the distributions of the treated and the control is an effective technique for predicting counterfactuals in machine learning and deep learning models [51],[41].

It would also be of interest to do more extensive hyperparameter studies in the future to find out how sensitive our models are to different hyperparameters, could we achieve even better estimates with no domain adaptation regularization? One major challenge faced during the development is the hyper-parameter tuning viz-a viz the running time particularly with benchmarks having larger realizations such as IHDP. This typically takes more than 24 hours for some models to be evaluated. For every hyperparameter adjustment, one has to wait for those number of hours to see the effect.

Appendix A

IHDP Results

Model	Benchmark: IHDP		
	ITE_ε	ATE_ε	$PEHE$
CTS (N=128,$\lambda_m = \lambda_w = 0$)	1.544 ±0.0392	0.372 ± 0.019	2.291 ±0.098
CTS (N=128,$\lambda_m = \lambda_w = 1$)	1.542 ±0.039	0.364 ± 0.019	2.286± 0.098
CTS (N=128,$\lambda_m = \lambda_w = 10$)	1.542 ±0.039	0.360 ± 0.019	2.29± 0.098
CTS (N=128,$\lambda_m = \lambda_w = 100$)	1.545 ±0.039	0.344 ± 0.016	2.289± 0.098
CTS (N=128,$\lambda_m = \lambda_w = 1000$)	1.541 ±0.039	0.357 ± 0.019	2.294± 0.098
CTS (N=128,$\lambda_m = \lambda_w = 10000$)	1.546 ±0.039	0.365 ± 0.017	2.298± 0.098
CTS (N=128,$\lambda_m = \lambda_w = 100000$)	1.548 ±0.039	0.373 ± 0.019	2.30± 0.098
$CDANN$ ($N = 1, \lambda = 0$)	1.875 ± 0.066	0.148 ± 0.005	0.524 ±0.021
$CDANN$ ($N = 1, \lambda = 1$)	2.30 ± 0.124	0.181 ± 0.006	0.677 ±0.027
$CDANN$ ($N = 1, \lambda = 10$)	2.350 ± 0.093	0.250 ± 0.01	1.20 ±0.055
$CDANN$ ($N = 1, \lambda = 100$)	2.33 ± 0.104	0.310 ± 0.018	1.753 ±0.09
$CDANN$ ($N = 1, \lambda = 1000$)	2.637 ± 0.106	0.500 ± 0.03	3.467 ±0.156
$CDANN$ ($N = 1, \lambda = 10000$)	2.93± 0.112	0.646 ±0.037	5.177 ±0.222
$CDANN$ ($N = 1, \lambda = 100000$)	2.90 ± 0.102	0.652 ± 0.037	5.412 ±0.23

Table A.1: Results of CTS, $CDANN$ models and their variants on IHDP

Model	Benchmark: IHDP		
	ITE_ε	ATE_ε	$PEHE$
$CR \quad (N = 128, \lambda = 0)$	1.318 ±0.014	0.435 ±0.02	1.216 ± 0.034
$CgCA \quad (N = 128, \lambda = 1)$	1.22 ± 0.021	0.390 ±0.021	1.221 ± 0.047
$CgCA \quad (N = 128, \lambda = 10)$	1.387 ±0.028	0.370 ±0.017	1.660 ± 0.067
$CgCA \quad (N = 128, \lambda = 100)$	1.45 ± 0.031	0.406 ±0.02	1.87 ± 0.076
$CgCA \quad (N = 128, \lambda = 1000)$	1.466 ±0.032	0.432 ±0.024	1.92 ± 0.081
$CgCA \quad (N = 128, \lambda = 10000)$	1.44 ±0.028	0.40 ±0.02	1.89 ± 0.077
$CgCA \quad (N = 128, \lambda = 100000)$	1.467 ± 0.031	0.419 ±0.02	1.90 ± 0.075
$CeCA \quad (N = 128, \lambda = 1)$	1.34 ± 0.017	0.460 ±0.025	1.245 ± 0.039
$CeCA \quad (N = 128, \lambda = 10)$	1.35 ±0.018	0.490 ±0.028	1.250 ± 0.040
$CeCA \quad (N = 128, \lambda = 100)$	1.335 ±0.018	0.466 ±0.024	1.240 ± 0.037
$CeCA \quad (N = 128, \lambda = 1000)$	1.333 ±0.017	0.450 ±0.033	1.262 ± 0.044
$CeCA \quad (N = 128, \lambda = 10000)$	1.342 ±0.018	0.470 ±0.023	1.252± 0.039
$CeCA \quad (N = 128, \lambda = 100000)$	1.333 ±0.020	0.502 ±0.042	1.281 ± 0.052

Table A.2: Results of correlation alignment models and their variants on IHDP

Appendix B

NEWS Results

Model	Benchmark: NEWS		
	ITE_ε	ATE_ε	$PEHE$
CTS (N=256,$\lambda_m = \lambda_w = 0$)	1.578 ±0.020	0.207 ± 0.024	1.803 ±0.049
CTS (N=256,$\lambda_m = \lambda_w = 1$)	1.579 ±0.022	0.189 ± 0.018	1.80 ±0.050
CTS (N=256,$\lambda_m = \lambda_w = 10$)	1.594±0.024	0.271 ± 0.0277	1.826 ±0.052
CTS (N=256,$\lambda_m = \lambda_w = 100$)	1.574 ±0.021	0.159 ± 0.017	1.795 ±0.052
CTS (N=256,$\lambda_m = \lambda_w = 1000$)	1.573 ±0.020	0.194 ± 0.022	1.80±0.048
CTS (N=256,$\lambda_m = \lambda_w = 10000$)	1.60 ±0.023	0.220 ± 0.028	1.82 ± 0.052
CTS (N=256,$\lambda_m = \lambda_w = 100000$)	1.60 ±0.022	0.173 ± 0.021	1.81 ± 0.053
$CDANN$ ($N = 16, \lambda = 0$)	3.160 ± 0.114	0.327 ± 0.040	2.986 ±0.112
$CDANN$ ($N = 16, \lambda = 1$)	3.91 ± 0.40	0.724 ±0.074	3.60 ±0.194
$CDANN$ ($N = 16, \lambda = 10$)	3.30 ± 0.167	0.776 ±0.096	3.47 ±0.225
$CDANN$ ($N = 16, \lambda = 100$)	2.383 ±0.157	0.775 ±0.090	2.686 ±0.176
$CDANN$ ($N = 16, \lambda = 1000$)	2.564 ± 0.12	0.80 ±0.10	3.388 ±0.192
$CDANN$ ($N = 16, \lambda = 10000$)	2.44 ± 0.090	0.750 ±0.086	3.43 ±0.188
$CDANN$ ($N = 16, \lambda = 100000$)	2.38 ± 0.076	0.812 ±0.089	3.46 ±0.190

Table B.1: Results of CTS, $CDANN$ models and their variants on NEWS

Model	Benchmark: NEWS		
	ITE_ε	ATE_ε	$PEHE$
$CR \quad (N = 256, \lambda = 0)$	1.670 ±0.068	0.203 ± 0.025	1.70 ± 0.089
$CgCA \quad (N = 256, \lambda = 1)$	1.713 ±0.087	0.25 ±0.038	1.738 ± 0.11
$CgCA \quad (N = 256, \lambda = 10)$	1.71±0.10	0.213 ±0.032	1.73 ± 0.11
$CgCA \quad (N = 256, \lambda = 100)$	1.683 ±0.080	0.22 ±0.029	1.705 ± 0.10
$CgCA \quad (N = 256, \lambda = 1000)$	1.665 ±0.075	0.183 ±0.027	1.705 ± 0.092
$CgCA \quad (N = 256, \lambda = 10000)$	1.562 ±0.026	0.21 ±0.021	1.658 ± 0.068
$CgCA \quad (N = 256, \lambda = 100000)$	1.574 ±0.029	0.307±0.036	**1.651 ± 0.074**
$CeCA \quad (N = 256, \lambda = 1)$	1.680±0.062	0.234 ±0.035	1.711 ± 0.085
$CeCA \quad (N = 256, \lambda = 10)$	1.69±0.082	0.216 ±0.0289	1.722 ± 0.103
$CeCA \quad (N = 256, \lambda = 100)$	1.70±0.087	0.225 ±0.029	1.722 ± 0.104
$CeCA \quad (N = 256, \lambda = 1000)$	1.690±0.089	0.191 ±0.033	1.720 ± 0.103
$CeCA \quad (N = 256, \lambda = 10000)$	1.70±0.055	0.20 ±0.026	1.732 ± 0.078
$CeCA \quad (N = 256, \lambda = 100000)$	1.732±0.039	0.164 ±0.019	1.883 ± 0.076

Table B.2: Results of correlation alignment models and their variants on NEWS

Appendix C

JOBS Results

Model	Benchmark: JOBS	
	ATT_{ε}	P_{Risk}
CTS ($\lambda_m = \lambda_w = 0$,N=2)	0.046±0.009	0.263 ± 0.021
CTS ($\lambda_m = \lambda_w = 1$,N=2)	0.0838±0.0174	0.268 ± 0.020
CTS ($\lambda_m = \lambda_w = 10$,N=2)	0.090 ±0.0224	0.314 ± 0.023
CTS ($\lambda_m = \lambda_w = 100$,N=2)	0.103 ±0.034	0.362 ± 0.019
CTS ($\lambda_m = \lambda_w = 1000$,N=2)	0.0616 ±0.016	0.313 ± 0.011
CTS ($\lambda_m = \lambda_w = 10000$,N=2)	0.022 ±0.006	0.328 ± 0.016
CTS ($\lambda_m = \lambda_w = 100000$,N=2)	0.061 ±0.018	0.357 ± 0.040
CTS ($\lambda_m = \lambda_w = 0$,N=256)	0.046±0.009	0.263 ± 0.021
CTS ($\lambda_m = \lambda_w = 1$,N=256)	0.093±0.0175	0.294 ± 0.020
CTS ($\lambda_m = \lambda_w = 10$,N=256)	0.114 ±0.013	0.311 ± 0.0125
CTS ($\lambda_m = \lambda_w = 100$,N=256)	0.109 ±0.0175	0.328 ± 0.014
CTS ($\lambda_m = \lambda_w = 1000$,N=256)	0.148 ±0.020	0.322 ± 0.045
CTS ($\lambda_m = \lambda_w = 10000$,N=256)	0.132 ±0.035	0.312 ± 0.040
CTS ($\lambda_m = \lambda_w = 100000$,N=256)	0.098 ±0.016	0.357 ± 0.040

Table C.1: Results of CTS, model and its variants on JOBS

Model	Benchmark: JOBS	
	ATT_ε	P_{Risk}
$CDANN \quad (\lambda = 0, N = 8)$	0.069 ± 0.010	0.311 ± 0.077
$CDANN \quad (\lambda = 1, N = 8)$	0.072±0.0103	0.261 ±0.009
$CDANN \quad (\lambda = 10, N = 8)$	0.027±0.004	0.292 ±0.010
$CDANN \quad (\lambda = 100, N = 8)$	0.031±0.006	0.304 ±0.008
$CDANN \quad (\lambda = 1000, N = 8)$	0.0488±0.010	0.315 ±0.037
$CDANN \quad (\lambda = 10000, N = 8)$	0.040±0.0064	0.309 ±0.008
$CDANN \quad (\lambda = 100000, N = 8)$	0.060±0.004	0.308 ±0.008
$CDANN \quad (\lambda = 0, N = 32)$	0.0657 ± 0.011	0.416 ± 0.0977
$CDANN \quad (\lambda = 1, N = 32)$	0.067±0.011	0.237 ±0.0286
$CDANN \quad (\lambda = 10, N = 32)$	0.053±0.013	0.303 ±0.008
$CDANN \quad (\lambda = 100, N = 32)$	0.049±0.010	0.308 ±0.007
$CDANN \quad (\lambda = 1000, N = 32)$	0.051±0.006	0.305 ±0.006
$CDANN \quad (\lambda = 10000, N = 32)$	0.066±0.01	0.313 ±0.003
$CDANN \quad (\lambda = 100000, N = 32)$	0.0468±0.007	0.309 ±0.005

