UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

STATISTICAL AND DEEP LEARNING METHODS IN CAUSAL INFERENCE

Albert Whata (1818479)

Supervisor

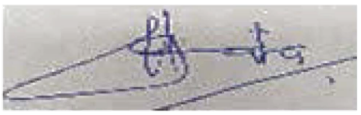
Dr Charles Chimedza

A thesis submitted to the Faculty of Science, University of the Witwatersrand, in fulfilment of the requirements for the Degree of Doctor of Philosophy.

October 31, 2021

DECLARATION

I declare that this thesis is my own, unaided work. It is being submitted for the Degree of Doctor of Philosophy at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.



Signature of Candidate

31st day of October 2021

Date

ABSTRACT

Machine learning (ML) algorithms are excellent at predicting outcomes rather than explaining causality. On the other hand, deep learning algorithms such as deep neural networks (DNN) are especially good at uncovering some hidden patterns in large data sets, but they struggle when it comes to making simple causal inferences. Causal inference is a statistical tool that can be used by machine learning and deep learning to measure the causal effects of multiple variables. This research was carried out to show researchers that it is very important to start incorporating causal inference into machine learning systems and not to just focus on predicting outcomes. A propensity scores-potential outcomes framework was used to evaluate machine learning and statistical causal inference. Using the propensity scores-potential outcomes framework , it was successfully demonstrated that a deep learning algorithm such as DNN can be adapted and used for the classification tasks. In addition, the results in this thesis have shown that using DNN, one can successfully estimate propensity scores, and also reduce absolute bias in the treatment effects that are estimated using these propensity scores. A hybrid model that consisted of a long-short term memory autoencoder (LSTMAE) and the kernel quantile estimator (KQE) algorithm was also successfully developed to detect change-points. Additionally, a multivariate regression discontinuity design (MRDD) was effectively employed to evaluate the statistical causal effect using two assignment variables. Also, the study demonstrated the importance of accompanying every conventional or multivariate regression discontinuity design with supplementary analyses to give more credibility to the causal estimates. A hybrid deep learning algorithm that uses a convolutional neural network (CNN) as well as a bidirectional long-short term memory (Bi-LSTM) neural network was developed for the classification of the severe acute respiratory syndrome coronavirus 2 (SARS CoV-2) among Coronaviruses. The model achieved impressive on metrics such as classification accuracy, area under curve receiver operating characteristic (AUC ROC), and Cohen's Kappa. The results show that deep learning algorithms can be used as alternative avenues to detect SARS CoV-2 among Coronaviruses.

Structure and Outputs of the Thesis

Chapter 1 presents the background of the thesis and the context of the work. In addition, the chapter gives an idea of recent techniques in the field of statistical causal inference and the contributions made by the thesis.

Chapter 2 presents a brief description of the literature review. This chapter presents an evaluation of some of the available literature in the field of statistics and causal inference. It documents the methods of evaluating causal inference and identifies the novel aspects of the research.

Chapters 3, 4, 5, and 6 (*Papers 1, 2, 3, and 4*) are a collection of detailed research papers from this research on statistical and deep learning methods for evaluating causal inference. The chapters follow the order of the list of publications stated below. Some of the chapters have already been published in peer-reviewed journals in statistics and machine learning, and some are still under review.

Chapter 7 is a synthesis of the papers/articles in Chapters 3, 4, 5 and 6. All the papers use related techniques (the potential outcomes or counterfactual framework for causal inference), with common objectives (detection, prediction, and estimating treatment effects) that revolve around establishing collaborations between machine learning, statistics, and causal inference.

Chapter 8 concludes the thesis drawing on the key issues that have emerged in the thesis through the study of the different chapters or research papers and the discussion of the contributions of each research paper. The chapter emphasises the new knowledge contributed by the thesis.

List of Publications from this Research

Published Journals

Paper 2: **Whata, A.** and Chimedza, C. (2021), A Machine Learning Evaluation of the Effects of South Africa's COVID-19 Lockdown Measures on Population Mobility. *Machine Learning and Knowledge Extraction*, **3(2)**: 481-506;

Paper 4: **Whata, A.** and Chimedza, C. (2021). Deep learning for Sars Cov-2 Genome Sequences. *IEEE Access*, **9**: 59597-59611;

Journal Manuscripts under Review

Paper 1: **Whata, A.** and Chimedza, C. (2021). Evaluating uses of Deep Learning Methods for Causal Inference. Journal paper under review by *IEEE Access*:

Paper 3: **Whata, A.** and Chimedza, C. (2021). Credibility of Causal Estimates from Regression Discontinuity Designs with Multiple Assignment Variables. Journal paper under review by *Stats — An Open Access Journal from MDPI*

DEDICATION

I would like to dedicate this thesis to my family. Your patience, support, and understanding are gratefully appreciated.

ACKNOWLEDGEMENTS

Firstly, I would like to extend my heartfelt gratitude to my supervisor Dr. Charles Chimedza from the School of Statistics and Actuarial Studies (University of the Witwatersrand). No matter how busy he was, he always made time for me and was a constant source of support and encouragement. I feel privileged to have had such a great supervisor and mentor. I am indeed indebted to him in many ways that words cannot fully express. I acknowledge and thank my employer, Sol Plaatje University for the funding, time, and facilities used throughout the study. I would like to particularly thank Dr Adesuwa Vanessa Agbedahin, Programme Manager: Academic Career Path Development, Sol Plaatje University, for always ensuring that the PhD tuition fees due to the University of the Witwatersrand were always paid on time through the University Capacity Development Grant (UCDG) (Sol Plaatje University). I acknowledge the support of the UCDG towards the article processing charges (APC). I also acknowledge the School of Statistics and Actuarial Studies (University of the Witwatersrand) for support with the article processing charges (APC).

Last but not least, I also thank my family for supporting me throughout my PhD journey and my life in general.

Contents

List of Figures

List of Tables

List of Acronyms

Acc	Accuracy
AE	Autoencoder
ANN	Artificial neural network
ASAMD	Average standardised absolute mean difference
ATE	Average treatment effect
AUC ROC	Area under curve receiver operating characteristic
Bi-LSTM	Bi-directional long short-term memory
BLAST	Basic local alignment search tool
BPTT	Back-propagation through time
BSTSM	Bayesian structural time series model
CGPA	Cumulative grade point average
CNN	Convolutional neural network
CNN-Bi-LSTM	Convolutional neural network bi-directional long short-term memory
COVID-19	Coronavirus disease of 2019
DD	Difference-in-differences
DNA	Deoxyribonucleic acid
DNN	Deep neural networks
GPA	Grade point average
GRE	Graduate record examination (GRE)
INC	Income
IPTW	Inverse probability of treatment weighting
KQE	Kernel quantile estimator
LR	Logistic regression

LSTM	Long short-term memory
LSTMAE	Long-short term memory autoencoder
MCC	Mathews Correlation Coefficient
ML	Machine learning
MP	Matric points
MRDD	Multivariate regression discontinuity designs
NIR	No information rate
NSFAS	National student financial aid scheme (NSFAS)
PN	PropensityNet
PSM	Propensity score matching
RCM	Rubin Causal Model
RDD	Regression discontinuity design
ReLU	Rectified linear unit
RNA	Ribonucleic acid
RNN	Recurrent neural network
RT-qPCR	Reverse transcription-quantitative real-time polymerase chain reaction
SARS CoV-2	Severe acute respiratory syndrome coronavirus 2
Sens	Sensitivity
Spec	Specificity
SUTVA	Stable unit treatment value assumption

CHAPTER 1

Introduction

1.1 Background

Causal inference presents an area where there are opportunities for fruitful collaborations between computer science, statistics, and machine learning (Varian, 2014). For causal inference, statisticians and econometricians have used several tools, including *matching* (Cannas and Arpino, 2019), *instrumental variables* (Hernán and Robins, 2006), *regression discontinuity* (Lee and Lemieux, 2010), and *difference-in-differences* (Lechner et al., 2011). Varian (2014) points out that there have been theoretical advances in literature that have looked at machine learning and its applications to causal inferences, but these advances have not, for example, seen much use in practice in statistics and econometrics.

It should be highlighted that prediction models offered by machine learning models may not necessarily allow someone to make conclusions about causality by themselves, the models may help in estimating the causal impact of an intervention when it arises (Varian, 2014). However, to evaluate causal inference in observational studies, the counterfactual model (Rubin, 1974, 2003, 2005) can be used. The counterfactual model is also referred to as the potential outcomes framework (Holland, 1986; Rubin et al., 2006). The counterfactual model is gaining momentum in statistics and machine learning (Dasgupta et al., 2019; Hernán et al., 2019; Osman and Sakib, 2020). In addition, the approach has been applied in other fields such as sociology, psychology, and political science (Morgan and Winship, 2015). This approach has been used in causation, es-

pecially in philosophy, dating back to the work by Lewis (1974). The work by Lewis (1974) will be similar to the counterfactual model for observational data analysis that will be presented in this thesis. An important assumption of the counterfactual approach is that for a given individual there exists a potential outcome for a given treatment state (Winship and Morgan, 1999; Morgan, 2013). However, for each individual, and at any given time, we can only observe one state of the treatment. For a binary case, the counterfactual says that there are two possible outcomes for each individual. These two possible outcomes are usually labelled as *treatment* and *control*.

Rubin (1974, 2005); Schuler and Rose (2017) indicate that causal effects are often formulated as comparisons between counterfactuals. The counterfactual framework for causal inference presents a mathematical definition of causal effects that envisions that units may occupy multiple causal states, and each has multiple potential outcomes, one for each causal state. The causal effect then becomes the difference between these potential outcomes for two causal states. As units can occupy only one causal state at a time, the remaining potential outcomes for the remaining causal states then become the unobserved *counterfactuals*. According to the *fundamental problem of causal inference* (Holland, 1986), only one of two potential outcomes can be realised for a specific level of treatment on a unit. We define potential outcomes or counterfactuals as follows: a given unit has a potential outcome denoted Y_1 (with treatment) and another potential outcome denoted Y_0 (without treatment). In addition, these can be thought of as outcomes in alternative states of the world. Thus, the difference, Y_1 - Y_0, gives the individual treatment effect. Observing individual treatment effects is impossible according to the fundamental problem of causal inference, and this has led researchers to focus on other treatment effects, such as the average treatment effect (ATE) (Nilsson, 2013). The implications of the fundamental problem of causal inference is that it may not be possible to evaluate causal inference (Holland, 1986). However, the author mentions that by exposing some units to the treatment whilst other units are not exposed and then calculating the *average causal effect*, T, the fundamental problem of causal inference can be addressed. That is, for a population U, and given a treatment variable W_i, $i = 0,1$, where W_1 means that a unit received treatment, W_0 means that a unit did not receive treatment, the average causal effect, T then becomes the expected value of $Y(1) - Y(0)$ over the units in U. This

expected value of the difference $Y(1) - Y(0)$ is given by:

$$T = E(Y(1) - Y(0)), \tag{1.1}$$

which can be expressed as

$$T = E(Y(1)) - E(Y(0)). \tag{1.2}$$

Equation 1.2 shows that by observing different units in a population, we can obtain information about the average causal effects T. This means that the units exposed to the treatment provide information about E(Y(1)) and those that are not exposed to the treatment may be used to provide information about E(Y(0)). Holland (1986) points out that by using Equation 1.2 we have a way of overcoming the difficulty of observing individual treatment effects through the evaluation of the average causal effects over a population U of units. We adopt this approach to evaluate the effect of some potential causes, such as interventions or policy changes, on some outcome. Additionally, we implement the potential outcomes framework or Rubin causal model (Rubin, 1974; Holland and Rubin, 1987) when evaluating causal inference. The potential outcomes framework has been extensively used in the literature. For example, Mithas and Krishnan (2009) deployed the potential outcomes framework to estimate the causal effect of having a Master's in Business Administration (MBA) degree on the salary of information technology (IT) professionals in the United States. Karwa et al. (2011) explored the applicability of Causal Bayesian Networks and Potential Outcomes for a specific transportation safety problem. Imbens (2019) looked at two main frameworks to causal inference in different disciplines, namely: (i) the potential outcome (PO) framework, associated with the work by Rubin (1974) and (ii) the directed acyclic graphs (DAGs). Much of the work on causal inference using DAGs is associated with the work by Pearl et al. (2009). Imbens (2019) found that these two frameworks complement each other and have different pros and cons that make them suitable for answering different causal questions. Also, Rubin (2005) indicated that the potential outcomes strategy for estimating causal effects has achieved greater acceptance.