Table C.2: Results of $CDANN$ model and its variants on JOBS

Model	Benchmark: JOBS	
	ATT_{ε}	P_{Risk}
$CR \quad (N = 1, \lambda = 0)$	0.012 ±0.002	0.314 ± 0.000
$CeCA \quad (N = 1, \lambda = 1)$	0.0152± 0.002	0.314 ±0.000
$CeCA \quad (N = 1, \lambda = 10)$	0.009± 0.002	0.314 ±0.000
$CeCA \quad (N = 1, \lambda = 100)$	0.009± 0.001	0.314 ±0.000
$CeCA \quad (N = 1, \lambda = 1000)$	0.0048± 0.001	0.314 ±0.000
$CeCA \quad (N = 1, \lambda = 10000)$	0.0087± 0.004	0.314 ±0.000
$CeCA \quad (N = 1, \lambda = 100000)$	0.0085± 0.002	0.314 ±0.000
$CR \quad (N = 32, \lambda = 0)$	0.101 ±0.021	0.228 ± 0.021
$CeCA \quad (N = 32, \lambda = 1)$	0.084± 0.015	0.285 ±0.016
$CeCA \quad (N = 32, \lambda = 10)$	0.074± 0.017	0.287 ±0.018
$CeCA \quad (N = 32, \lambda = 100)$	0.0815± 0.018	0.275±0.018
$CeCA \quad (N = 32, \lambda = 1000)$	0.085± 0.008	0.372 ±0.083
$CeCA \quad (N = 32, \lambda = 10000)$	0.0558± 0.006	0.367 ±0.051
$CeCA \quad (N = 32, \lambda = 100000)$	0.036± 0.01	0.322 ±0.013
$CgCA \quad (N = 1, \lambda = 1)$	0.009 ± 0.002	0.314 ±0.000
$CgCA \quad (N = 1, \lambda = 10)$	0.009 ± 0.002	0.314 ±0.000
$CgCA \quad (N = 1, \lambda = 100)$	0.006 ± 0.001	0.314 ±0.000
$CgCA \quad (N = 1, \lambda = 1000)$	0.007 ± 0.001	0.314 ±0.000
$CgCA \quad (N = 1, \lambda = 10000)$	0.007 ± 0.002	0.314 ±0.000
$CgCA \quad (N = 1, \lambda = 100000)$	0.011 ± 0.003	0.315 ±0.001
$CgCA \quad (N = 32, \lambda = 1)$	0.0723 ±0.01	0.282 ±0.01
$CgCA \quad (N = 32, \lambda = 10)$	0.073 ±0.006	0.268 ±0.009
$CgCA \quad (N = 32, \lambda = 100)$	0.077 ±0.011	0.286 ±0.008
$CgCA \quad (N = 32, \lambda = 1000)$	0.064 ±0.01	0.286 ±0.02
$CgCA \quad (N = 32, \lambda = 10000)$	0.064 ±0.013	0.278 ±0.013
$CgCA \quad (N = 32, \lambda = 100000)$	0.070 ±0.011	0.273 ±0.013

Table C.3: Results of correlation alignment models and their variants on JOBS

Appendix D

TWINS Results

Model	Benchmark: TWINS		
	ATE_{ε}	$PEHE$	AUC
CTS (N=4,$\lambda_m = \lambda_w = 0$)	0.070 ±0.022	0.399 ± 0.026	0.694 ±0.008
CTS (N=4,$\lambda_m = \lambda_w = 1$)	0.053 ±0.005	0.357 ±0.006	0.724± 0.006
CTS (N=4,$\lambda_m = \lambda_w = 10$)	0.044 ±0.003	0.348 ± 0.002	0.694 ±0.003
CTS (N=4,$\lambda_m = \lambda_w = 100$)	0.061 ±0.001	0.348 ± 0.001	0.557 ±0.005
CTS (N=4,$\lambda_m = \lambda_w = 1000$)	0.066 ±0.00	0.347 ± 0.001	0.520 ±0.004
CTS (N=4,$\lambda_m = \lambda_w = 10000$)	0.0646 ±0.001	0.346 ± 0.00	0.524 ±0.003
CTS (N=4,$\lambda_m = \lambda_w = 100000$)	0.064 ±0.00	0.346 ± 0.00	0.517 ±0.002
CTS (N=8,$\lambda_m = \lambda_w = 0$)	0.07 ±0.022	0.399 ± 0.026	0.694 ±0.008
CTS (N=8,$\lambda_m = \lambda_w = 1$)	0.042 ±0.008	0.374 ±0.009	0.708 ± 0.005
CTS (N=8,$\lambda_m = \lambda_w = 10$)	0.042 ±0.008	0.359 ±0.007	0.708 ± 0.005
CTS (N=8,$\lambda_m = \lambda_w = 100$)	0.046 ±0.011	0.358 ±0.01	0.708 ± 0.005
CTS (N=8,$\lambda_m = \lambda_w = 1000$)	0.045 ±0.004	0.355 ±0.003	0.644± 0.006
CTS (N=8,$\lambda_m = \lambda_w = 10000$)	0.0667 ±0.00	0.348 ±0.00	0.530 ± 0.003
CTS (N=8,$\lambda_m = \lambda_w = 100000$)	0.0657 ±0.001	0.347 ±0.00	0.518 ± 0.004
CTS (N=256,$\lambda_m = \lambda_w = 0$)	0.023 ±0.005	0.402 ± 0.006	0.680 ±0.005
CTS (N=256,$\lambda_m = \lambda_w = 1$)	0.022 ±0.005	0.396 ±0.007	0.674 ± 0.004
CTS (N=256,$\lambda_m = \lambda_w = 10$)	0.0382 ±0.008	0.42 ±0.01	0.680 ± 0.003
CTS (N=256,$\lambda_m = \lambda_w = 100$)	0.0319 ±0.007	0.420 ±0.009	0.682 ± 0.004
CTS (N=256,$\lambda_m = \lambda_w = 1000$)	0.0341 ±0.005	0.417 ±0.006	0.677 ± 0.008
CTS (N=256,$\lambda_m = \lambda_w = 10000$)	0.047 ±0.006	0.429 ±0.008	0.685 ± 0.002
CTS (N=256,$\lambda_m = \lambda_w = 100000$)	0.033 ±0.004	0.411 ±0.006	0.682 ± 0.005

Table D.1: Results of CTS model and its variants on TWINS

Model	Benchmark: TWINS		
	ATE_ε	$PEHE$	AUC
$CR \quad (N=1, \lambda=0)$	0.032 ±0.000	0.325 ± 0.000	0.725 ± 0.003
$CeCA \quad (N=1, \lambda=1)$	0.0334 ±0.00	0.325 ±0.003	0.723 ± 0.003
$CeCA \quad (N=1, \lambda=10)$	0.0343 ±0.001	0.325 ±0.00	0.724 ± 0.003
$CeCA \quad (N=1, \lambda=100)$	0.032 ±0.00	0.724 ±0.00	0.724 ± 0.003
$CeCA \quad (N=1, \lambda=1000)$	0.033 ±0.00	0.325 ±0.00	0.730 ± 0.00
$CeCA \quad (N=1, \lambda=10000)$	0.031 ±0.001	0.324 ±0.00	0.722 ± 0.003
$CeCA \quad (N=1, \lambda=100000)$	0.0332 ±0.00	0.325 ±0.00	0.715 ± 0.004
$CR \quad (N=4, \lambda=0)$	0.035 ±0.002	0.330± 0.001	0.74 ± 0.001
$CeCA \quad (N=4, \lambda=1)$	0.035 ±0.003	0.330± 0.001	0.743 ± 0.002
$CeCA \quad (N=4, \lambda=10)$	0.0375 ±0.002	0.332± 0.001	0.746 ± 0.00
$CeCA \quad (N=4, \lambda=100)$	0.033 ±0.001	0.328± 0.001	0.743 ± 0.002
$CeCA \quad (N=4, \lambda=1000)$	0.0343±0.002	0.328 ±0.001	0.74 ± 0.004
$CeCA \quad (N=4, \lambda=10000)$	0.038 ±0.003	0.331± 0.001	0.739 ± 0.004
$CeCA \quad (N=4, \lambda=100000)$	0.05 ±0.015	0.347± 0.016	0.725 ± 0.011
$CR \quad (N=8, \lambda=0)$	0.0315 ±0.002	0.331 ± 0.001	0.74 ± 0.006
$CeCA \quad (N=8, \lambda=1)$	0.05 ±0.015	0.347± 0.016	0.725 ± 0.011
$CeCA \quad (N=8, \lambda=10)$	0.05 ±0.015	0.347± 0.016	0.725 ± 0.011
$CeCA \quad (N=8, \lambda=100)$	0.05 ±0.015	0.347± 0.016	0.725 ± 0.011
$CeCA \quad (N=8, \lambda=1000)$	0.05 ±0.015	0.347± 0.016	0.725 ± 0.011
$CeCA \quad (N=8, \lambda=10000)$	0.05 ±0.015	0.347± 0.016	0.725 ± 0.011
$CeCA \quad (N=8, \lambda=100000)$	0.05 ±0.015	0.347± 0.016	0.725 ± 0.011
$CgCA \quad (N=1, \lambda=1)$	0.035 ±0.001	0.326 ±0.00	0.665 ± 0.021
$CgCA \quad (N=1, \lambda=10)$	0.033 ±0.003	0.325 ±0.001	0.543 ± 0.025
$CgCA \quad (N=1, \lambda=100)$	0.032 ±0.004	0.325 ±0.002	0.516 ± 0.0051
$CgCA \quad (N=1, \lambda=1000)$	0.034 ±0.00	0.325 ±0.00	0.514 ± 0.004
$CgCA \quad (N=1, \lambda=10000)$	0.038 ±0.001	0.327 ±0.00	0.513 ± 0.004
$CgCA \quad (N=1, \lambda=100000)$	0.036 ±0.001	0.327 ±0.00	0.521 ± 0.006
$CgCA \quad (N=4, \lambda=1)$	0.038 ±0.002	0.331 ±0.001	0.746 ± 0.001
$CgCA \quad (N=4, \lambda=10)$	0.0347 ±0.002	0.328 ±0.00	0.729 ± 0.007
$CgCA \quad (N=4, \lambda=100)$	0.056 ±0.012	0.344 ±0.012	0.586 ± 0.018
$CgCA \quad (N=4, \lambda=1000)$	0.056 ±0.004	0.340 ±0.003	0.535 ± 0.002
$CgCA \quad (N=4, \lambda=10000)$	0.053 ±0.003	0.337 ±0.002	0.540 ± 0.006
$CgCA \quad (N=4, \lambda=100000)$	0.071 ±0.015	0.357 ±0.019	0.534 ± 0.006
$CgCA \quad (N=8, \lambda=1)$	0.0286 ±0.002	0.328 ±0.002	0.737 ± 0.005
$CgCA \quad (N=8, \lambda=10)$	0.033 ±0.001	0.329 ±0.00	0.746 ± 0.00
$CgCA \quad (N=8, \lambda=100)$	0.032 ±0.002	0.328 ±0.00	0.728 ± 0.001
$CgCA \quad (N=8, \lambda=1000)$	0.061 ±0.001	0.343 ±0.00	0.560 ± 0.01
$CgCA \quad (N=8, \lambda=10000)$	0.057 ±0.002	0.340 ±0.001	0.566 ± 0.01
$CgCA \quad (N=8, \lambda=100000)$	0.056 ±0.002	0.341 ±0.002	0.547 ± 0.005

Table D.2: Results of correlation alignment models and their variants on TWINS

Model	Benchmark: TWINS		
	ATE_ε	$PEHE$	AUC
$CDANN \quad (\lambda = 0, N = 1)$	0.0432 ± 0.003	0.333 ± 0.002	0.753 ±0.001
$CDANN \quad (\lambda = 1, N = 1)$	0.0657 ± 0.007	0.350± 0.006	0.739 ±0.005
$CDANN \quad (\lambda = 10, N = 1)$	0.0678 ± 0.006	0.351 ± 0.005	0.713 ±0.012
$CDANN \quad (\lambda = 100, N = 1)$	0.0654 ± 0.005	0.350 ± 0.004	0.70 ±0.022
$CDANN \quad (\lambda = 1000, N = 1)$	0.072 ± 0.005	0.354 ± 0.004	0.707 ±0.008
$CDANN \quad (\lambda = 10000, N = 1)$	0.081 ± 0.008	0.363 ± 0.008	0.705 ±0.015
$CDANN \quad (\lambda = 100000, N = 1)$	0.082 ± 0.007	0.364 ± 0.007	0.705 ±0.006
$CDANN \quad (N = 4, \lambda = 0)$	0.043 ± 0.003	0.334 ±0.002	0.751 ±0.001
$CDANN \quad (N = 4, \lambda = 1)$	0.080 ± 0.006	0.363 ±0.006	0.734 ±0.005
$CDANN \quad (N = 4, \lambda = 10)$	0.075 ± 0.005	0.358 ±0.005	0.715 ±0.005
$CDANN \quad (N = 4, \lambda = 100)$	0.069 ± 0.005	0.352 ±0.004	0.713 ±0.006
$CDANN \quad (N = 4, \lambda = 1000)$	0.084± 0.005	0.365 ±0.004	0.709 ±0.006
$CDANN \quad (N = 4, \lambda = 10000)$	0.076 ± 0.004	0.357 ±0.003	0.715 ±0.008
$CDANN \quad (N = 4, \lambda = 100000)$	0.076 ± 0.005	0.359 ±0.005	0.707 ±0.01
$CDANN \quad (N = 8, \lambda = 0)$	0.044 ± 0.003	0.334 ±0.002	0.753 ±0.001
$CDANN \quad (N = 8, \lambda = 1)$	0.064 ± 0.006	0.347 ±0.005	0.743 ±0.005
$CDANN \quad (N = 8, \lambda = 10)$	0.080 ± 0.007	0.363 ±0.007	0.713 ±0.007
$CDANN \quad (N = 8, \lambda = 100)$	0.08 ± 0.005	0.362 ±0.006	0.70 ±0.01
$CDANN \quad (N = 8, \lambda = 1000)$	0.070 ± 0.003	0.352 ±0.003	0.710 ±0.007
$CDANN \quad (N = 8, \lambda = 10000)$	0.066 ± 0.005	0.350 ±0.005	0.722 ±0.009
$CDANN \quad (N = 8, \lambda = 100000)$	0.073 ± 0.004	0.354 ±0.004	0.72 ±0.005

Table D.3: Results of $CDANN$ model and its variants on TWINS

Appendix E

Additional figures for ATT_ϵ with interaction between N and $\lambda > 0$

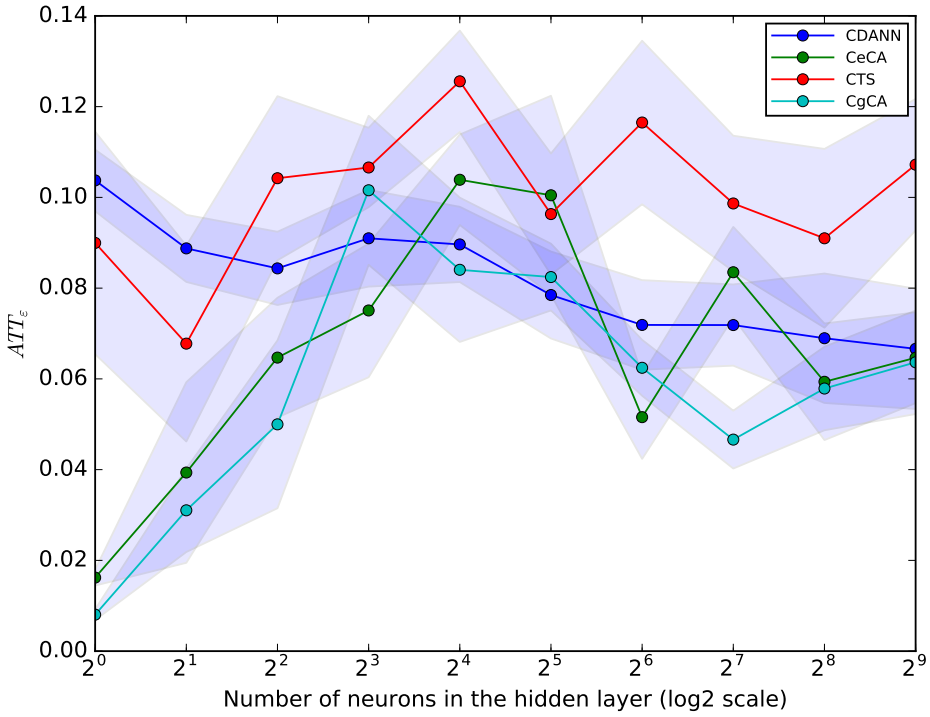


Figure E.1: Sensitivity of N to ATT_ε for all models on JOBS with $\lambda = 1$

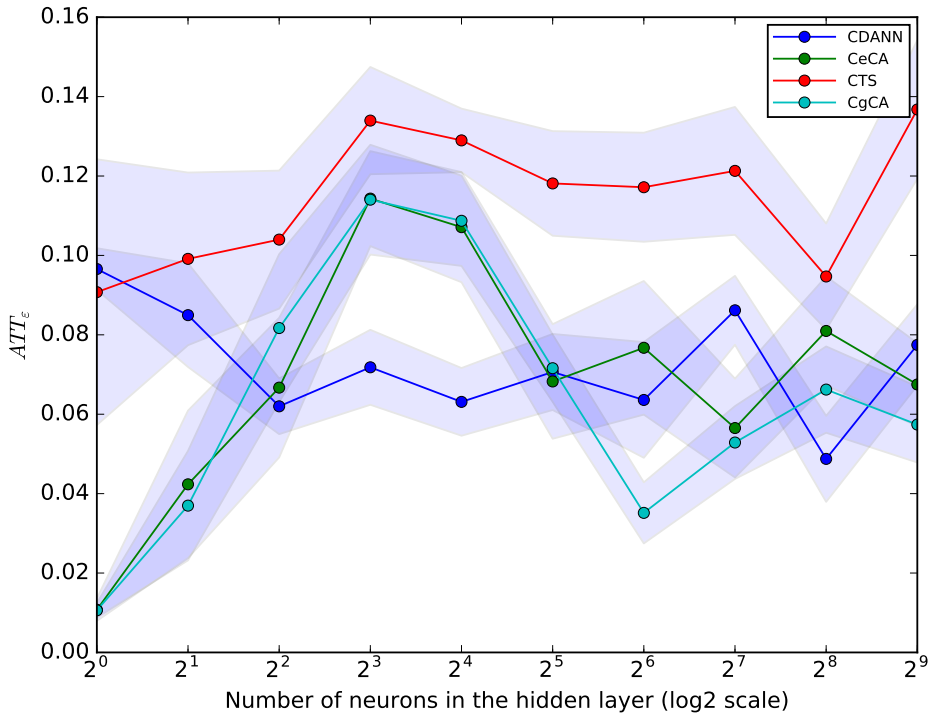


Figure E.2: Sensitivity of N to ATT_ε for all models on JOBS with $\lambda = 10$

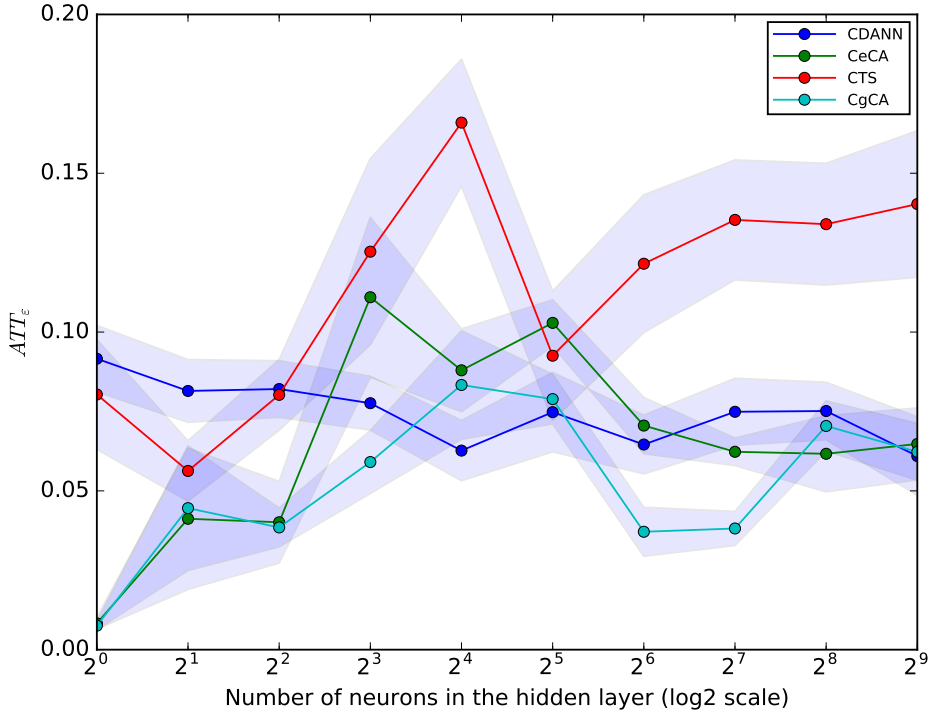


Figure E.3: Sensitivity of N to ATT_ε for all models on JOBS with $\lambda = 100$

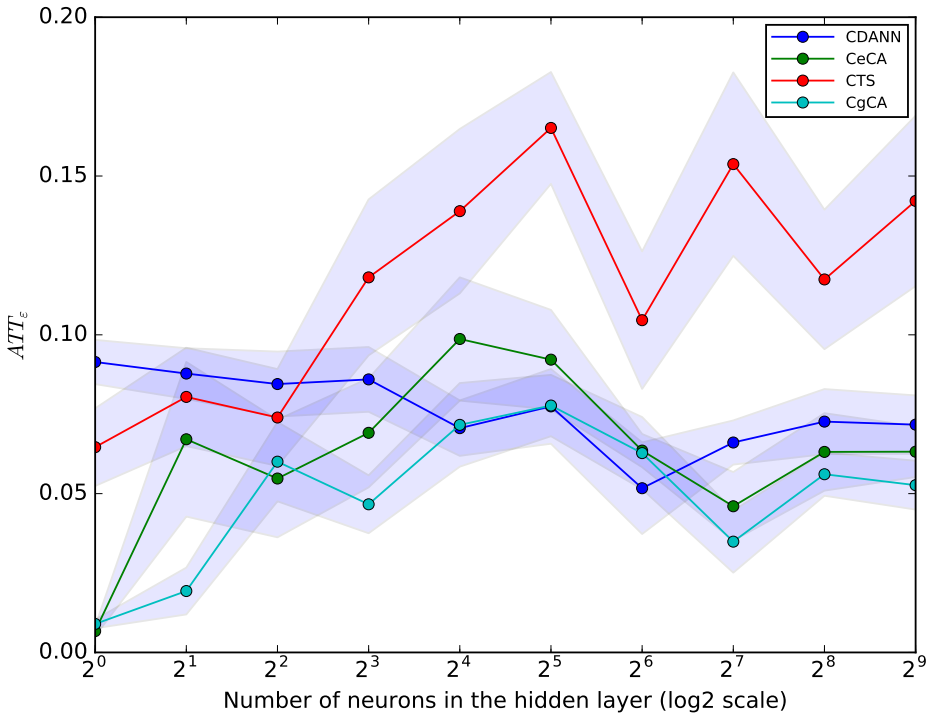


Figure E.4: Sensitivity of N to ATT_ε for all models on JOBS with $\lambda = 1000$