There are some assumptions that are useful for evaluating causal inference, such as the Stable Unit Treatment Value Assumption (SUTVA) (Rubin, 2004).

Causal inference methods use SUTVA. This assumption requires that the response of a particular unit should depend only on the treatment to which the unit was assigned and not on the treatments of the nearby units. Another useful assumption is the ignorability or unfoundedness assumption, which states that the treatment assignment mechanism is independent of the outcomes Y_i (Rubin, 2004), i.e.,

$$W_i \perp (Y_i(0), Y_i(1)) | X_i \tag{1.3}$$

Estimation of the causal effect of a treatment W_i on an outcome Y_i in observational studies is usually grounded on the *unconfoundedness* assumption (Wooldridge, 2009; Pearl et al., 2009; Imbens, 2019) assumption. According to the assumption of unconfoundedness, we can observe and control all the variables that affect both the treatment W_i and the outcome Y_i.

In this research, we estimate average treatment effects using propensity scores. The propensity score is the conditional probability of assignment to a certain treatment given a vector of observed covariates, features or pretreatment variables (Westreich et al., 2010). Thus, propensity scoring is a statistical technique that is very useful in evaluating treatment effects, especially when using quasiexperimental or observational data (Ali et al., 2019). The main goal of a propensity score analysis is to control for *confounding bias*. Propensity scores control for confounding bias by estimating the probability of treatment given individual covariates such that conditioning on this probability ensures that the treatment is independent of covariate patterns (Westreich et al., 2010). The Rubin Causal Model or potential outcomes framework (Holland, 1986) depends on this assumption. This assumption holds in a randomised experiment without the need to condition on covariates. However, Athey and Imbens (2015) state that the assumption can be justified in observational studies if the researcher is able to observe all the variables that affect the unit's assignment to a treatment.

In this research, we use the potential outcome framework in conjunction with propensity scores to quantify causal effects. By using the potential outcome framework, we have formally articulated the assumptions that are pertinent in estimating the average treatment effects for a given population. The potential outcome framework provides the mathematical link between the data and the causal effect. In this study, we use propensity scores to reduce confounding bias, and also make a valuable contribution by exploring the feasibility of ap-

plying the propensity scores framework and the potential outcomes framework to deep learning algorithms to evaluate causal effects. Wehle (2017) states that deep learning organises algorithms into layers to create an "artificial neural network" that can learn and make intelligent decisions on its own. On the other hand, machine learning uses algorithms to parse data, learn from those data, and make informed decisions based on what it has learned. Although deep learning is a subfield of machine learning, they both fall under the broad category of *artificial intelligence*, and deep learning is what powers the most human-like artificial intelligence.

1.2 Motivation

Although statisticians have made great progress in creating methods that reduce our reliance on parametric assumptions, fruitful collaborations can be made between statistics and machine learning to evaluate causal inference. Authors of causal inference methods papers most often compare their methods to just a few competitors. Typically, these comparisons are made to more established traditional methods and thus perhaps less "cutting edge". As a result, there is not much literature available on the collaborations of statistics and "cutting edge" deep learning methods to evaluate causal inference. This study provides some ideas on how we can combine statistical propensity scores or change point detection (anomaly detection) methods with deep learning algorithms to evaluate causal inference. Some of the most successful machine learning techniques that have been used in the literature to evaluate causal inference include (i) the Bayesian additive regression trees (BART) (Hill, 2011) and (ii) the super learner (Wyss et al., 2018). BART is a sum-of-trees approach that uses a Bayesian prior to prevent overfitting while allowing the model to be very flexible. Wyss et al. (2018) reported that combining the high-dimensional propensity score with the Super Learner was a vital and consistent strategy for reducing bias and mean square error (MSE) in the treatment effect estimates, and it was promising for semiautomated data-adaptive propensity score estimation in high-dimensional covariate data sets. There are propensity score estimation methods that target balance as part of their estimation. For example, the TWANG implementation of generalised boosted modelling (McCaffrey et al., 2004) selects the number of trees to use in computing predicted values from a boosted classification based on balance criteria selected by the user.

The covariate balancing propensity score (Imai and Ratkovic, 2014) incorporates mean balance directly into the estimation of a logistic regression model for the propensity score. There are other methods that bypass a propensity score model and go straight to estimating weights that balance covariates, these methods include entropy balancing (Hainmueller, 2012) and stable balancing weights (Zubizarreta, 2015). In addition, it has been found that these methods implicitly fit a propensity score model. A problem with these methods is that one has to have a good idea about the form of the outcome model. This study is motivated by the need to explore methods that (i) *do not require the correct specification of the functional form of the model*, and (ii) *can handle unstructured data and a large number of covariates*. Urban and Gates (2021) state that, unlike traditional statistical methods, deep learning algorithms can automatically extract their own latent representations of the data that they use to make predictions, thereby saving time by potentially avoiding extensive feature engineering. In addition, Najafabadi et al. (2015) indicate that deep learning algorithms yield results more quickly than standard machine learning approaches, as they can automatically discover high-level and complex abstractions as data representations through a hierarchical learning process. Thus, deep learning algorithms can automatically perform feature engineering, unlike machine learning algorithms that require the researcher to manually select the important features. This means that deep learning algorithms can scan the data to find features that correlate and combine them to enable faster learning without explicitly being told to do so. Regardless of whether one has used a machine learning or deep learning approach, there is a need to statistically assess the balance on the covariates. Thus, statistics and deep learning can be used together, for example, in propensity score estimation, and at the same time, achieve covariate balance. Covariate balance is done to manage the bias-variance trade-off by ensuring balance on as many covariates and their transformations as possible while retaining a high effective sample size.

Advancements in deep learning algorithms such as convolutional neural networks (CNN) or deep neural networks (DNN)), and the cheap availability of high-end general-purpose graphics processing units (GPGPUs) for high-speed computation have considerably improved the state-of-the-art techniques in speech recognition (van den Oord et al., 2020), computer vision (Hassaballah

and Awad, 2020), natural language processing (Otter et al., 2020), etc. However, these successes have not been witnessed in the field of causal inference. Thus, developing faster deep learning techniques to combine with statistical techniques to estimate propensity scores or probabilities of class membership or to detect change points (or anomalies) is vital in evaluating causal inference, and it is the main focus of this research. By incorporating deep learning techniques, statisticians may save months of work.

1.3 Aims and Objectives

1.3.1 Aims

This study seeks to integrate standard statistical analysis such as change point analysis, propensity score estimation, with newer deep learning techniques to evaluate causal inference. The possibility of integrating statistical methods such as potential outcomes and propensity scores with deep learning algorithms to solve a causal inference problem will be explored, with the aim of using the strengths and weaknesses of deep learning and statistical models to complement each other when they are applied in practice.

1.3.2 Specific Objectives

1. (*Paper 1*): To investigate whether or not deep learning methods can be used to estimate propensity scores, which are then used to statistically assess covariate balance and evaluate causal effects. Additionally, the paper evaluates the performance of logistic regression and deep learning algorithms to reduce bias and standard errors of causal effects.
2. (*Paper 2*): To develop a hybrid model that incorporates a deep learning algorithm (long short-term memory (LSTM)) model and a statistical nonparametric estimator, the kernel quantile estimator (KQE) to detect change points in time series data and apply the model to evaluate the causal effects of the COVID-19 interventions imposed by the Government of South Africa.
3. (*Paper 3*): To evaluate credibility of causal estimates from regression discontinuity designs with multiple assignment variables.

4. (*Paper 4*): Develop a hybrid deep learning model for the classification of SARS CoV-2 virus genome sequences and also use approximate statistical tests to compare the predictive performance of deep learning algorithms in classifying SARS-CoV-2 genes among Coronaviruses.

1.4 Assumptions and Scope of Research

This proposed study will focus on evaluating statistical causal inference using the counterfactual approach or the potential outcomes framework. Specifically, we apply the Rubin causal model (RCM) (Holland, 1986), popularly known as the Neyman–Rubin causal model, for the statistical analysis of cause and effect based on the potential outcome framework. We will deploy simulation techniques and generate our own synthetic data to test the proposed algorithms. Also, we will work with publicly available data to evaluate the applicability of the deep learning algorithms considered in this research in the highly complex nature of many real-world problems, where the true data-generating mechanism may be unknown.

CHAPTER 2

Literature Review

2.1 Background

There are several statistical methods that can be used to evaluate causal inference in practice, apart from counterfactual causality. For example, Williams et al. (2018) applied directed acyclic graphs (DAGs) in pediatrics studies, Robins et al. (2000) used propensity scores to calculate inverse probability weights for evaluating treatment effects, Zhao et al. (2016); Pan and Bai (2018); Zhao et al. (2020b); Toulis et al. (2018); Nichols et al. (2019) also used propensity scores to evaluate causal inference. In this thesis, we will use propensity scores to evaluate causal inference. According to Rosenbaum and Rubin (1983), the estimated propensity score e_i, for subject i, $(i = 1, ..., N)$ is defined as the conditional probability of assigning a subject to a treatment given a vector of observed covariates X_i, that is,

$$p(x) = Pr(W_i = 1|X_i = x), \tag{2.1}$$

where $W_i = 1$, means subject i received treatment, $W_i = 0$ means subject i did not receive treatment, X_i is the vector of observed covariates for the i^{th} subject.

Hernán et al. (2004); Brookhart et al. (2010); Nørgaard et al. (2017) report that *confounding bias* presents a primary challenge when evaluating treatment effects using observational studies. Any predictor in observational studies about which we wish to determine the causal effect is called a treatment. A variable that is associated with both the treatment and the outcome variable is a

confounder (Kamangar, 2012). Thus, a variable associated with both the treatment and outcome variables (confounder) may cause inaccurate determination of treatment effects.