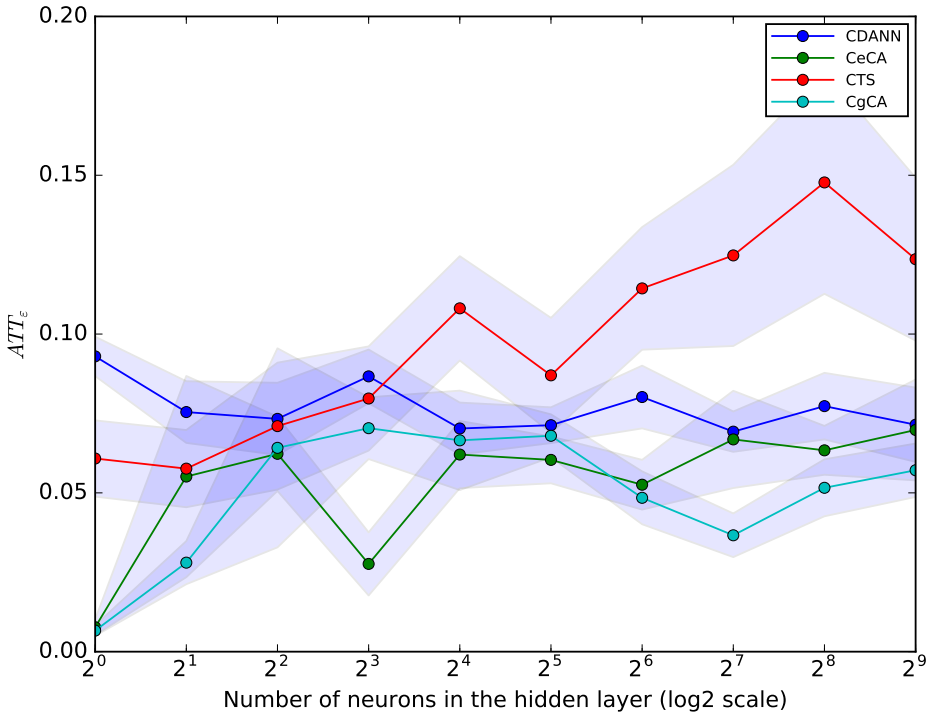


Figure E.5: Sensitivity of N to ATT_ε for all models on JOBS with $\lambda = 10^4$

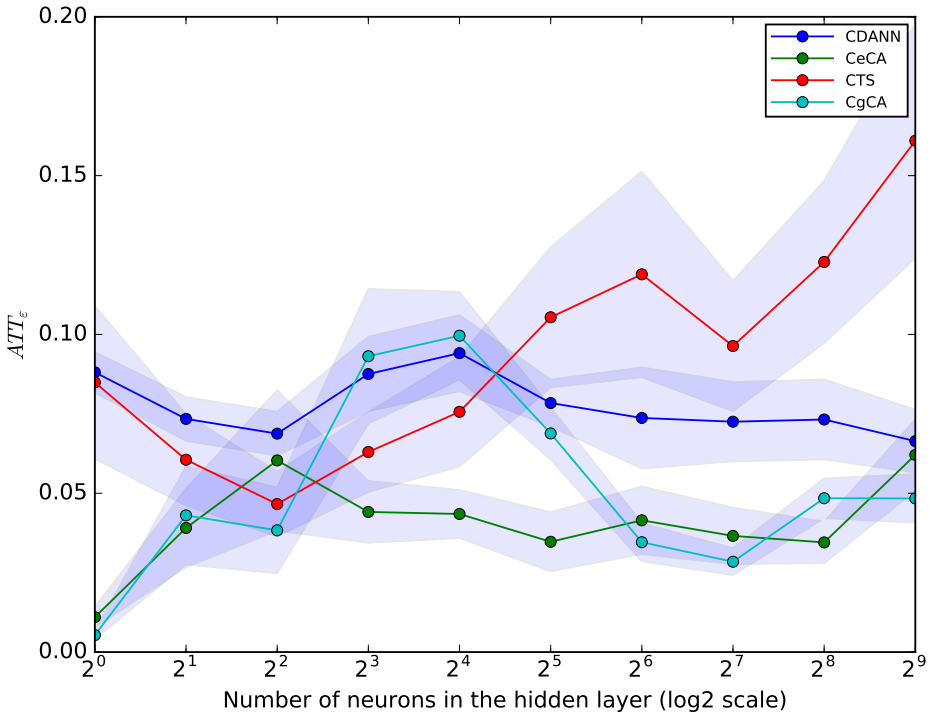


Figure E.6: Sensitivity of N to ATT_ε for all models on JOBS with $\lambda = 10^5$

Appendix F

Additional figures for P_{risk} with interaction between N and $\lambda > 0$

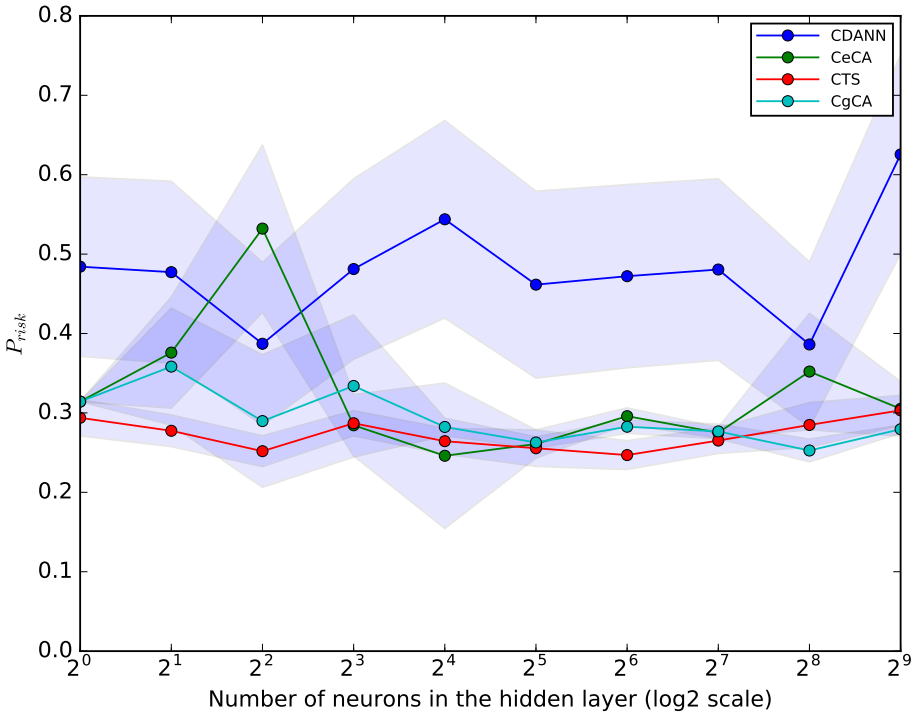


Figure F.1: Sensitivity of N to P_{risk} for all models on JOBS with $\lambda = 1$

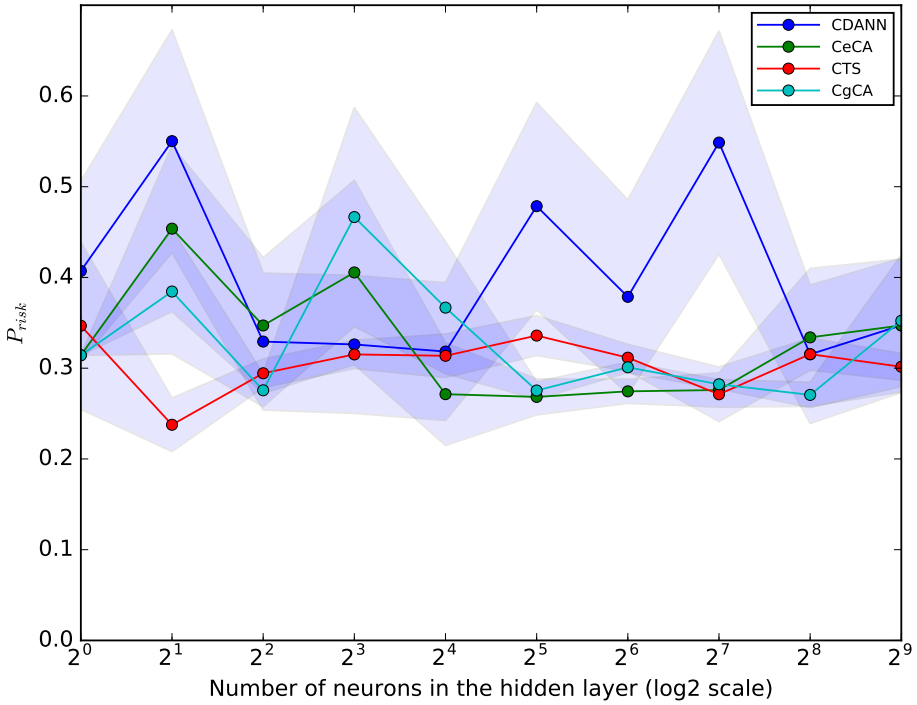


Figure F.2: Sensitivity of N to P_{risk} for all models on JOBS with $\lambda = 10$

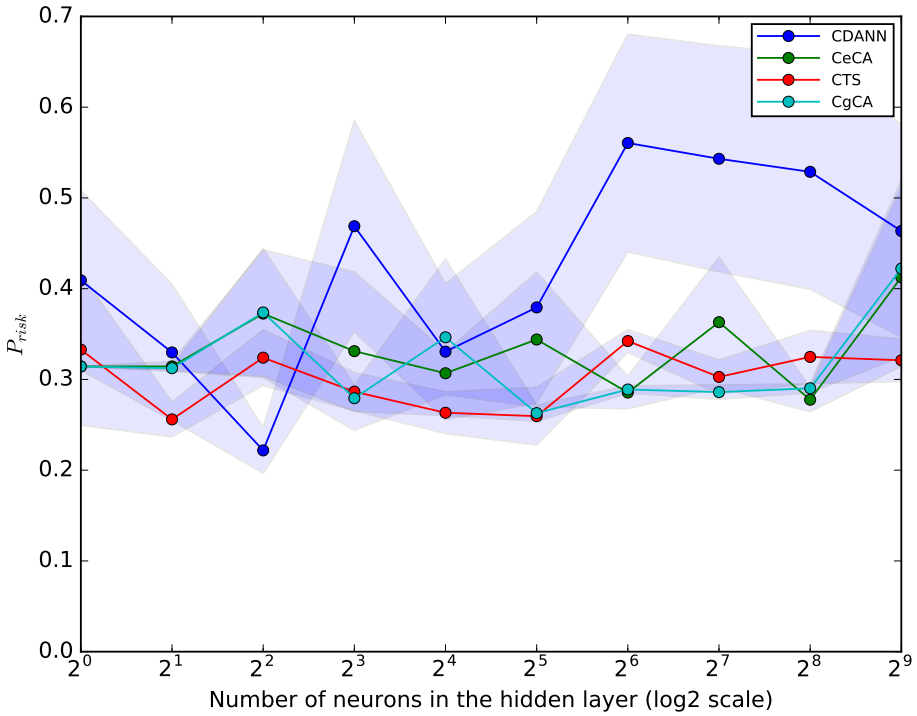


Figure F.3: Sensitivity of N to P_{risk} for all models on JOBS with $\lambda = 100$

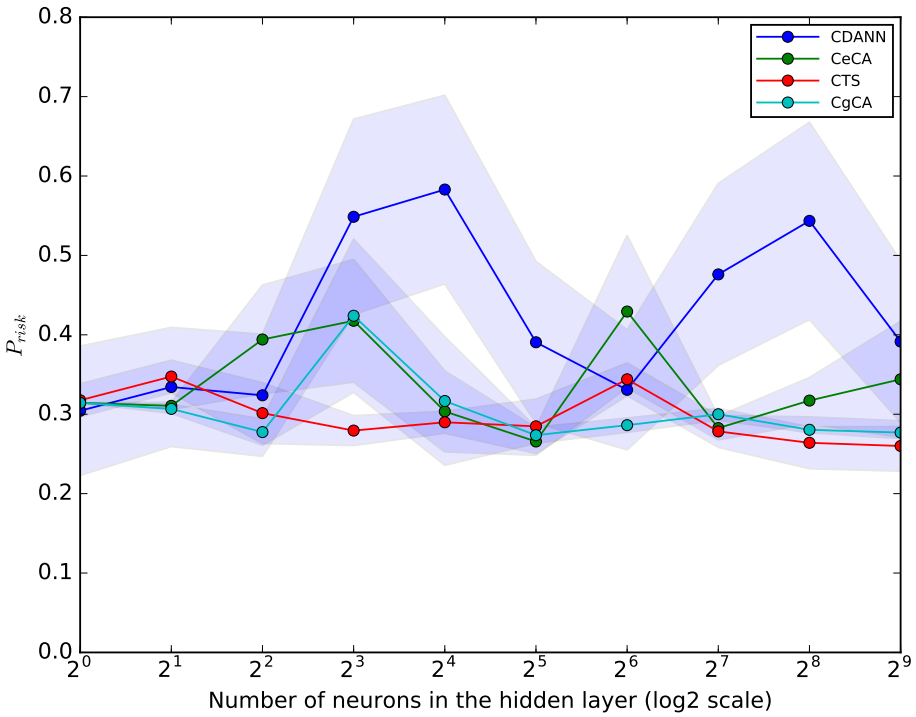


Figure F.4: Sensitivity of N to P_{risk} for all models on JOBS with $\lambda = 1000$

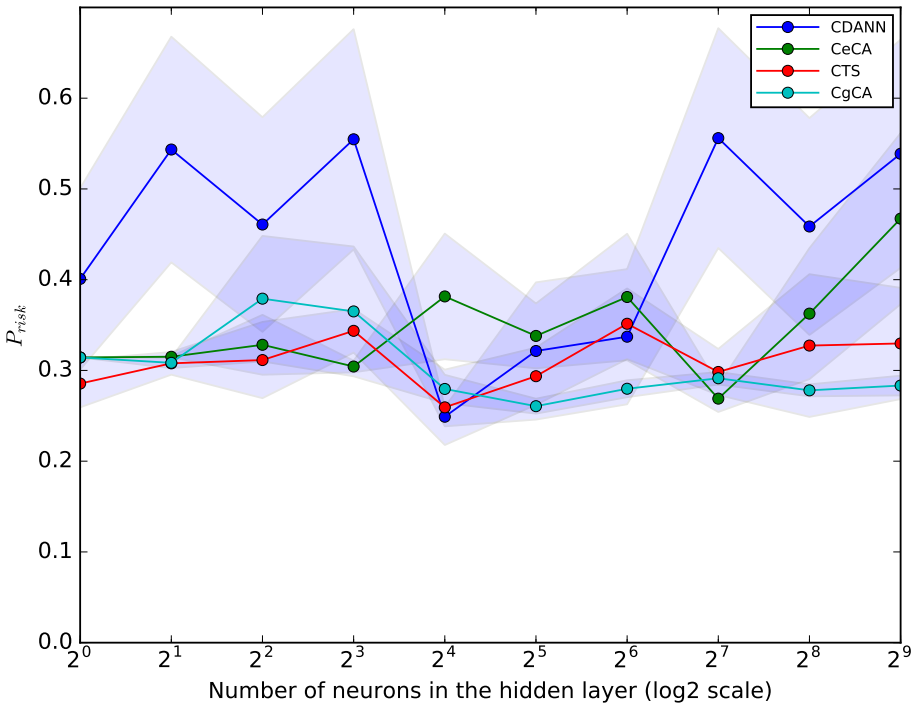


Figure F.5: Sensitivity of N to P_{risk} for all models on JOBS with $\lambda = 10^4$

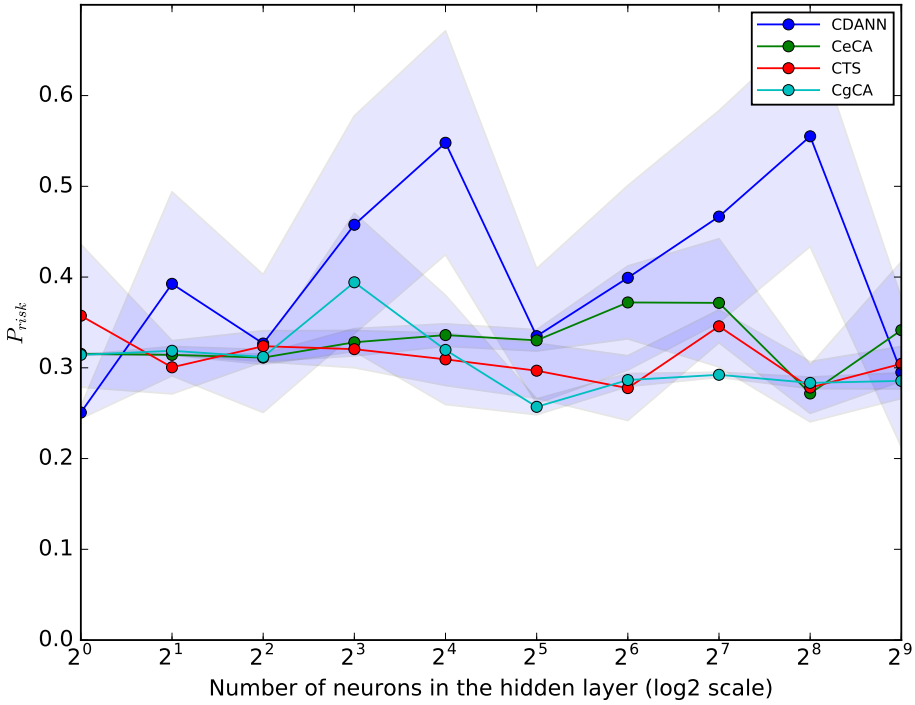


Figure F.6: Sensitivity of N to P_{risk} for all models on JOBS with $\lambda = 10^5$