Traditionally, statistical methods such as the analysis of covariance (ANCOVA) have been used to adjust for confounding (Allen, 2017). Additionally, Rosenbaum (2002) and Austin (2011) mention that propensity score methods can be used to control the effects of confounding variables. Lanza et al. (2013) state that the advantages of using propensity scores are that: (i) they can yield more accurate causal inferences as they can balance non-equivalent groups that arise from the use of a no-randomized design, (ii) the exact relation between the causes and effects can be determined, and then one can evaluate either the average treatment effect (ATE), the average treatment effect among the treated (ATT). Adjusting for confounding, propensity scores can be used to draw credible causal inferences. Most applications of propensity scores have focused mainly on the medical field (Luo et al., 2010; Inada, 2012). However, propensity scores have recently been used in social and behavioural research (Lanza et al., 2013; Morgan and Winship, 2015).

Austin (2011) offers a comprehensive review of several propensity score-based techniques. The procedure developed by Austin (2011) for estimating causal effects using propensity scores will be adopted. Thereafter, the procedure is applied to deep learning algorithms, as well as logistic regression to estimate propensity scores. The procedure has the following steps: (1) estimate propensity scores using the different methods (deep learning algorithms vs logistics regression), (2) use propensity scores to control for confounding effects, (3) check for covariate balance, and finally, (4) estimate the treatment effects whilst using propensity scores to adjust for confounding.

2.2 Propensity Score Matching

Propensity scores are usually used in practice for matching (Heinrich et al., 2010; Austin, 2011), regression adjustment (Vansteelandt and Daniel, 2014), and weighting (Austin, 2011). For each of these applications, logistic regression has typically been the method for estimating propensity scores (Westreich et al., 2010). For example, when propensity scores are used for matching, a control subject is determined such that it has a propensity score similar to a

propensity score of a subject in the treatment group (Austin, 2011). According to Ramachandra (2018), we can address the problem of estimating counterfactuals for the binary treatment case by matching each subject in the control group that did not receive treatment ($W_i = 0$) with a nearby subject in the treatment group that did receive treatment ($W_i = 1$).

The main goal of PSM is to reduce bias from confounding variables when estimating treatment effects by comparing the outcomes of the subjects who received treatment with the outcomes of the subjects who did not receive treatment. The PSM techniques implement the Rubin causal model for observational studies. There is a likelihood bias in observational studies that results from differences in the outcomes of subjects in the treatment and control groups. These differences depend on certain characteristics that affect whether or not a subject received a given treatment, and not because of the effect of the treatment per se. In contrast, randomised experiments produce unbiased estimates of treatment effects because randomisation ensures that covariates in the treatment groups are balanced on average (Deaton and Cartwright, 2018). For observational studies, the assignment of treatments to subjects under consideration is unfortunately not random (Cochran, 2015). Therefore, PSM attempts to mimic the random assignment to a treatment condition by closely matching the control and treatment subjects. When PSM mimics random assignment, the evaluation of treatment effects can be done by determining the differences in the outcomes of the control and treated subjects. Before matching, one needs to decide on (i) the number of control subjects to be matched to each subject in the treatment group, (ii) the technique for matching, and (iii) the metrics to be used for arriving at a match. Traditionally, one-to-one matching (Olmos and Govindasamy, 2015) has been used, but in practice, especially, in cases where there are small-samples and when treatment cases are "rare", each treatment subject can be matched to two or more control subjects.

Nearest neighbour and optimal matching are two widely used algorithms for PSM (Beal and Kupzyk, 2014). Nearest neighbour or greedy matching assigns a control subject to the closest treatment subject based on their propensity scores. Optimal matching is another matching algorithm that matches control subjects with treatment subjects by minimising the total absolute distance between the propensity scores of control and treatment subjects (Beal and Kupzyk, 2014). The difference between greedy matching and optimal match-

ing is that greedy matching uses a set maximum distance in probabilities or absolute difference in the logit of the propensity scores, whereas optimal matching obtains the lowest possible total distance across the sets of matches in the whole sample. The maximum difference in probabilities for greedy matching is called a calliper. Beal and Kupzyk (2014) state that there is no consistency in practice on the calliper widths that are used, and calliper widths that are 0.2 to 0.55 times the standard deviation of propensity scores is recommended for removing the bias due to the confounding variables. Since there is no consistency in the calliper sizes used in practice, the quality of matching is sometimes affected. This is because choosing a calliper size further away from the treated subjects with respect to their propensity scores increases the risk of estimating biased treatment effects. A narrower calliper can lead to greatly reduced bias and closer matches, but some subjects may not be matched (Lunt, 2014). This suggests that having a narrower calliper may improve the performance of propensity score matching. According to Austin (2011), the choice of a calliper size then becomes a *bias-variance* trade-off.

Stuart (2010) reports that matching is usually done without replacement, as sampling with replacement leads to more than one control subject being matched to more than one treatment subject. In addition, sampling with replacement can lead to violations of the independence-of-cases assumption (Verma and Abdel-Salam, 2019), when traditional statistical analysis is performed. Stuart (2010) recommends using weighting in situations where matching with replacement is carried out.

When a propensity score is used for stratification, a whole sample is divided into strata using the rank-ordered propensity scores, and each stratum is then used for further analysis. Another application area of propensity scores is regression adjustment. In this application, any regression model that estimates the treatment effect includes the propensity score as a covariate (Vansteelandt and Daniel, 2014). Finally, the inverse of the propensity scores also referred to as the inverse probability of treatment weighting (IPTW) can be used to weight observations. The IPTW technique can accommodate several confounders. Each subject in the sample is given a weight that is based on the probability of being exposed to the treatment effect under investigation. By applying this weight, the effect of confounders is effectively removed when performing statistical tests or fitting regression models.

2.3 Covariate Balance

Random assignment is an effective technique for evaluating causal inference. This is because it produces groups that are comparable. With PSM or IPTW there is a need to assess covariate balance and thus ensure the *validity* of causal inference. Because of this, covariate balance should be conducted in most analyses that use PSM or IPTW or stratification. In observational studies, differences in demographics or other baseline characteristics are used to check for covariate balance.

In this thesis, we will adopt an approach for assessing covariate balance that compares means and standardized differences before and after matching. The metric used is the average standardised absolute mean difference (ASAMD) (Girman et al., 2014) between the treatment and control groups after applying propensity score weights. ASAMD takes the average of the absolute values of the standardised difference in means across all covariates. For a detailed discussion of this approach, see Austin (2011). Austin (2011) discourages using statistical tests of significance to check for covariate balance in samples that are matched based on the propensity score because (i) significance levels may be affected by the sample size. This is because after propensity score matching we will have a smaller sample compared to the original sample. Thus, misleading results may be produced if we rely on significance testing to detect an imbalance. For example, nonsignificant differences between groups may be as a result of using a smaller matched sample. On the other hand, statistically significant differences in large samples may simply be due to the power of the test, which may be high despite the close similarities in the covariate means, (ii) covariate balance is a unique property of a sample under consideration and inferring the covariate balance of a super-population may be inappropriate. Moreover, causal inference may be difficult to evaluate if there are differences in the treatment and control groups when there are few successful matches. Therefore it is important to take a closer look at the distribution of subjects in the treatment and control groups. This is because we require that there be an *overlap* in the distribution of the propensity scores across these groups for them to be comparable (McDonald et al., 2013). Additionally, not having enough overlap may result in inaccurate treatment effects (Baser et al., 2007).

Under the ignorability or unconfoundedness assumption (Equation 1.3), the average treatment effect (ATE) can be estimated by:

$$\begin{aligned}\tau(x) &= E[Y(1) - Y(0)|X = x] \\ &= E[Y(1)|W = 1, X = x] - E[Y(0)|W = 0, X = x] \\ &= E[Y|W = 1, X = x] - E[Y|W = 0, X = x].\end{aligned} \tag{2.2}$$

Propensity scores can be used to estimate $\tau(x)$ by considering a set-up where there are N units indexed by $i = 1,\ldots, N$ and $W_i \in \{0,1\}$, a binary indicator for treatments where $W_i = 0$ indicates that unit i received the control and $W_i = 1$ indicates that unit i received the treatment (Athey and Imbens, 2015). Furthermore, if we let X_i be an *L-component* vector of features, covariates, or pretreatment variables which are known to be unaffected by the treatment, then the propensity score defined by Equation 2.1 is the conditional probability of assignment to a certain treatment given a vector of observed covariates, features, or pretreatment variables (Westreich et al., 2010; Athey and Imbens, 2015). Thus, according to Equation 2.1, the propensity score gives the probability of selection into the treatment group. We assume that the propensity score is bounded between zero and one such that:

$$0 < Pr(W = 1|X_i = x) < 1 \tag{2.3}$$

This is also known as the *overlap* assumption or common support condition.. It is infeasible to estimate both $E[Y|W = 1, X = x]$ and $E[Y|W = 0, X = x]$ if the overlap assumption is violated at $X_i = x$.

2.4 Machine Learning and Causal Inference

As discussed earlier, causal inference studies in econometrics (Varian, 2016), statistics (Pearl et al., 2009), biostatistics and epidemiology (Goldstein et al., 2020) have used statistical techniques for causal inference when it is important to answer questions about the counterfactual. The counterfactual approach has several assumptions discussed in Morgan and Winship (2015); Naimi and Kaufman (2015), which, when satisfied, enable the researcher to make conclusions similar to those that would be made from running a randomised experiment. The most important assumption of the counterfactual approach is that for each individual under consideration from a population of interest,

there exists a potential outcome under each treatment state, even though each individual can be observed in only one treatment state at any point in time (Morgan and Winship, 2015).

Athey and Imbens (2015) indicate that the literature on machine learning (ML) techniques focuses on how they are used for prediction. Most of these techniques cannot be used directly for causal inference, but they can be used in the process to estimate propensity scores that are then used to infer causal effects. Logistic regression has generally been used to estimate propensity scores (Cepeda et al., 2003; Austin, 2011; Westreich et al., 2010). There are some disadvantages of using logistic regression compared to ML techniques. For example, logistic regression requires assumptions pertaining to the selection of variables, the specification of the functional form including the distribution of variables, and the specification of interactions (Wright, 1995). When these assumptions are satisfied, covariate balance can be achieved by conditioning on the propensity score, resulting in unbiased estimates of treatment effects (Lee et al., 2010; Harder et al., 2010; Ali et al., 2019).

ML methods have recently been introduced to deal with situations where there are many covariates (Athey and Imbens, 2017). The proposed estimators closely mimicked those developed in the literature with a fixed number of covariates. Controlling for a large number of confounding variables helps to formulate credible identification assumptions. According to Athey and Imbens (2017), there are three methods that have gained popularity in ML in dealing with many covariates namely: *propensity scores* (Rosenbaum and Rubin, 1983), *regularised regression* (Blei, 2015) and *Balancing and Regression* (Doudchenko and Imbens, 2016). Hahn (1998) and Hirano et al. (2003) have focused on estimators that utilise the propensity score for a fixed number of covariates and have found that these lead to semiparametrically efficient estimators for the ATE. Propensity scores were very popular, but they have one limitation in that they are only applicable to *treatment-control* studies that have a binary treatment. However, propensity score methodologies that include treatment regimens that are not binary have been proposed (Zhao et al., 2020b) and this has resulted in the expansion of the use of propensity scores.