Appendix G

Execution Time

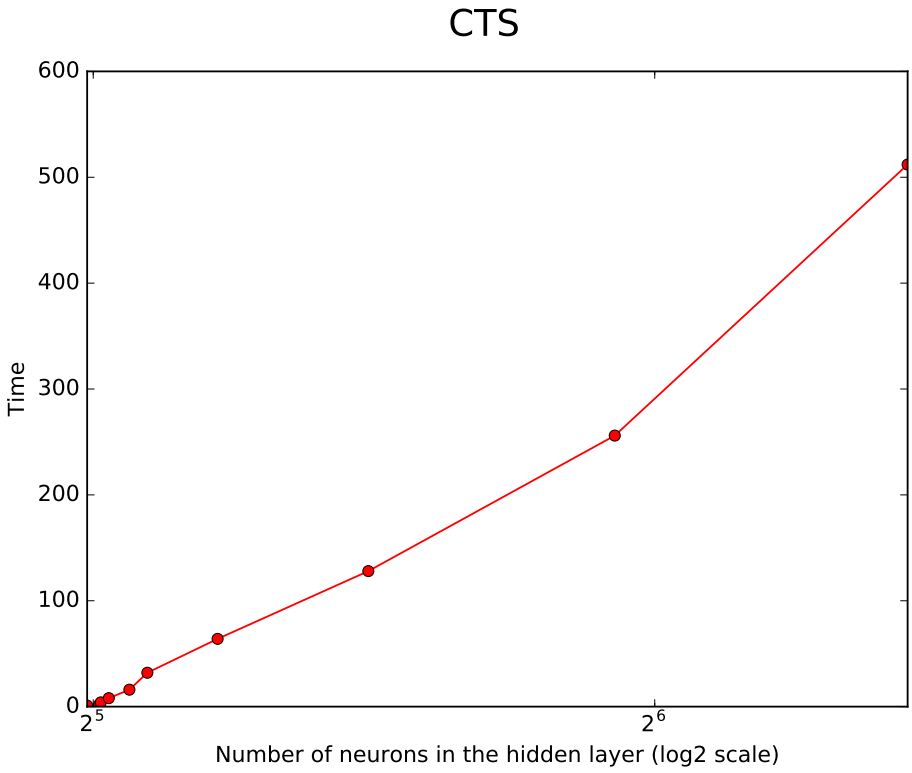


Figure G.1: N against time for CTS model on JOBS

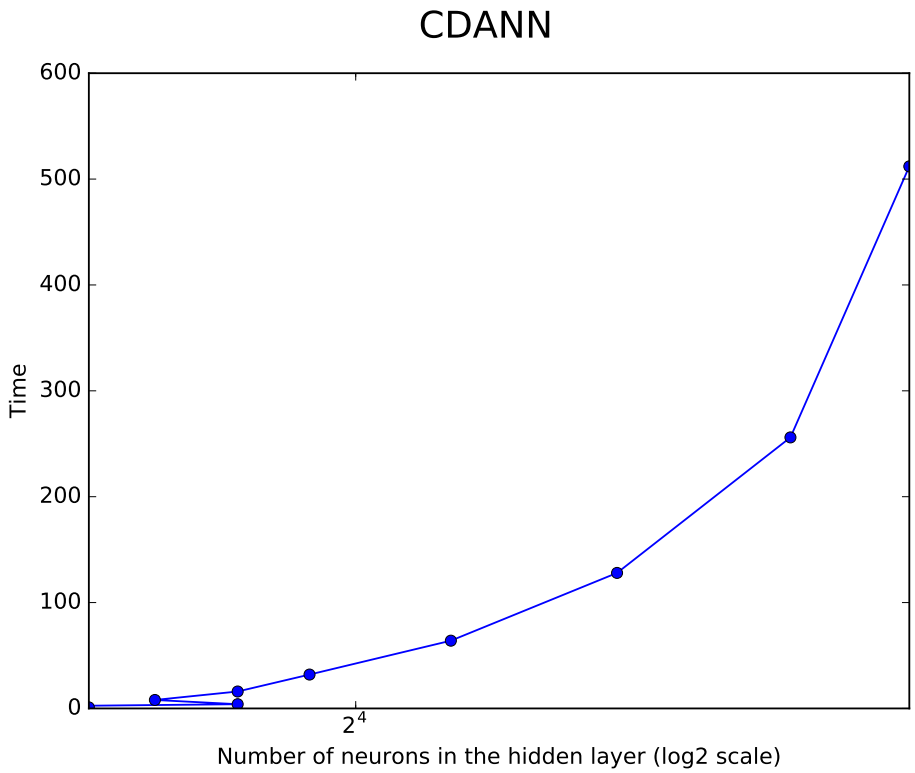


Figure G.2: N against time for $CDANN$ model on JOBS

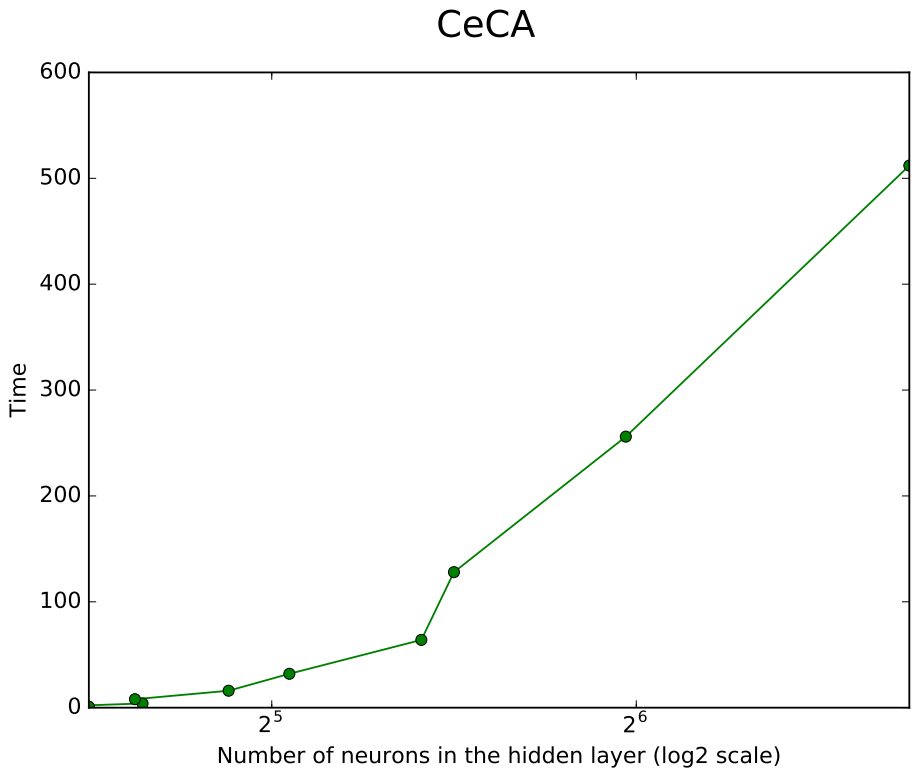


Figure G.3: N against time for $CeCA$ model on JOBS

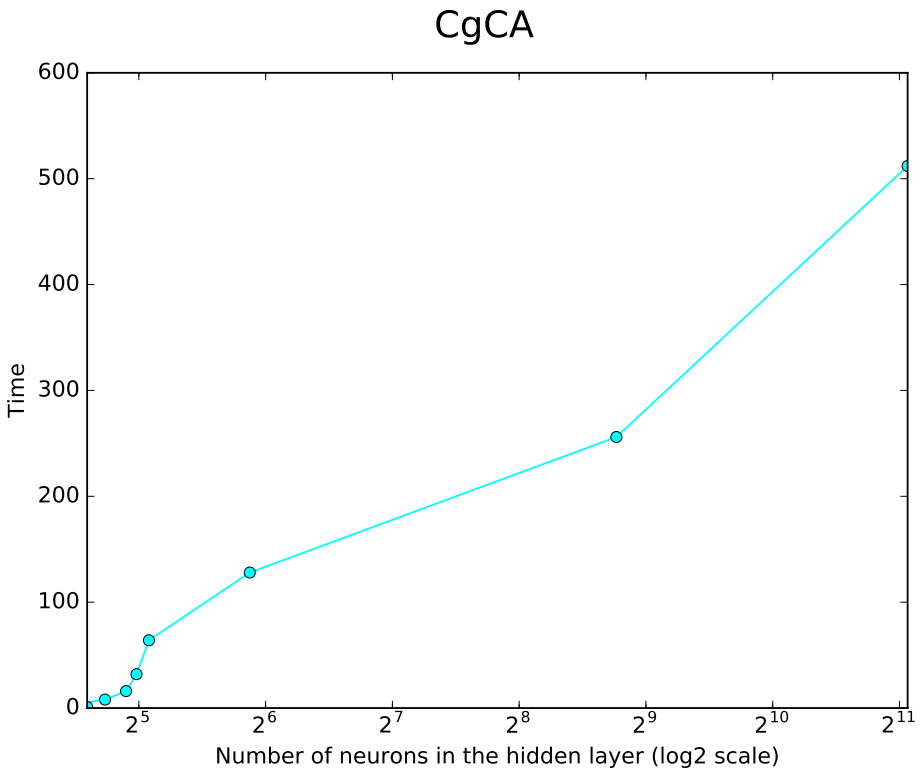


Figure G.4: N against time for $CgCA$ model on JOBS

Appendix H

Description of Features

Benchmark: IHDP

S/N	*Feature*	*Description*
1	birth weight	weight of the child at birth
2	head circumference	head circumference of the child
3	weeks born preterm	number of weeks before the child was born
4	birth order	
5	whether it is first born	whether is the first child of the mother
6	neonatal health index	
7	sex	gender of the twins
8	twin status	whether the child is twin
9	smoking	whether the mother smoked cigarette
10	drank alcohol	whether the mother drank alcohol while pregnant
11	drugs	whether the mother is into drugs
12	age	the age of the mother
13	marital status	whether the mother is married or single
14	educational attainment	(whether the mother attended high school, completed college etc.)
15	work	whether the mother was working during pregnancy
16	pre-natal	whether she received pre-natal care
17	address	The addresses with which the family resided during the entire period of the study

Table H.1: Description of Features for IHDP

Benchmark: JOBS

S/N	*Feature*	*Description*
1	age	age of participant
2	school	number of school years
3	black	1 if black 0 otherwise
4	hisp	1 if Hispanic 0 otherwise
5	No degree,	1 if participant had no school degrees, 0 otherwise
6	Married	1 if married, 0 otherwise
7	U75	1 if unemployed in 1975 0 otherwise
8	RE78 .	real earnings (1982US$) in 1978

Table H.2: Description of Features for JOBS

Benchmark: TWINS

S/N	*Feature*	*Description*
1	adequacy	adequacy of care
2	alcohol	: risk factor alcohol use
3	anemia ,	risk factor, Anemia
4	birattnd	medical person attending birth
5	birmon	birth month Jan-Dec
6	$bord_0$	birth order of lighter twin
7	$bord_1$	birth order of heavier twin
8	brstate	state of residence NCHS
9	$brstate_reg$	US census region of brstate
10	cardiac	risk factor, Cardiac
11	chyper .	risk factor, Hypertension, chronic
12	cigar6	num of cigarettes /day, quantiled
13	crace	race of child
14	csex	sex of child
15	$data_year$	year: 1989, 1990 or 1991
16	dfageq	octile age of father
17	diabetes	risk factor, Diabetes
18	$dlivord_min$	number of live births before twins
19	dmar	married
20	drink5	num of drinks /week, quantiled
21	$dtotord_min$	total number of births before twins
22	eclamp	risk factor, Eclampsia
23	feduc6	education category
24	frace	dad race
25	gestat10	gestation 10 categories
26	hemo	risk factor Hemoglobinopathy
27	herpes	risk factor, Herpes
28	hydra	risk factor Hvdramnios/Oliqohvdramnios
29	incervix	risk factor, Incompetent cervix

30	$infant_id_0$	infant id of lighter twin in original df
31	$infant_id_1$	infant id of heavier twin in original df
32	lung	risk factor, Lung
33	mager8	mom age
34	meduc6	mom education
35	mplbir	mom place of birth
36	mpre5	trimester prenatal care begun, 4 is none
37	mrace	mom race
38	nprevistq	quintile number of prenatal visits
39	orfath	dad hispanic
40	ormoth	mom hispanic
41	phyper	risk factor, Hypertension, preqnancy-associated
42	pldel	place of delivery
43	pre4000	risk factor, Previous infant 4000+ grams
44	renal	risk factor, Renal disease
45	stoccfipb	state of occurence FIPB
46	tobacco	risk factor, tobacco use

Table H.3: Description of Features for TWINS

Bibliography

[1] R. A. Fisher, *The design of experiments*. Oliver And Boyd; Edinburgh; London, 1937.

[2] D. B. Rubin, "The design versus the analysis of observational studies for causal effects: parallels with the design of randomized trials," *Statistics in medicine*, vol. 26, no. 1, pp. 20–36, 2007.

[3] A. Akobeng, "Understanding randomised controlled trials," *Archives of disease in childhood*, vol. 90, no. 8, pp. 840–844, 2005.

[4] B. Sibbald and M. Roland, "Understanding controlled trials. why are randomised controlled trials important?," *BMJ: British Medical Journal*, vol. 316, no. 7126, p. 201, 1998.

[5] L. Bereznicki, R. Castelino, *et al.*, "Understanding randomised controlled trials," *Australian Pharmacist*, vol. 32, no. 4, p. 71, 2013.

[6] D. B. Rubin, "Estimating causal effects of treatments in randomized and nonrandomized studies.," *Journal of educational Psychology*, vol. 66, no. 5, p. 688, 1974.

[7] K. A. Levin, "Study design vii. randomised controlled trials," *Evidence-based dentistry*, vol. 8, no. 1, pp. 22–23, 2007.

[8] N. Gleicher and D. H. Barad, "Misplaced obsession with prospectively randomized studies," *Reproductive biomedicine online*, vol. 21, no. 4, pp. 440–443, 2010.

[9] H.-H. Wong, A. Jessup, A. Sertkaya, A. Birkenbach, A. Berlind, and J. Eyraud, "Examination of clinical trial costs and barriers for drug development final," 2014.

[10] W. G. Cochran, "Observational studies," *Introduction to Observational Studies and the Reprint of Cochran's paper "Observational Studies" and Comments*, p. 126, 1972.

[11] W. G. Cochran and S. P. Chambers, "The planning of observational studies of human populations," *Journal of the Royal Statistical Society. Series A (General)*, vol. 128, no. 2, pp. 234–266, 1965.

[12] N. Black, "Why we need observational studies to evaluate the effectiveness of health care.," *BMJ: British Medical Journal*, vol. 312, no. 7040, p. 1215, 1996.

[13] P. R. Rosenbaum, "Observational studies," in *Observational Studies*, pp. 1–17, Springer, 2002.

[14] G. P. Hammer, J.-B. du Prel, and M. Blettner, "Avoiding bias in observational studies," *Dtsch Arzteblatt Int*, vol. 106, pp. 664–8, 2009.

[15] J. K. Kirklin, D. C. Naftel, R. L. Kormos, L. W. Stevenson, F. D. Pagani, M. A. Miller, K. L. Ulisney, J. T. Baldwin, and J. B. Young, "Second intermacs annual report: more than 1,000 primary left ventricular assist device implants," *The Journal of Heart and Lung Transplantation*, vol. 29, no. 1, pp. 1–10, 2010.

[16] H. Seibold, A. Zeileis, and T. Hothorn, "Individual treatment effect prediction for als patients," *arXiv preprint arXiv:1604.08720*, 2016.

[17] D. M. Kent, E. Steyerberg, and D. van Klaveren, "Personalized evidence based medicine: predictive approaches to heterogeneous treatment effects," *Bmj*, vol. 363, p. k4245, 2018.

[18] Y. H. Cho, J. K. Kim, and S. H. Kim, "A personalized recommender system based on web usage mining and decision tree induction," *Expert systems with Applications*, vol. 23, no. 3, pp. 329–342, 2002.

[19] X. Min-jie and Z. Jin-ge, "Research on personalized recommendation system for e-commerce based on web log mining and user browsing behaviors," in *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, vol. 12, pp. V12–408, IEEE, 2010.

[20] L.-p. Hung, "A personalized recommendation system based on product taxonomy for one-to-one marketing online," *Expert systems with applications*, vol. 29, no. 2, pp. 383–392, 2005.

[21] E. A. Stuart, "Matching methods for causal inference: A review and a look forward," *Statistical science: a review journal of the Institute of Mathematical Statistics*, vol. 25, no. 1, p. 1, 2010.

[22] D. B. Rubin, "Matching to remove bias in observational studies," *Biometrics*, pp. 159–183, 1973.

[23] P. C. Austin, "An introduction to propensity score methods for reducing the effects of confounding in observational studies," *Multivariate behavioral research*, vol. 46, no. 3, pp. 399–424, 2011.

[24] P. R. Rosenbaum and D. B. Rubin, "The central role of the propensity score in observational studies for causal effects," *Biometrika*, vol. 70, no. 1, pp. 41–55, 1983.