Propensity scores can also be estimated using random forests, boosting, or LASSO. The use of random forests, and lasso for example, for estimating the propensity score has focused mainly on a few covariate cases. Athey and

Imbens (2017) report that when these methods were applied to cases where there are several covariates, they have yielded relatively poor properties, as they do not necessarily specify the variables that are associated with both the treatment and the outcome variables. Additionally, Goller et al. (2020) mention that using random forests to estimate propensity scores may result in poor performance when the treatment groups and the control groups are unbalanced. Goller et al. (2020) also point out that with imbalanced data, the random forest method cannot manage to remove *selection bias*, and this results in estimates that are not credible.

ML methods have opened avenues to process and infer causal inference. For example, Prosperi et al. (2020) report that data-driven machine learning approaches are increasingly being adapted to model causal inference and uncover new causes of say, a disease or assess treatment effects. ML tasks are predictive or descriptive in nature. However, in this study, we investigate how we can use machine learning methods to evaluate causality. McConnell and Lindner (2019) have used ML methods to estimate ATE and have offered comparisons to traditional methods to estimate treatment effects based on regressions. Their results showed that ML methods yielded treatment effects that had smaller biases compared to treatment effects obtained from regression-based approaches. Additionally, for some scenarios, ML methods demonstrated substantial bias reduction (McConnell and Lindner, 2019).

This study explores how we can adapt deep learning, a subfield of machine learning, to evaluate causal inference. One way in which we can link deep learning to causal inference is through propensity scores. Thus, we propose using deep learning techniques to estimate propensity scores instead of using traditional logistic regression. The estimated propensity scores are then used to estimate ATE. Deep learning algorithms are a subset of ML algorithms that can perform much better on unstructured data (Mathew et al., 2020). Furthermore, deep learning techniques have outperformed some current ML techniques (Mathew et al., 2020; Singaravel et al., 2018; Mathew et al., 2020). Deep learning techniques allow computational models to learn features progressively from data at multiple levels. Also, Mathew et al. (2020) state that deep learning achieves higher power and flexibility due to its ability to process a large number of features when it deals with unstructured data. Deep learning algorithms are suitable for unstructured data as they are capable of

extracting features progressively from layer to layer.

2.4.1 The Cross-Entropy Loss Function

The cross-entropy loss function is almost the sole choice for classification tasks in practice. This loss function is used to estimate the degree of inconsistency between the predicted value $\hat{\mathbf{Y}}$ of the model and the true value $\boldsymbol{Y}$. It is a non-negative real-valued function that is usually represented by *L(**Y**, f(**X**))*. The setup for a classification task is that we are usually given a clean data set $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$, where $\boldsymbol{X} \subset \mathbf{R}^d$ represents the feature space and $\mathbf{Y} = \{1, ..., K\}$ is the label space (Zhong and Zhao, 2020). Also, each $(x_i, y_i) \in (\boldsymbol{X} \times \boldsymbol{Y})$. Thus, we define the classification task an optimisation problem in which the objective function can be expressed as the cross-entropy between $\boldsymbol{Y}$ and $\hat{\mathbf{Y}}$. A classifier, is therefore, a function that maps input feature space to the label space f:x$\rightarrow R^K$.

For multiclassification the cross-entropy loss function is defined as :

$$L(Y, f(X)) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\log(f_j;\theta), \tag{2.4}$$

where θ is the set of parameters of the classifier, y_{ij} corresponds to the j^{th} element of *one-hot* encoded label of the $x_i, y_i \in \{0,1\}^K$ such that $\mathbb{1}^T y_i = \mathbb{1} \forall i$ and f_j denotes the j^{th} element of f. Note that $\sum_{j=1}^{N} f_j(x_{ij};\theta) = 1$, and $f_j(x_{ij};\theta) \geq 0$, $\forall j, i, \theta$ as the output layer is a *softmax* (Zhang and Sabuncu, 2018). The *softmax* function is a function that converts a vector of K real numbers to a vector of K values that add up to 1. Thus, the *softmax* takes as input zeros or positive or negative numbers and transforms them into values that are between 0 and 1, in order to interpret the values as probabilities. A mathematical expression of *softmax* is given in Equation 3.7.

Equation 2.4 represents the deviation of the predicted outputs (probabilities) from the target, averaged over all samples. The target vector and the predictions represent the probability mass function of the target and the predicted classes, respectively (Sivaram et al., 2020). Sivaram et al. (2020) point out that using the above context, a deep neural network as an example, can be described as a classification model $\mathcal{N}$ with a predefined architecture $\mathcal{A}$ and a set of parameters Θ that express the output as a function of the input as follows:

$$\hat{Y} = \mathcal{N}(X; \mathcal{A}, \Theta) \tag{2.5}$$

$\mathcal{A}$ will represent the design choices of the deep neural network and Θ are the parameters that will be tuned during training. The design choices $\mathcal{A}$ usually include (i) how the layers are organised, (ii) the activation function which could be a logistic function, hyperbolic *tahn* or rectified linear unit *ReLU* (Jagtap et al., 2020), (iii) the number of layers of the deep neural network and (iv) the number of nodes in each layer, n_l, $l = 1, 2, 3, ..., L$. The parameter Θ will include the weights W_l and the biases b_l of each layer, $l = 1, 2, 3, ..., L$.

2.4.2 Deep Neural Networks for Classification

The backpropagation (BP) neural network is one of the most widely used neural network models for classification problems. This is due to its strong learning ability (Hu et al., 2020). Backpropagation is an essential process for neural network training. It involves fine-tuning the weights of a neural network based on the error rate (loss) obtained in the previous period (iteration). Proper weight adjustment ensures lower error rates, making the model more reliable. Figure 2.1 shows an example of a general fully connected DNN, for classification that incorporates backpropagation (Ramachandra, 2018).

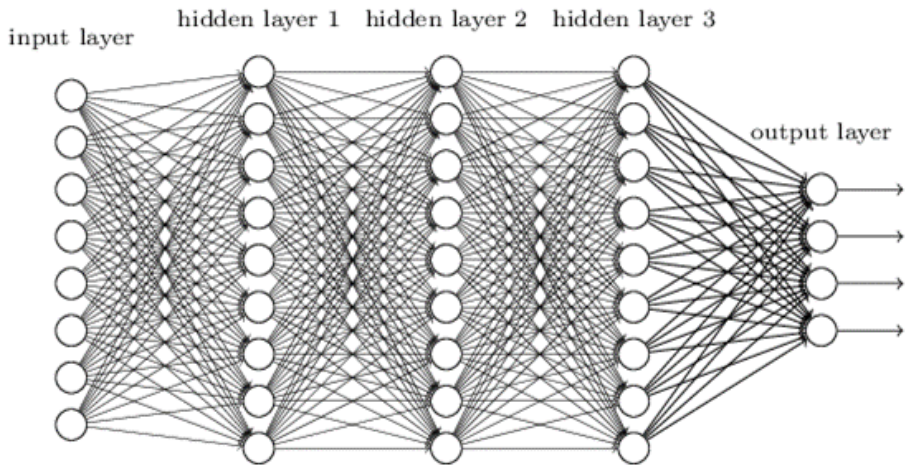


Figure 2.1: Fully connected DNN for classification.

The DNN learning process such as the one in Figure 2.1 involves two important steps: the first step is the forward propagation phase of the training data, which first imports the raw data from the input layer. The second stage involves the

backpropagation of the error signal. A detailed description of these two steps can be found in Goodfellow et al. (2016) and Hu et al. (2020).

For multiclass models, the last layer of the DNN (Figure 2.1) is a *softmax layer* Sivaram et al. (2020) that retains the probabilities of each class, with the target class having the highest probability (Nwankpa et al., 2018). The softmax layer produces an output between 0 and 1, with the sum of the probabilities equal to 1. In this study, we are concerned with a binary classification problem where subjects are assigned to the treatment group or control group in estimating propensity scores. Thus, the output value of the class is either 0 or 1. A *sigmoid* activation function can also be used in the output layer to predict class values of 0 or 1. The model is optimized using the binary cross-entropy loss function which is suitable for binary classifcation problems and the efficient **Adam** (Kingma and Adam, 2014a) version of gradient descent as the optimiser. Propensity scores can be estimated by feeding the inputs which are the covariates $\mathbf{X}$ and the outcome $\mathbf{Y}$ across all units into the deep neural network. Thus, instead of predicting the class output values, 0 or 1, the last softmax layer of the network can be adapted and trained to give a probability between 0 and 1 for each new/test unit. These probabilities are the estimated propensity scores. In this way, the procedure becomes a generalisation of the logistic regression function. The estimated propensity scores can be checked to ensure that they are balanced between the treatment and comparison groups. This will lead us to check if the covariates are indeed balanced across treatment and control unit groups within a particular interval of the propensity score (Ramachandra, 2018).

The hidden (or intermediate) layers can take any form, and the output of each layer is typically passed through nonlinear functions (Ramachandra, 2018). These nonlinear functions are referred to as *activation functions*. Examples of activation functions include the logistic function, the hyperbolic tangent *tahn*, and the rectified linear unit $ReLU$. Their main purpose is to convert an input signal from an input node in a DNN to an output signal. That output signal is then used as input in the next layer in the stack. A neural network without an activation function would not be able to learn and model other complicated kinds of data such as images, videos, audio, speech, or even numeric data. In addition, a neural network without an activation function would simply be a linear regression model, which has limited power and poor performance for

non-linear relationships. In this work, we use the *ReLU* activation function, as it substantially reduces the computational cost of training and guarantees faster computation and convergence (Nwankpa et al., 2018). *ReLu* offers better performance and generalization in deep learning compared to the *sigmoid* and *tahn* activation functions. For a detailed discussion and comparison of the different activation functions, see Nwankpa et al. (2018).

Classification tasks typically involve *feature extraction* followed by *classification*. The hidden layers are used to extract the relevant features of the data and the final layer is used for classification. Karsoliya (2012) points out that if the neural network is to be used for classification into groups, then it is preferable to have one output neuron for each group to which an input item is assigned.

2.4.3 Deep Neural Networks for Propensity Scores

Machine learning algorithms are typically not designed to estimate class-membership probabilities per se, rather they are designed to minimise misclassification rates (Cannas and Arpino, 2019). However, machine learning algorithms can be tweaked to estimate class-membership probabilities. For example, Lee et al. (2010); Westreich et al. (2010); Setoguchi et al. (2008); Wyss et al. (2014) used machine learning algorithms to estimate class-membership probabilities and found that machine learning algorithms work rather well in this regard and can be successfully used to estimate propensity scores using the class-membership probabilities. Estimating class-membership probabilities using deep neural networks has not been widely used. Therefore, the literature on the use of deep neural networks to estimate propensity scores using class-membership probabilities is still limited. In this study we assess the performance of deep learning algorithms in estimating propensity scores compared to logistic regression. Thus, there is a need to develop statistical deep learning algorithms to estimate propensity scores. That is because unlike logistic regression, deep learning algorithms do not require assumptions regarding variable selection, the functional form, distribution of variables, and specification of interactions. In addition, it is necessary to investigate how deep learning algorithms compare with logistic regression in reducing absolute bias when estimating average treatment effects.

2.5 Time Series and Causal Inference

Deep learning methods can be used to evaluate causal inference by exploiting change points in time series data. The use of deep learning for change point detection is inspired by successes in the use of machine learning methods in anomaly detection. Real-world examples where machine learning algorithms have been used for anomaly detection include cyber-attacks (Kozik and Choraś, 2014), fraudulent insurance claims (Roy and George, 2017), and detecting fraud transactions (Shpyrko and Koval, 2019). Causal inference in time series data is an important problem in fields such as neuroscience (Eichler, 2013), and finance (Chen, 2020). Traditional statistical methods have used regression models for this problem. Chikahara and Fujino (2018) point out that the accuracy of these statistical time series causal methods depends greatly on whether or not the model can be well-fitted to the data, and the selection of an appropriate regression model. With increasing developments of powerful artificial intelligence algorithms, this study explores the use of deep learning methods to evaluate causality in time series data. Chen (2020) indicates that providing insight into causality information through data is of paramount importance, and most machine learning methods fall short in this regard. The use of deep learning methods to evaluate causal inference is motivated by the fact that statistical tests for causality, such as the Granger causality test for causal inference from time series data (Eichler, 2013), are significantly harder to construct. According to Chattopadhyay (2014), it is important to design an efficient causality test that may be carried out in the absence of restrictive presuppositions on the underlying dynamical structure of the data at hand. nonparametric approaches for causal inference in time series have been used in Schreiber (2000) and Tsapeli et al. (2017). This study proposes a nonparametric framework for inferring causality in time series data that does not make any restrictive assumptions and requirements of prior knowledge of the data. The framework uses a deep learning algorithm to detect change points. At a change point, we then evaluate the causal effect of the change point. For example, the change point could represent an intervention or a policy change, and the goal would be to quantify the causal effect of the intervention.

The long-short term memory (LSTM) algorithm has been used to detect smaller but sustained changes or anomalies in time series data. Several authors

have used the LSTM algorithm in the real world for anomaly detection. For example, Bontemps et al. (2016) combined an LSTM and a recurrent neural network (RNN) to detect anomalies using a time series version of the KDD 1999 data set (Hettich, 1999). Wolpher (2018) Wolpher (2018) used replicator neural networks, isolation forests, and a long short-term memory autoencoder for anomaly detection on unstructured time series data, to assess whether the algorithms were effective in detecting intrusions in network traffic. Dutta et al. (2020) also successfully developed a deep learning ensemble algorithm that combined a DNN and LSTM, followed by a meta-classifier for network anomaly and cyberattack detection. Because of the successes of deep learning algorithms in anomaly detection, we will adapt the LSTM and autoencoder for change point detection in time series data. The use of LSTMs and autoencoders algorithms has been shown to provide higher anomaly detection rates as well as reduce the processing time significantly (Elsayed et al., 2020). In addition, LSTMs can be used in sequences with varying lengths without making any assumptions about the number of previous points that are needed to make predictions (Jansson, 2017). LSTMs are structured to exploit temporal dependencies in sequential data, and they do not assume any functional form between the outcome variables and regressors or explanatory variables (Poulos, 2017).

After using LSTMs to detect change points, the causal effect of the change point(s) can be evaluated using existing packages such as the **CausalImpact R** package (Brodersen et al., 2015).

2.6 Regression Discontinuity Designs for Causal Inference

A regression discontinuity design (RDD) is a statistical approach to inferring causal effects. Arai et al. (2019) report that discontinuities in regression functions that are caused by the assignment variable can be used to determine causal effects. ATE at a discontinuity have been determined by using RDD (Reardon and Robinson, 2012; Papay et al., 2011; Imbens and Zajonc, 2011). RDDs have been used to estimate causal parameters of interest by contrasting the left and right limits of some conditional mean functions. We add to the growing literature on evaluating causal inference by combining deep learning

and statistical techniques to automatically detect discontinuities in data, and hence estimate the treatment effect at a discontinuity. A survey of the early literature on RDDs can be found in (Imbens and Lemieux, 2008).
The RDD approach is tackled in the context of causal effects using the potential outcomes framework. For unit i, there are two potential outcomes, $Y_i(0)$ and $Y_i(1)$, with the causal effect then defined as the difference $Y_i(1)$ - $Y_i(0)$ and the observed outcome being equal to:

$$Y_i = (1 - W_i)Y_i(0) + W_iY_i(1) = \begin{cases} Y_i(0) & if\ W_i = 0, \\ Y_i(1) & if\ W_i = 1, \end{cases} \tag{2.6}$$

and $W_i \in (0, 1)$ (Imbens and Wooldridge, 2009).

RDD is described as a quasi experimental design in observational studies due to its reliance on a cut-off point on a continuous baseline variable to assign individuals to treatment (Linden and Adams, 2012). There are assumptions that have to be met when using both single and multivariate assignment variables, namely:

(i) The cut-off point which determines treatment assignment is exogenously set;

(ii) Potential outcomes are continuous functions of the assignment variable at the cut-off point, that is, $E[Y_i(0)|X = c]$ and $E[Y_i(1)|X = c]$ are continuous in c (Imbens and Lemieux, 2008; Reardon and Robinson, 2012) and

(iii) The functional form of the model is correctly specified.

Assumption (iii) is a very strong assumption that assumes that when using a local linear regression method to estimate causal effects to the right and left of the cut-off point, the underlying regression model is linear in the assignment variable X, as was originally postulated by Thistlethwaite and Campbell (1960). The consequence of assumption (iii) is that misspecification of the functional form may lead to bias in the treatment effects (Lee and Lemieux, 2010; Reardon and Robinson, 2012). In practice, when the regression function is not linear over the whole range of X, the estimation range is restricted only to values that are closer to the cut-off point, where the linear approximation of the regression function is less likely to lead to a large bias in the RDD estimates (Lee and Lemieux, 2010).

Most studies on RDD analysis focus on a single continuous assignment variable (Papay et al., 2011). However, Reardon and Robinson (2012); Wong et al. (2013) indicate that in practice, two or more continuous variables can be used to assign units to treatments. This means that treatment effects can be evaluated in cases where we have two or more cut-off points and two or more assignment variables as compared to having just one cut-off point using a single assignment variable. The approach that uses two or more assignment variables is referred to as multivariate regression discontinuity designs (MRDD). MRDD raise challenges that are different from those identified in traditional RDD, as treatment effects for MRDD can be identified across multiple cut-off frontiers as opposed to a single point along the assignment variable in RDD (Wong et al., 2013).

Estimation of the causal (treatment estimates) is called *primary analysis*. After evaluating estimates of treatment effects, supplementary analyses that go beyond simply obtaining the causal estimates can be carried out (Athey and Imbens, 2017). Supplementary analyses are intended to give credibility to the causal estimates obtained from primary analysis.

2.6.1 Supplementary Analyses

The causal estimates obtained in RDD have limited external validity as they are only identified for observations in the immediate vicinity of the cut-off scores (Papay et al., 2011). One of the goals of this study is to assess the credibility of the causal effects estimated by the RDD. Therefore, it is important that after estimating the average treatment effect (τ_{RDD}), we then check for the plausibility of the assumptions of the RDD estimates. With more credible estimates, inference about causality can reduce the reliance of these causal estimates on the following modelling assumptions (Reardon and Robinson, 2012):

1. *the cut-off scores determining treatment assignment are exogenously set*,
2. *potential outcomes are continuous functions of the assignment scores at the cut-off scores* and
3. *the functional form of the model is correctly specified*.

These assumptions will be assessed where there is a treatment assignment discontinuity using the following supplementary analyses techniques: *McCrary*

Test (McCrary, 2008), *placebo analysis*, and *robustness and sensitivity* proposed by (Athey and Imbens, 2017). The findings of the supplementary analyses will add to the debate on the plausibility and credibility of causal estimates when using RDD. Additionally, the findings will equip policy makers and decision-making institutions with more tools to estimate the average effects of the treatments of interventions.

The primary aims and objectives of this thesis were achieved by addressing the specific chapter research aims and objectives, which included the application of deep neural network (DNN) to real world data and verifying its accuracy in estimating propensity scores, and the development of a hybrid model that consisted of a long-short term memory autoencoder (LSTMAE) and the kernel quantile estimator (KQE) algorithm to automatically detect change points from a time series or a sequence of values. At the change point, the propensity scores-potential outcomes framework was applied to estimate the causal effect of a change-point or intervention using the Bayesian structural time series model (BSTSM) that has fewer assumptions. Several studies were carried out to address the objectives and goals of this thesis. The outcomes of these studies are discussed in the enclosed research articles in *Chapters 3 to 6*.

CHAPTER 3

Evaluating uses of Deep Learning Methods for Causal Inference

---

ALBERT WHATA [1,*] and CHARLES CHIMEDZA [2]

[1] School of Natural and Applied Sciences, Sol Plaatje University

[2] School of Statistics and Actuarial Science, University of the Witwatersrand

Journal under review: *IEEE Access*.

Statement of Contributions of Joint Authorship

Albert Whata **(Candidate)**
Conducted the research, writing and compilation of manuscript);

Charles Chimedza **(Supervisor)**

Supervised, edited and coauthor of the manuscript.

This Chapter is an exact copy of the journal paper mentioned above.

ABSTRACT

Logistic regression (LR) is a very popular method for estimating propensity scores in observational studies. We evaluated how the deep learning methods perform in estimating propensity scores and average treatment effects. Using simulations, we evaluated the performance of the deep neural network (DNN), Autoencoder (AE), PropensityNet (PN), and LR in evaluating causal inference. In addition, we evaluated covariate balance using the propensity scores derived from these methods by employing the absolute standardized average mean difference (ASAMD). The performance of the DNN, AE, PN, and LR was evaluated using the metrics: absolute bias, standard errors, ASAMD, area under the curve receiver operating characteristic (AUC-ROC), classification accuracy, Cohen's Kappa, No Information Rate (NIR), and 95% coverage probability. Monte-Carlo simulations were employed to simulate the data sets that were used in this paper. In addition, a real-world data set from the Atlantic Causal Inference Conference (ACIC) Data Challenge 2019 was used to evaluate how the methods perform in the real world. Overall, the DNN performed better than PN, LR, and AE in reducing absolute bias using the simulated data sets. Furthermore, DNN produced better values for classification accuracy, receiver operating characteristic area under curve (AUC-ROC), Cohen's Kappa, and 95% CI coverage as sample sizes increased from N = 500 to *N = 2000*. On the other hand, LR achieved covariate balance on average for the different sample sizes. The DNN also gave excellent predictive performance when it was applied to a real-world data set.