[25] P. R. Rosenbaum and D. B. Rubin, "Constructing a control group using multivariate matched sampling methods that incorporate the propensity score," *The American Statistician*, vol. 39, no. 1, pp. 33–38, 1985.

[26] P. R. Rosenbaum and D. B. Rubin, "Reducing bias in observational studies using subclassification on the propensity score," *Journal of the American statistical Association*, vol. 79, no. 387, pp. 516–524, 1984.

[27] D. B. Rubin and N. Thomas, "Matching using estimated propensity scores: relating theory to practice," *Biometrics*, pp. 249–264, 1996.

[28] K. Imai and M. Ratkovic, "Covariate balancing propensity score," *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pp. 243–263, 2014.

[29] P. C. Austin and E. A. Stuart, "Moving towards best practice when using inverse probability of treatment weighting (iptw) using the propensity score to estimate causal treatment effects in observational studies," *Statistics in medicine*, vol. 34, no. 28, pp. 3661–3679, 2015.

[30] J. J. Heckman, H. Ichimura, and P. Todd, "Matching As An Econometric Evaluation Estimator," *The Review of Economic Studies*, vol. 65, pp. 261–294, 04 1998.

[31] M. J. Funk, D. Westreich, C. Wiesen, T. Stürmer, M. A. Brookhart, and M. Davidian, "Doubly robust estimation of causal effects," *American journal of epidemiology*, vol. 173, no. 7, pp. 761–767, 2011.

[32] E. V. B. C. L. Alsberg, "Tariff on oils the tariff on animal and vegetable oils. by philip g. wright–the institute of eco-nomics, 12too. xviii–379 pp. the."

[33] P. G. Wright *et al.*, "Tariff on animal and vegetable oils," 1928.

[34] R. J. Bowden and D. A. Turkington, *Instrumental variables*. No. 8, Cambridge University Press, 1990.

[35] S. Greenland, "An introduction to instrumental variables for epidemiologists," *International journal of epidemiology*, vol. 29, no. 4, pp. 722–729, 2000.

[36] M. A. Hernán and J. M. Robins, "Instruments for causal inference: an epidemiologist's dream?," *Epidemiology*, pp. 360–372, 2006.

[37] D. L. Thistlethwaite and D. T. Campbell, "Regression-discontinuity analysis: An alternative to the ex post facto experiment," *Journal of Educational psychology*, vol. 51, no. 6, pp. 309–317, 1960.

[38] G. W. Imbens and T. Lemieux, "Regression discontinuity designs: A guide to practice," *Journal of econometrics*, vol. 142, no. 2, pp. 615–635, 2008.

[39] J. Hahn, P. Todd, and W. Van der Klaauw, "Identification and estimation of treatment effects with a regression-discontinuity design," *Econometrica*, vol. 69, no. 1, pp. 201–209, 2001.

[40] D. M. Hawkins, "Regression adjustment for variables in multivariate quality control," *Journal of Quality Technology*, vol. 25, no. 3, pp. 170–182, 1993.

[41] F. D. Johansson, U. Shalit, and D. Sontag, "Learning representations for counterfactual inference," *arXiv preprint arXiv:1605.03661*, 2016.

[42] F. D. Johansson, N. Kallus, U. Shalit, and D. Sontag, "Learning weighted representations for generalization across designs," *arXiv preprint arXiv:1802.08598*, 2018.

[43] L. Bottou, J. Peters, J. Q. Candela, D. X. Charles, M. Chickering, E. Portugaly, D. Ray, P. Y. Simard, and E. Snelson, "Counterfactual reasoning and learning systems: the example of computational advertising.," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 3207–3260, 2013.

[44] S. Wager and S. Athey, "Estimation and inference of heterogeneous treatment effects using random forests," *Journal of the American Statistical Association*, no. just-accepted, 2017.

[45] U. Shalit, F. D. Johansson, and D. Sontag, "Estimating individual treatment effect: generalization bounds and algorithms," in *Proceedings of the 34th International Conferenc e on Machine Learning-Volume 70*, pp. 3076–3085, JMLR. org, 2017.

[46] N. Rosenfeld, Y. Mansour, and E. Yom-Tov, "Predicting counterfactuals from large historical data and small randomized trials," in *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 602–609, International World Wide Web Conferences Steering Committee, 2017.

[47] J. L. Hill, "Bayesian nonparametric modeling for causal inference," *Journal of Computational and Graphical Statistics*, vol. 20, no. 1, pp. 217–240, 2011.

[48] H. A. Chipman, E. I. George, and R. E. McCulloch, "Bart: Bayesian additive regression trees," 2010.

[49] A. M. Alaa, M. Weisz, and M. van der Schaar, "Deep counterfactual networks with propensity-dropout," *arXiv preprint arXiv:1706.05966*, 2017.

[50] J. Yoon, J. Jordon, and M. van der Schaar, "Ganite: Estimation of individualized treatment effects using generative adversarial nets," 2018.

[51] L. Yao, S. Li, Y. Li, M. Huai, J. Gao, and A. Zhang, "Representation learning for treatment effect estimation from observational data," in *Advances in Neural Information Processing Systems*, pp. 2633–2643, 2018.

[52] P. Morerio, J. Cavazza, and V. Murino, "Minimal-entropy correlation alignment for unsupervised deep domain adaptation," *CoRR*, vol. abs/1711.10288, 2017.

[53] B. Sun and K. Saenko, "Deep CORAL: correlation alignment for deep domain adaptation," *CoRR*, vol. abs/1607.01719, 2016.

[54] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[55] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *arXiv preprint arXiv:1603.06432*, 2016.

[56] "Statistics covariance2." https://towardsdatascience.com/statistical-thinking-understanding-correlation-5f7c63934699. Accessed: 2020-07-11.

[57] "Negative Correlation ." https://intl.siyavula.com/read/maths/grade-12/statistics/09-statistics-03. Accessed: 2020-07-19.

[58] "Statistics covariance." https://web.stanford.edu/class/archive/cs/cs109/cs109.1178/lectureHandouts/150-covariance.pdf. Accessed: 2020-07-11.

[59] J. S. Mill, *A system of logic ratiocinative and inductive: Being a connected view of the principles of evidence and the methods of scientific investigation*. Harper, 1884.

[60] D. Hume, L. A. Selby-Bigge, and P. H. Nidditch, "Enquiries concerning human understanding and concerning the principles of morals," 1976.

[61] D. Lewis, "Counterfactuals and comparative possibility," in *Ifs*, pp. 57–85, Springer, 1973.

[62] L. Meuli and F. Dick, "Understanding confounding in observational studies," *European Journal of Vascular and Endovascular Surgery*, vol. 55, no. 5, p. 737, 2018.

[63] K. Jager, C. Zoccali, A. Macleod, and F. Dekker, "Confounding: what it is and how to deal with it," *Kidney international*, vol. 73, no. 3, pp. 256–260, 2008.

[64] M. A. Pourhoseingholi, A. R. Baghestani, and M. Vahedi, "How to control confounding effects by statistical analysis," *Gastroenterology and hepatology from bed to bench*, vol. 5, no. 2, p. 79, 2012.

[65] "Matching methods." https://data.library.virginia.edu/getting-started-with-matching-methods/. Accessed: 2020-06-21.

[66] P. R. Rosenbaum *et al.*, "The role of a second control group in an observational study," *Statistical Science*, vol. 2, no. 3, pp. 292–306, 1987.

[67] D. B. Rubin, "Using propensity scores to help design observational studies: application to the tobacco litigation," *Health Services and Outcomes Research Methodology*, vol. 2, no. 3-4, pp. 169–188, 2001.

[68] D. B. Rubin, "Using multivariate matched sampling and regression adjustment to control bias in observational studies," *Journal of the American Statistical Association*, vol. 74, no. 366a, pp. 318–328, 1979.

[69] Z. Zhao, "Using matching to estimate treatment effects: Data requirements, matching metrics, and monte carlo evidence," *review of economics and statistics*, vol. 86, no. 1, pp. 91–107, 2004.

[70] S. M. Iacus, G. King, and G. Porro, "Cem: software for coarsened exact matching," 2009.

[71] R. H. Dehejia and S. Wahba, "Propensity score-matching methods for nonexperimental causal studies," *Review of Economics and statistics*, vol. 84, no. 1, pp. 151–161, 2002.

[72] P. R. Rosenbaum, "Optimal matching for observational studies," *Journal of the American Statistical Association*, vol. 84, no. 408, pp. 1024–1032, 1989.

[73] W. Feng, Y. Jun, R. Xu, *et al.*, "A method/macro based on propensity score and mahalanobis distance to reduce bias in treatment comparison in observational study," in *SAS PharmaSUG 2006 Conference*, 2006.

[74] " Comparing regression, propensity matching and coarsened exact matching in healthcare observational studies using SAS®: An example from the Medical Expenditure Panel Survey (MEPS) ." `https://www.lexjansen.com/mwsug/2014/AA/MWSUG-2014-AA02.pdf`. Accessed: 2021-03-25.

[75] W. G. Cochran, "The effectiveness of adjustment by subclassification in removing bias in observational studies," *Biometrics*, pp. 295–313, 1968.

[76] J. K. Lunceford and M. Davidian, "Stratification and weighting via the propensity score in estimation of causal treatment effects: a comparative study," *Statistics in medicine*, vol. 23, no. 19, pp. 2937–2960, 2004.

[77] H. L. Smith, "6. matching with multiple controls to estimate treatment effects in observational studies," *Sociological methodology*, vol. 27, no. 1, pp. 325–353, 1997.

[78] D. B. Rubin and N. Thomas, "Combining propensity score matching with additional adjustments for prognostic covariates," *Journal of the American Statistical Association*, vol. 95, no. 450, pp. 573–585, 2000.

[79] R. H. Dehejia and S. Wahba, "Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs," *Journal of the American statistical Association*, vol. 94, no. 448, pp. 1053–1062, 1999.

[80] J. L. Czajka, S. M. Hirabayashi, R. J. Little, and D. B. Rubin, "Projecting from advance data using propensity modeling: An application to income and tax statistics," *Journal of Business & Economic Statistics*, vol. 10, no. 2, pp. 117–131, 1992.

[81] J. M. Robins, M. A. Hernan, and B. Brumback, "Marginal structural models and causal inference in epidemiology," 2000.

[82] K. Hirano, G. W. Imbens, and G. Ridder, "Efficient estimation of average treatment effects using the estimated propensity score," *Econometrica*, vol. 71, no. 4, pp. 1161–1189, 2003.

[83] G. W. Imbens, "The role of the propensity score in estimating dose-response functions," *Biometrika*, vol. 87, no. 3, pp. 706–710, 2000.

[84] J. D. Kang, J. L. Schafer, *et al.*, "Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data," *Statistical science*, vol. 22, no. 4, pp. 523–539, 2007.

[85] " Cross-Sectional study." http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Multivariable/BS704_Multivariable7.html. Accessed: 2020-06-13.

[86] A. Pokropek, "Introduction to instrumental variables and their application to large-scale assessment data," *Large-scale Assessments in Education*, vol. 4, no. 1, p. 4, 2016.

[87] M. L. Lousdal, "An introduction to instrumental variable assumptions, validation and estimation," *Emerging themes in epidemiology*, vol. 15, no. 1, p. 1, 2018.

[88] J. D. Angrist, G. W. Imbens, and D. B. Rubin, "Identification of causal effects using instrumental variables," *Journal of the American statistical Association*, vol. 91, no. 434, pp. 444–455, 1996.

[89] J. A. Rassen, M. A. Brookhart, R. J. Glynn, M. A. Mittleman, and S. Schneeweiss, "Instrumental variables i: instrumental variables exploit natural variation in nonexperimental data to estimate causal relationships," *Journal of clinical epidemiology*, vol. 62, no. 12, pp. 1226–1232, 2009.