3.1 Introduction

Past research attempting to estimate causal inference in statistics (Reiter, 2000; Rubin, 2003), econometrics (Heckman, 2008) and biostatistics (Egleston et al., 2007) just to mention a few have focused more on the potential outcomes framework or the Rubin Causal Model (RCM) (Holland, 1986). The RCM

considers a set up where there are two treatment groups, and two potential outcomes $Y_i(0)$ and $Y_i(1)$ for unit i (Athey et al., 2017). $Y_i(1) - Y_i(0)$ is the unit level effect of the treatment. Given a binary treatment indicator, $W \in \{0, 1\}$, then $W_i = 0$ indicates that unit i was given the control, $W_i = 1$ indicates that unit i was given the active treatment. Subsequently, $Y_i(0)$ or $Y_i(1)$ are the outcomes when unit i either receives the control or the active treatment. Therefore, it is not possible to observe $Y_i(0)$ and $Y_i(1)$ on the same unit at the same time. This is referred to as the *fundamental problem of causal inference* Holland (1986). Because of the fundamental problem of causal inference, Holland (1986) states that the average causal effect is then used to estimate the average treatment effects (ATEs). Since the outcomes $Y_i(1)$ and $Y_i(0)$ are observable, then an estimate of ATE is given by: $\tau = E(Y_i(1))$ - $E(Y_i(0))$.

Titiunik (2015) reports that determining causal inference then becomes a search for assumptions under which we can infer the values of the unobserved counterfactual from the observed data. One way to guarantee that the counterfactual approach works is to ensure that the only difference present between the control and treatment groups is the desired treatment effect. This ensures that all extraneous variables are either controlled or eliminated by random assignment (Edgington, 1985). Due to ethical considerations, a random assignment may not always be accomplished. Olmos and Govindasamy (2015) states that assigning individuals randomly to either the control condition or treatment condition may be unethical. For example, individuals assigned to the control group may not benefit from an important resource (e.g., receiving antiretroviral drugs that save lives) compared to those in the treatment group who receive the important resource. Propensity scores provide a useful way to assign individuals to different treatment conditions when random assignment fails due to the presence of ethical constraints. Generally, propensity scores are estimated using logistic regression.

3.2 Theoretical Background

We estimate average treatment effects using propensity scores by considering a set-up where there are N units indexed by $i = 1,\ldots, N$ and $W_i \in \{0,1\}$, a binary indicator for treatments where $W_i = 0$ indicates that unit i received

the control treatment and $W_i = 1$ indicates that unit i received the treatment. Furthermore, if we let X_i be an *L-component* vector of features, covariates or pretreatment variables which are known not to be affected by the treatment, we can formally define the propensity score as $p(x) = Pr(W_i|X_i = x)$ (Athey and Imbens, 2015). Thus, the propensity score is the conditional probability of assignment to a certain treatment given a vector of observed covariates, features, or pretreatment variables (Westreich et al., 2010). Propensity scoring is a statistical technique that is very useful in evaluating treatment effects, especially when using quasi-experimental or observational data (Ali et al., 2019). However, two vital assumptions connected to causality need to be considered before using propensity scores, and these are the *Ignorable Treatment Assignment* assumption (Austin, 2011), and the *endogeneity* (Antonakis et al., 2014) assumption.

The counterfactual approach depends on the (*"unconfoundedness"*) assumption stated below (Athey et al., 2016a):

$$W_i \perp (Y_i(0), Y_i(1))|X_i \tag{3.1}$$

The assumption indicates that the outcomes $(Y_i(1), Y_i(0))$ are independent of W_i given the covariates X_i (Olmos and Govindasamy, 2015). This assumption is also stated as the *Ignorable Treatment Assignment Assumption* (Austin, 2011). Confounding bias is usually controlled by using propensity scores. Propensity scores achieve this by estimating the probability of W_i given the covariates X_i. The Rubin Causal Model or potential outcomes framework (Holland, 1986) depends on this assumption. This assumption holds in a randomised experiment without the need to condition on covariates. However, Athey and Imbens (2015) state that the assumption can be justified in observational studies if the researcher is able to observe all the variables that affect the assignment of the unit to a treatment. Kang et al. (2016) states that it is important to ensure that the propensity score be strictly between 0 and 1. This requirement is known as the *positivity* assumption. Estimates of treatment effects may be biased when the *positivity* assumption (Petersen et al., 2012) is not met.

Cannas and Arpino (2019) states that if Equation 3.1 holds conditional on the set of covariates (X_i), then it should also hold conditional on the propensity score $(p(x))$. This means that if the distribution of the propensity scores

is balanced between the control and treatment groups, then the distribution of the observed covariates will also be balanced in expectation between the groups. Imai and Ratkovic (2014) refers to this as the *balancing property* of the propensity score. Subsequently, the one-dimensional propensity score can be used in place of the multivariate set of observed variables, X_i's, to achieve covariate balance. This paper focuses on the inverse propensity of treatment weighting to adjust for confounding. There are basically two methods that can be considered to implement covariate balance, namely: *propensity ac matching* (PSM) (Beal and Kupzyk, 2014) and *inverse probability of treatment weighting* (IPTW) (Li et al., 2018a; Austin and Stuart, 2015). PSM involves finding units with the same propensity scores in the control and treated groups, then forming a matched data set of the original data (Olmos and Govindasamy, 2015). If the *unconfoundedness* and *positivity* assumptions hold (Stuart, 2010), then the average treatment effect is estimated by comparing the control and treatment groups in the matched data set. In addition, IPTW employs propensity scores to weight the observations to achieve covariate balance. When calculating the ATE, units in the treatment group are assigned a weight equal to $1/p(x)$. On the other hand, a weight of $1/(1\text{-}p(x))$ is assigned to units in the control group (Cannas and Arpino, 2019). Attaching these weights to the treatment and control groups, respectively, ensures that the covariates distributions of the treatment and control groups are comparable. Therefore, the unbiased treatment effects estimates under unconfoundedness (Austin, 2011) are obtained from the weighted differences in the average outcomes of the treated and control observations. The IPTW will be employed to evaluate covariate balance because according to Cannas and Arpino (2019), IPTW gives a lower bias compared to PSM.

3.2.1 Problem Statement

Propensity scores are generally estimated using logistic regression. Estimation of propensity scores using logistic regression requires assumptions regarding (i) how variables are selected, (ii) specification of the correct functional form, (iii) statistical distributions of the variables, and (iv) interactions are specified (Lee et al., 2010). If the assumptions are not met, one may obtain biased estimates of treatment effects due to not achieving covariate balance. Propensity scores are primarily used to achieve covariate balance between the treatment group

and the control group to obtain valid and unbiased estimates of the treatment effect. According to Imai and Ratkovic (2014), propensity scores are used to adjust for observed confounding through matching, subclassification, weighting, regression or their combinations. Main effects logistic regression propensity score models have generally been found to provide acceptable covariate balance (Lee et al., 2010). However, as models become more complex with interactions and nonlinear terms, the logistic regression propensity score models have produced large biases when estimating average treatment effects (Lee et al., 2010). Machine learning algorithms have generally been employed to perform classification as well as for prediction. Classification tasks use an input data set $\mathcal{D}$ of size N and the corresponding target classes: $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where $\boldsymbol{X} \subset \mathbb{R}$ represents the feature space and $\mathbf{Y} = \{1, ..., K\}$ is the label space (Zhong and Zhao, 2020). We define the problem of classification as one of mathematical optimisation. The loss (objective) function can be expressed as the cross-entropy between $\mathbf{Y}$ and $\hat{\mathbf{Y}}$. The cross-entropy loss function is almost the only choice for classification tasks in practice. This loss function, $L(\boldsymbol{Y}, f(\boldsymbol{X})) \geq 0$, estimates the extent to which the true value of a model $\boldsymbol{Y}$ differs from the predicted value $\hat{\mathbf{Y}}$. Ideally, we are given $\mathcal{D}$, where each $(\mathbf{x}_i, y_i) \in (\boldsymbol{X} \times \boldsymbol{Y})$ and a classifier is therefore a function that maps the input feature space to the label space $f{:}\mathbf{x} \rightarrow R^K$. Cannas and Arpino (2019) states that machine learning algorithms are typically designed to minimise misclassification rates and not to estimate class-membership probabilities. However, the classification task can be used to estimate the probabilities of class-membership. Example, Setoguchi et al. (2008); Westreich et al. (2010); Lee et al. (2010); Wyss et al. (2014) used machine learning algorithms to estimate class membership probabilities and found that machine learning algorithms work rather well in this regard and can be used successfully to estimate propensity scores using class membership probabilities. Estimating class membership probabilities using deep neural networks has not been widely used. Therefore, the literature on the use of deep learning methods to estimate propensity scores using class-membership probabilities is still limited. It is vital to investigate whether deep neural networks can reduce or eliminate the reliance on logistic regression assumptions that include the functional form, variable selection, distribution of variables, and the specification of interactions. In addition, statistical machine learning techniques such as deep neural networks are needed to estimate propensity scores and evaluate whether these deep neural networks

perform better than logistic regression in the estimation of propensity scores and bias reduction when estimating average treatment effects.

3.2.2 Related Work

Researchers have made several efforts to use the power of machine learning techniques for causal inference problems (Westreich et al., 2010). Cannas and Arpino (2019); de Vries et al. (2018) have compared machine learning algorithms for modelling propensity scores. Brown et al. (2018) reported on the full potential of machine learning for estimation of average treatment effects with propensity score methods and found that machine learning methods can be helpful in high-dimensional data sets (i.e., large number of covariates and observations). Zhao et al. (2016) proposed matching methods based on the random forest to obtain covariate balance between the control and treatment groups for arge observational study data. The authors pointed out that their approach provided better estimates of the effect of treatment. Westreich et al. (2010) concluded in their paper that although the assumptions of logistic regression are well understood, those assumptions are frequently ignored. They noted that boosting (meta-classifiers) (Zhu et al., 2015) and, to a lesser extent, decision trees (particularly CART) (Lee et al., 2010), appear to be the most important in propensity score analysis, but extensive simulation studies are needed to establish their utility in practice. Yuan et al. (2020) constructed a normalized empirical probability density distribution (NEPDF) matrix and trained a convolutional neural network (CNN) on the NEPDF matrix for causality predictions. The authors demonstrated that the use of the NEPDF matrix enabled CNN to work very well for image classification problems for the task of causal inference. By using experiments on simulated and real data, their method generally worked well on a diverse set of input data types.

Śmieja et al. (2018) proposed an approach to adapt neural networks to process incomplete data, and they found that neural networks give results comparable to methods that require complete data in training. Farrell et al. (2018) states that there has been limited adoption of deep learning algorithms in the social sciences due to a lack of sufficient data. Ramachandra (2018) estimated propensity scores through simulation studies using a deep neural network referred to in their research as PropensityNet instead of traditional logistic re-

gression and verified the superior performance of their proposed PropensityNet over logistic regression in estimating propensity scores. This paper extends the work of Ramachandra (2018) by developing a deep neural network (DNN) that aims to improve PropensityNet performance. The aim is to determine whether the deep learning methods; DNN, Autoencoder (AE), and PropensityNet can be used in estimating propensity scores. Furthermore, the article seeks to assess whether deep learning methods are better at reducing bias in estimated average treatment effects compared to logistic regression. Specifically, the paper makes the following contributions.

(a) Estimate propensity scores and assess covariate balance for logistic regression, deep neural network (DNN), PropensityNet, and the Autoencoder (AE),

(b) Compare the performance of deep learning methods and logistic regression in estimating the average treatment effects using simulation techniques.

(c) Assess the performance of the deep learning methods when they are applied to a real-world data set.

3.3 Research Method

3.3.1 Data Generation Using Monte-Carlo Simulations

To assess the performance of deep learning models and logistic regression, we performed a series of Monte Carlo simulation experiments that follow the structure of Setodji et al. (2017) adapted from Setoguchi et al. (2008); Lee et al. (2010); de Vries et al. (2018); Cannas and Arpino (2019). Samples of sizes *N= 500, N= 1000 and N = 2000*, a binary treatment W_i having $p(W_i) \approx 0.5$, and a binary outcome Y_i having $p(Y_i) \approx 0.02$ were simulated. In addition, 15 covariates were generated as standard normal random variables (Lee et al., 2010; Setoguchi et al., 2008; de Vries et al., 2018; Gharibzadeh et al., 2018). Five of these covariates $X_2, X_4, X_9, X_{10}, X_{11}$ were continuous variables. Correlations were established between some of the variables such that: $\rho(X_3, X_8) = 0.2$, $\rho(X_{12}, X_{14}) = 0.9$, $\rho(X_4, X_9) = 0.9$, $\rho(X_1, X_5) = 0.2$, $\rho(X_2, X_6) = 0.9$, and $\rho(X_{11}, X_{13}) = 0.2$. Thereafter, the ten covariates $(X_1, X_3, X_5, X_6, X_7, X_8, X_{12}, X_{13}, X_{14}, X_{15})$ were generated as dichotomised versions of the standard normal variables. Covariates include: *direct confounders* (X_1, X_2, X_3, X_4); *distal confounders* (X_5, X_6); an *instrument* (X_7); *outcome*

only predictors (X_8, X_9, X_10) with X_8 and X_9 distally related to the treatment; and *distractors* $(X_{11}, ..., X_{15})$.

Following Setoguchi et al. (2008), logistic regression was used to model the treatment variable, W_i as a function of X_i. Seven scenarios that differed in the nature of the true propensity score model were considered (Lee et al., 2010; Setoguchi et al., 2008). The scenarios were: (A) linearity and additivity; (B) mild nonlinearity; (C) moderate nonlinearity; (D) mild non-additivity; (E) mild non-additivity and nonlinearity; (F) moderate non-additivity; (G) moderate non-additivity and nonlinearity (Gharibzadeh et al., 2018). Scenarios (A-G) varied with the levels of linearity and/or additivity of the modeled relationships between the treatment and the covariates. More details on the data generation process are presented in (Setoguchi et al., 2008). Random numbers between 0 and 1 were generated from the uniform distribution using the *R* software. In addition, 1 was allocated to W_i if the randomly generated number was less than $p(x) = Pr(W_i|X_i = x)$, and to 0 if the number generated was greater than $p(x) = Pr(W_i|X_i = x)$. Using logistic regression, a binary outcome variable Y_i was generated (for each scenario A-G) as a function of W_i and X_i, setting the effect of treatment W_i to be constant with the coefficient γ_i = *-0.4* as proposed by Setoguchi et al. (2008); Lee et al. (2010):

$$\begin{aligned} Pr[Y_i|W_i, X_i] = (1 + \exp\{-(&\alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3 \\ &+\alpha_4 X_4 + \alpha_5 X_8 + \alpha_6 X_9 + \alpha_7 X_{10} \\ &+\gamma_1 W_i)\})^{-1} \end{aligned} \tag{3.2}$$

Random numbers between 0 and 1 were generated from a uniform distribution using **R** software, setting $Y_i = 1$ if the randomly generated number was less than $Pr[Y_i|W_i, X_i]$ and 0 otherwise. The binary outcome variable for each scenario was used in training the deep learning models (DNN, PropensityNet, and autoencoder) and consequently predict propensity scores for each of these models. Logistic regression was used to generate a continuous outcome variable Y_i (for each scenario A-G), as a function of W_i and X_i and setting the effect of treatment W_i to be constant with coefficient γ_i = *-0.4* as proposed by Lee et al. (2010):

$$\begin{aligned} Y_i = &-\alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3 + \alpha_4 X_4 + \alpha_5 X_8 \\ &+\alpha_6 X_9 + \alpha_7 X_{10} + \gamma_1 W_i \end{aligned} \tag{3.3}$$

Weighted linear regressions of Y_i as a function of W_i and X_i were performed for Scenarios A-G using $1/p(x)$ and $1/(1\text{-}p(x))$ (Austin, 2011) as weights to estimate the treatment effect for each scenario. We used the same parameter values α_1 through α_7 as were used by (Lee et al., 2010; Setoguchi et al., 2008) in Equations 3.2 and 3.3.

3.3.2 Logistic Regression

Logistic regression (LR) is a common and useful statistical technique that is used to estimate propensity scores (Woo et al., 2008; Olmos and Govindasamy, 2015). Westreich et al. (2010) report that other techniques include discriminant analysis (Dehejia and Wahba, 2002), general boosted models (Setoguchi et al., 2008), classification trees (Linden et al., 2016), and neural networks (Macukow, 2016) just to mention a few. They point out that several propensity score analyses use LR to estimate the scores. In its basic form, LR is a statistical model that uses a logistic function to model a binary dependent variable. The general LR model is expressed as follows (Dobson and Barnett, 2018):

$$ln\left(\frac{Pr(W_i = 1|X_i = x)}{1 - Pr(W_i = 1|X_i = x)}\right) = \boldsymbol{X_i^T}\boldsymbol{\beta}, \tag{3.4}$$

where $\boldsymbol{X_i}$ is a vector of the continuous and dummy variables described in Section 3.3.1, and $\boldsymbol{\beta}$ is the vector of parameters. According to Athey and Imbens (2015), $Pr(W_i = 1|X_i = x)$ gives the propensity score (Equation 2.1) In this paper, LR will be used to estimate the propensity scores.

Logistic regression is mathematically constrained to produce probabilities between [0, 1] and thus makes it attractive for probability prediction (Muller and MacLehose, 2014). Logistic regression can be implemented easily in a wide variety of statistical software such as R, SPSS, STATA, and SAS. There are several shortcomings that can result from estimating propensity scores using logistic regression. Zhao et al. (2016) states that logistic regression is prone to misspecification errors that result in imprecise estimates of the propensity score. Missing data presents problems when estimating propensity scores using logistic regression, and these missing data have to be dealt with beforehand. Covariates with a large proportion of observations with missing data become unusable when implementing logistic regression. According to Westreich et al. (2010), the performance of logistic regression is poor compared to other meth-

ods that estimate propensity scores, such as tree ensembles or other machine learning algorithms.

3.3.3 Deep Neural Networks for Classification

3.3.3.1 The Cross-Entropy Loss Function

The cross-entropy loss function, *L(**Y**, f(**X**))*, is almost the only choice for classification tasks in practice. As discussed in Section 3.2.1, a classifier is represented by the mapping f:x$\rightarrow R^K$.

For multiclassification $L(Y, f(X))$ is defined as :

$$L(Y, f(X)) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\log(f_j;\theta), \tag{3.5}$$

where θ is the set of parameters of the classifier, y_{ij} corresponds to the j^{th} element of the *one-hot* encoded label of the $x_i, y_i \in \{0,1\}^K$ such that $\mathbb{1}^T y_i = \mathbb{1} \forall i$ and f_j denotes the j^{th} element of f. Note that $\sum_{j=1}^{N} f_j(x_{ij};\theta) = 1$, and $f_j(x_{ij};\theta) \geq 0$, $\forall j, i, \theta$, are the outputs obtained using *softmax* (Zhang and Sabuncu, 2018).

Equation 3.5 gives the average differences of the predicted outputs (probabilities) and the target probabilities. Sivaram et al. (2020) point out that using the above context, DNN, as an example, can be described as a classification model $\mathcal{N}$ with an architecture $\mathcal{A}$, and a vector of parameters Θ that express the output as a function of the input as follows:

$$\hat{Y} = \mathcal{N}(X; \mathcal{A}, \Theta) \tag{3.6}$$

$\mathcal{A}$ will represent the deep neural network design choices and Θ are the parameters that will be tuned during training. Design choices $\mathcal{A}$ usually include (i) how the layers are organised, (ii) the activation function which could be a rectified linear unit *ReLU* (Jagtap et al., 2020), hyperbolic *tahn* , logistic function, (iii) the number of layers of the deep neural network and (iv) a layer's nodes, n_l, $l = 1, 2, 3, ..., L$. The parameter Θ will include the weights W_l and the biases b_l of layers, $l = 1, 2, 3, ..., L$. A DNN model is created based on algorithms called artificial neural networks (ANN) that are structured as stacks of layers on top of each other. It can employ supervised and unsupervised learning (Sivaram et al., 2020). Farrell et al. (2018) report that ANNs are

not as familiar in fields such as econometrics compared to other methods such as logistic regression. DNN models use weights that are contained in hidden layers. These weights are adjusted during training as they take in and process inputs. The purpose of adjusting the weights is to find patterns that give better predictions. A DNN self-learns, and the researcher is not required to specify in advance any patterns to consider. Deep learning methods are based on a branch of machine learning called representation learning (feature learning) (Bengio et al., 2013). These methods perform automatic feature selection compared to machine leaning algorithms that require feature selection by the researcher before they are used.

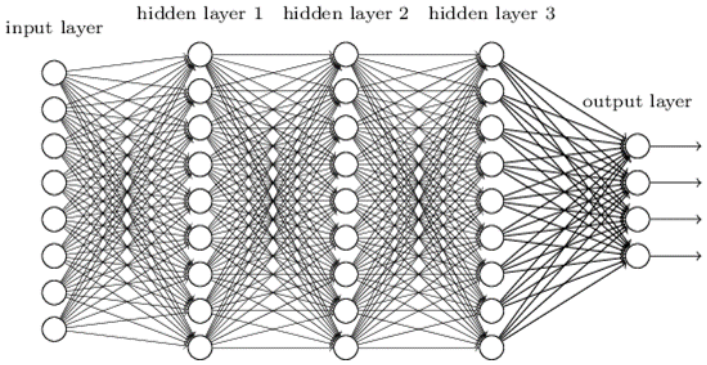


Figure 3.1: Fully connected DNN for classification.

The DNN learning process such as that in Figure 3.1 involves two important steps: the first step is the forward propagation phase of the training data, which takes in the raw data from the input layer. Hidden layers' neurons are then used to process the data which is passed on to the output layers to generate the output data. Hu et al. (2020) provides a detailed procedure of how the first step works. The second step involves the back-propagation of the error signal (Hu et al., 2020). The actual output and the expected (ideal) output are conveyed from the output layer to the input layer of the network. A cost function such as the cross entropy function described in Equation 3.5 is deployed. Thereafter, a backward propagation of errors occurs through the neural network.