[90] N. M. Davies, G. D. Smith, F. Windmeijer, and R. M. Martin, "Brief report: Issues in the reporting and conduct of instrumental variable studies: A systematic review," *Epidemiology*, pp. 363–369, 2013.

[91] J. Hartford, G. Lewis, K. Leyton-Brown, and M. Taddy, "Deep iv: A flexible approach for counterfactual prediction," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1414–1423, JMLR. org, 2017.

[92] L. Yao, S. Li, Y. Li, M. Huai, J. Gao, and A. Zhang, "Representation learning for treatment effect estimation from observational data," in *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, (USA), pp. 2638–2648, Curran Associates Inc., 2018.

[93] R. Guo, L. Cheng, J. Li, P. R. Hahn, and H. Liu, "A survey of learning causality with data: Problems and methods," *CoRR*, vol. abs/1809.09337, 2018.

[94] M. Kabisch, C. Ruckes, M. Seibert-Grafe, and M. Blettner, "Randomized controlled trials: part 17 of a series on evaluation of scientific publications," *Deutsches Ärzteblatt International*, vol. 108, no. 39, p. 663, 2011.

[95] S. Houle, "An introduction to the fundamentals of randomized controlled trials in pharmacy research," *The Canadian journal of hospital pharmacy*, vol. 68, no. 1, p. 28, 2015.

[96] A. Bhide, P. S. Shah, and G. Acharya, "A simplified guide to randomized controlled trials," *Acta obstetricia et gynecologica Scandinavica*, vol. 97, no. 4, pp. 380–387, 2018.

[97] " Experimental Studies ." https://www.cebm.net/2014/04/study-designs/. Accessed: 2020-10-11.

[98] "Research methods in psychology."

[99] "Deep Learning Architecture ." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6121485/. Accessed: 2020-08-07.

[100] " Cross-Sectional study." https://lo.unisa.edu.au/mod/book/view.php?id=646428&chapterid=117794. Accessed: 2020-06-13.

[101] A. Gelman, "Causality and statistical learning," *American Journal of Sociology*, vol. 117, no. 3, pp. 955–966, 2011.

[102] S. L. Morgan and C. Winship, *Counterfactuals and causal inference*. Cambridge University Press, 2015.

[103] J. Splawa-Neyman, D. M. Dabrowska, T. P. Speed, *et al.*, "On the application of probability theory to agricultural experiments. essay on principles. section 9," *Statistical Science*, vol. 5, no. 4, pp. 465–472, 1990.

[104] P. W. Holland, "Statistics and causal inference," *Journal of the American statistical Association*, vol. 81, no. 396, pp. 945–960, 1986.

[105] D. R. Cox, "Planning of experiments.," 1958.

[106] D. B. Rubin, "Causal inference using potential outcomes," *Journal of the American Statistical Association*, 2011.

[107] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.

[108] B. Coppin, *Artificial intelligence illuminated*. Jones & Bartlett Learning, 2004.

[109] "Machine learning circle." https://towardsdatascience.com/cousins-of-artificial-intelligence-dda4edc27b55. Accessed: 2020-07-23.

[110] T. M. Mitchell, J. G. Carbonell, and R. S. Michalski, *Machine learning: a guide to current research*, vol. 12. Springer Science & Business Media, 1986.

[111] "Traditional programming vs machine learning." https://towardsdatascience.com/a-gentle-introduction-to-deep-learning-part-1-introducti Accessed: 2020-07-15.

[112] N. J. Nilsson and N. J. Nilsson, *Artificial intelligence: a new synthesis*. Morgan Kaufmann, 1998.

[113] R. S. Sutton, "Introduction: The challenge of reinforcement learning," in *Reinforcement Learning*, pp. 1–3, Springer, 1992.

[114] "Unsupervised learning 2." https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a. Accessed: 2020-06-03.

[115] "Unsupervised learning 1." https://www.datarobot.com/wiki/unsupervised-machine-learning/. Accessed: 2020-06-03.

[116] "Biological neuron." https://towardsdatascience.com/deep-learning-versus-biological-neurons-floating-point-numbers-spikes-and-neu Accessed: 2020-07-15.

[117] K. Gurney, *An introduction to neural networks*. CRC press, 1997.

[118] "Deep Learning Architecture ." https://cs231n.github.io/neural-networks-1/. Accessed: 2020-07-22.

[119] A. J. Maren, C. T. Harston, and R. M. Pap, *Handbook of neural computing applications*. Academic Press, 2014.

[120] "Activation Functions 1 ." https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-network Accessed: 2021-03-29.

[121] "Activation Functions 2." https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6. Accessed: 2021-03-29.

[122] "Gradient descent." https://srdas.github.io/DLBook/GradientDescentTechniques.html. Accessed: 2020-09-30.

[123] "An overview of gradient descent optimization algorithms ." https://ruder.io/optimizing-gradient-descent/. Accessed: 2020-09-22.

[124] "Understanding adaptive optimization techniques in deep learning." https://analyticsindiamag.com/understanding-adaptive-optimization-techniques-in-deep-learning/. Accessed: 2020-10-19.

[125] "Loss Functions 2 ." https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23. Accessed: 2021-03-29.

[126] "Cross-entropy." https://machinelearningmastery.com/cross-entropy-for-machine-learning/. Accessed: 2020-09-30.

[127] "Loss Functions 1 ." https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/. Accessed: 2021-03-29.

[128] " Representation learning ." https://www.quora.com/What-is-representation-learning-in-deep-learning. Accessed: 2020-08-05.

[129] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[130] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in neural information processing systems*, pp. 2924–2932, 2014.

[131] M. Basseville, "Divergence measures for statistical data processing—an annotated bibliography," *Signal Processing*, vol. 93, no. 4, pp. 621–633, 2013.

[132] A. J. Smola, A. Gretton, and K. Borgwardt, "Maximum mean discrepancy," in *13th International Conference, ICONIP 2006, Hong Kong, China, October 3-6, 2006: Proceedings*, 2006.

[133] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. 25, pp. 723–773, 2012.

[134] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[135] A. Berlinet and C. Thomas-Agnan, *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[136] M. Sugiyama, M. Krauledat, and K.-R. MÃžller, "Covariate shift adaptation by importance weighted cross validation," *Journal of Machine Learning Research*, vol. 8, no. May, pp. 985–1005, 2007.

[137] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning under covariate shift," *Journal of Machine Learning Research*, vol. 10, no. Sep, pp. 2137–2155, 2009.

[138] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Advances in neural information processing systems*, pp. 601–608, 2007.

[139] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1-2, pp. 151–175, 2010.

[140] W. M. Kouw, "An introduction to domain adaptation and transfer learning," *CoRR*, vol. abs/1812.11806, 2018.

[141] M. Kubat, R. C. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Machine Learning*, vol. 30, pp. 195–215, Feb 1998.

[142] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[143] "Data Shift in Machine Learning: what is it and how to detect it ." `https://gsarantitis.wordpress.com/2020/04/16/data-shift-in-machine-learning-what-is-it-and-how-to-detect-it/`. Accessed: 2020-07-11.

[144] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *Computational Intelligence Magazine, IEEE*, vol. 10, pp. 12–25, 11 2015.

[145] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[146] B. Sun and K. Saenko, "Subspace distribution alignment for unsupervised domain adaptation.," in *BMVC*, vol. 4, pp. 24–1, 2015.

[147] S. Roy, A. Siarohin, E. Sangineto, S. R. Bulò, N. Sebe, and E. Ricci, "Unsupervised domain adaptation using feature-whitening and consensus loss," *CoRR*, vol. abs/1903.03215, 2019.

[148] S. Wager and S. Athey, "Estimation and Inference of Heterogeneous Treatment Effects using Random Forests," *ArXiv e-prints*, Oct. 2015.

[149] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[150] P. Schwab, L. Linhardt, and W. Karlen, "Perfect match: A simple method for learning representations for counterfactual inference with neural networks," *arXiv preprint arXiv:1810.00656*, 2018.

[151] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," *arXiv preprint arXiv:0902.3430*, 2009.

[152] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, *et al.*, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, p. 137, 2007.

[153] M. Cuturi and A. Doucet, "Fast computation of wasserstein barycenters," in *International conference on machine learning*, pp. 685–693, PMLR, 2014.

[154] X. Du, L. Sun, W. Duivesteijn, A. Nikolaev, and M. Pechenizkiy, "Adversarial balancing-based representation learning for causal effect inference with observational data," *arXiv preprint arXiv:1904.13335*, 2019.

[155] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. Zemel, and M. Welling, "Causal effect inference with deep latent-variable models," in *Advances in Neural Information Processing Systems*, pp. 6446–6456, 2017.

[156] J. Pearl, "[bayesian analysis in expert systems]: comment: graphical models, causality and intervention," *Statistical Science*, vol. 8, no. 3, pp. 266–269, 1993.

[157] H. A. Chipman, E. I. George, R. E. McCulloch, *et al.*, "Bart: Bayesian additive regression trees," *The Annals of Applied Statistics*, vol. 4, no. 1, pp. 266–298, 2010.

[158] "Difference Between Euclidean and Geodesic ." `https://images.app.goo.gl/9jbtz1Haj4mP9ty`, note = Accessed: 2020-09-05.

[159] U. I. Abdullahi, S. Samothrakis, and M. Fasli, "Counterfactual domain adversarial training of neural networks," in *2017 International Conference on the Frontiers and Advances in Data Science (FADS)*, pp. 151–155, IEEE, 2017.

[160] J. Brooks-Gunn, F.-r. Liaw, and P. K. Klebanov, "Effects of early intervention on cognitive function of low birth weight preterm infants," *The Journal of pediatrics*, vol. 120, no. 3, pp. 350–359, 1992.

[161] "Topic Modelling ." `https://towardsdatascience.com/topic-modeling-for-the-new-york-times-news-dataset-1f643e15caac`. Accessed: 2021-04-15.

[162] J. A. Smith and P. E. Todd, "Does matching overcome lalonde's critique of nonexperimental estimators?," *Journal of econometrics*, vol. 125, no. 1-2, pp. 305–353, 2005.

[163] R. J. LaLonde, "Evaluating the econometric evaluations of training programs with experimental data," *The American economic review*, pp. 604–620, 1986.

[164] D. Almond, K. Y. Chay, and D. S. Lee, "The costs of low birth weight," *The Quarterly Journal of Economics*, vol. 120, no. 3, pp. 1031–1083, 2005.

[165] E. L. Lehmann, "Significance level and power," *The Annals of Mathematical Statistics*, pp. 1167–1176, 1958.

[166] L. S. Feldt, D. J. Woodruff, and F. A. Salih, "Statistical inference for coefficient alpha," *Applied psychological measurement*, vol. 11, no. 1, pp. 93–103, 1987.

[167] D. S. Moore and S. Kirkland, *The basic practice of statistics*, vol. 2. WH Freeman New York, 2007.

[168] "The p-value explained ." `https://www.scribbr.com/statistics/p-value/`, note = Accessed: 2021-06-17.

[169] J. W. Pratt and J. D. Gibbons, "Kolmogorov-smirnov two-sample tests," in *Concepts of nonparametric theory*, pp. 318–344, Springer, 1981.

[170] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020.

[171] R. K. Crump, V. J. Hotz, G. W. Imbens, and O. A. Mitnik, "Nonparametric tests for treatment effect heterogeneity," *The Review of Economics and Statistics*, vol. 90, no. 3, pp. 389–405, 2008.

[172] S. Wager and S. Athey, "Estimation and inference of heterogeneous treatment effects using random forests. arxiv. org," 2015.

[173] J. B. Ullman and P. M. Bentler, "Structural equation modeling," *Handbook of psychology*, pp. 607–634, 2003.

[174] R. H. Hoyle, "The structural equation modeling approach: Basic concepts and fundamental issues.," 1995.

[175] N. Kallus, "Deepmatch: Balancing deep covariate representations for causal inference using adversarial training," *arXiv preprint arXiv:1802.05664*, 2018.