For multiclass models, the DNN uses a *softmax layer* (Sivaram et al., 2020) as its last layer to retain the probabilities of each class. The *softmax layer* that is used for classification into K classes is defined as follows (Sivaram et al., 2020; Friedman et al., 2001):

$$f(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad \forall j \in \{1, ..., K\} \tag{3.7}$$

.

The softmax function, $f(x_i)$, produces an output between 0 and 1, with the sum of the probabilities equal to 1. In this paper, we are concerned with a binary classification problem where subjects are assigned to either the treatment group or the control group in estimating propensity scores. Thus, the output value of the class is 0 or 1. A *sigmoid* activation function in the output layer is used to predict class values of 0 or 1. The DNN is optimized using the binary cross-entropy loss function and the efficient **Adam** version of gradient descent (Kingma and Adam, 2014a).

Each hidden layer's output is passed on through to the next stage by nonlinear functions (Ramachandra, 2018). These nonlinear functions are referred to as *activation functions*. Examples of activation functions include the logistic function, hyperbolic tangent *tahn*, and rectified linear unit $ReLU$. Their main purpose is to convert an input signal from an input node in a DNN to an output signal. That output signal is then used as an input in the next layer of the stack. A neural network without an activation function would not be able to learn and model other complicated kinds of data such as images, videos, audio, speech, or even numeric data. The neural network without an activation function would simply be a linear regression model, which has limited power and poor performance for nonlinear relationships. In this work, we use the $ReLU$ activation function as it substantially reduces the computational cost of training and guarantees faster computation and convergence (Nwankpa et al., 2018). *ReLu* offers better performance and generalization in deep learning compared to the *sigmoid* and *tahn* activation functions. For a detailed discussion and comparison of the different activation functions, see Nwankpa et al. (2018).

Classification tasks typically involves extracting features, and then performing *classification*. The relevant features of the data are extracted by the hidden layers, and classification is performed by the last layer. Karsoliya (2012) points out that one output neuron for each group to which an input item is assigned is preferable for classification.

3.3.4 Autoencoders

An autoencoder is a deep learning algorithm that functions by compressing and encoding input data to a reduced representation, and then it reconstructs the reduced encoded representation to a representation that is as close as possible to the original input data. The autoencoder consists of an encoder that learns the underlying features of a process, and compresses them to a reduced dimension (Meng et al., 2017). In addition, the consists of a decoder which recreates the original data from these underlying features (Hoffmann et al., 2019). The code is responsible for converting the data into a *latent-space representation*, which is a reduced and compressed form of the input data. Autoencorders are primarily used for dimension reduction, image processing applications, and feature extraction (Wang et al., 2016). The architecture autoencoder used in this study is shown in Table 3.1. The reader is referred to Goodfellow et al. (2016) for a detailed mathematical description of an autoencoder.

Table 3.1: Architecture of the autoencoder

Layer (type)	**Output Shape**	**Param #**
input_2 (InputLayer)	(None, 15)	0
dense_4 (Dense)	(None, 10)	160
dense_5 (Dense)	(None, 15)	165
Total params: 325		
Trainable params: 325		
Non-trainable params: 0		

3.3.5 PropensityNet

PropensityNet (PN) is a deep neural network for propensity score matching that was proposed by Ramachandra (2018). It consists of five dense layers, i.e. fully connected layers. A full description of PN can be found in (Ramachandra, Vikas and Sun, Haoqiao, 2020). PropensityNet uses **Adadelta** (Zeiler, 2012) as an optimiser and the binary cross-entropy as an error metric, as it is solving a binary classification problem. A *sigmoid* function is used as the last layer to give probabilities between 0 and 1. These probabilities are a measure of the propensity score for each new/test unit. *Keras* with *Tensorflow* backend in **R** (Chollet et al., 2017) was used to build PropensityNet.

3.3.6 Experiments

We perform experiments to fit deep learning methods and evaluate their performance in classification tasks and estimation of propensity scores. We provide some comparisons of logistic regression, deep neural network (DNN) (Chollet et al., 2017), PropensityNet (Ramachandra, 2018), and autoencoder (Goodfellow et al., 2016). Our focus in this paper is about analysing the performance of the aforementioned deep learning models in estimating propensity scores and not explaining the technical details of the methods. To estimate propensity scores using DNN, we used the covariates $\boldsymbol{X_{ij}}$ and the outcome variable $\boldsymbol{Y_{ij}}$ for all the units as input and W_i as the output.

The structure of the DNN used in this study is shown in Figure 3.1. The DNN is a variation of the PropensityNet. The DNN has three hidden layers. The activation functions are $ReLU$ for both PropensityNet and DNN. Ramachandra (2018) used **Adadelta** (Zeiler, 2012) as an optimiser for PropensityNet. Our DNN will use the **Adam** (Kingma and Adam, 2014a) optimiser. Since we are performing binary classification, the binary cross-entropy is employed as the loss function. The dropout technique is used in both the DNN and PropensityNet in order to prevent overfitting (Srivastava et al., 2014a). The last layer which gives the output is a *sigmoid* function which is used for estimating probabilities for PropensityNet and DNN. The output is made up of probabilities, $p(x) = Pr(W_i|X_i = x) \in [\ 0,\ 1]$ for each unit from the test data set and these probabilities are the measure(s) of the *propensity scores*.

We modify the autoencoder and turn it into a classifier by adding a sigmoid output layer for binary classification as shown in Table 3.2.

Table 3.2: Modified autoencoder for binary classification

Layer (type)	**Output Shape**	**Param #**
input_2 (InputLayer)	(None, 15)	0
dense_4 (Dense)	(None, 10)	160
dense_5 (Dense)	(None, 1)	11
Total params: 171		
Trainable params: 171		
Non-trainable params: 0		

In essence, any neural network can be turned into a classifier by adding a sig-

moid or softmax output layer (Nwankpa et al., 2018). In order to build an autoencorder for estimating propensity scores, we use the following architecture: The input and output layers will have N neurons for a given data set of size N. The N-dimensional input data is reduced to say an M-dimensional middle layer, and thus the middle layers will have M neurons. The middle layer with M dimensions also contains hidden layers and the *code*. We train the autoencoder in two stages: *Stage 1* is the encoding phase where we train the encoder in the normal way to extract the encoded data set (i.e., the reduced features). *Stage 2* then retrains the autoencoder with the encoded (reduced) features extracted from stage 1. *Stage 2* differs from stage 1 in that the decoding phase is now replaced with a *sigmoid* layer. When training the autoencoder in stage 2, we do not change the encoder weights from stage 1 in order to guarantee tuning the output layer only. The classifier created in stage 2 consequently uses a small set of features. The $ReLU$ activation function is used during both stages of training. **Adam** (Kingma and Ba, 2014) is used as the optimiser and the binary cross-entropy as the loss function.

Hidden layers and the number of neurons of the hidden layers are crucial in determining the performance of backpropagation neural networks (Karsoliya, 2012). Wang and Li (2018) states that the number of neurons in a layer is defined as the width of a layer. A wider layer has more neurons. As far as the number of hidden layers are concerned we will use three hidden layers for DNN and check if the classification accuracy improves significantly from that of PropensityNet with more hidden layers but also taking care to avoid overfitting. There is no fixed rule or "best" rule for deciding the number of hidden layers and the number of neurons to use and thus, the "rule of thumb" is the most common technique. Karsoliya (2012) indicates that the use of one or two hidden layers is sufficient to solve any nonlinear complex problem and a third layer can be added if accuracy is the main and most needed criteria for designing the network.

KERAS Chollet et al. (2017) with *TensorFlow* back end in **R** is used to build the deep learning models described above. Using R R Core Team (2017), we build *logistic regression* models using W_i as the dependent variable and X_i as the regressors to estimate propensity scores. For the deep learning models we train them with (X_i, Y_i) as the inputs for each unit. We have used a simulation-based research for performance evaluation of the models because

the true treatment effects are usually unknown in the real world especially when working with observational data (Lee et al., 2010).

3.3.7 Evaluation Methodology

The performance of the models used in this paper were evaluated using the seven scenarios A−G, that represent different levels of linearity and additivity (including quadratic and interaction terms) in the true propensity score models. 1000 data sets for each of the sample sizes equal to *N= 500, N= 1000 and N = 2000* were used in each of the seven scenarios (A−G). To assess the performance of the deep learning models and the logistic regression model the following metrics were used:

(i) *Absolute Bias*: The absolute bias is used to estimate how the average treatment effect of 1000 simulations for the different sample sizes, and for each scenario agrees with the true value -0.4,

(ii) *Standard error (s.e)*: This is calculated as the average standard error of the treatment effects resulting from the 1000 simulations for each scenario and different sample size. The smaller the average standard error, the less the spread and the more likely that an estimated treatment effect sample mean is close to the true value *-0.4* of the treatment effect,

(iii) *Average Standardised Absolute Mean Difference* (ASAMD): We used Cobalt (Greifer, 2017) to calculate the average standardised absolute mean difference between the treatment and control groups after incorporating propensity score weights. The ASAMD is the average of the absolute values of the standardised difference in means across all covariates for different scenarios and sample sizes. The average value of *1000*-simulations is referred to as the mean ASAMD (Lee et al., 2010). Lower values of ASAMD suggest that the treatment and control groups are comparable for a given set of covariates,

(iv) *Accuracy (Acc)*: Classificatioon acccuracy is used as a metric for evaluating the classification performance of our models. The classification accuracy simply gives the proportion of predictions that our model got right. For example the higher the classification, the better the model is at classifying 0s as 0s and 1s as 1s. We note that the classification accuracy may not be a good performance metric when one is working with a rare outcome Y_i, where there is a significant disparity between the number of

0s and 1s. Our outcome variable is a *rare* binary outcome Y_i with $p(Y_i) \approx 0.02$ of the minority class and it is highly imbalanced. As a result, other performance metrics such as Cohen's Kappa, AUC-ROC, and No Information Rate (NIR) are also used for evaluating the class-imbalanced data considered in this study instead of the accuracy.

(v) *Cohen's Kappa (κ)*: is calculated using the formula, $\kappa = \frac{Pr(a)-Pr(e)}{1-Pr(e)}$, where *Pr(a)* represents the observed actual agreement, and *Pr(e)* represents the chance agreement. In binary classification, accuracy is a common performance metric. However, it can be misleading in the case of imbalanced data (Akosa, 2017). For an imbalanced data set, the classification task may be influenced by the majority class. Therefore, instead of using the accuracy metric Cohen's Kappa will be used as one of the metrics to evaluate the *agreement* between the actual classes and the classes predicted by DNN, PropensityNet, autoencoder and logistic regression models. Cohen's Kappa takes values between 0 and 1 with a value of 1 implying perfect agreement and values less than 1 imply less perfect agreement between the actual and predicted classes (Landis and Koch, 1977a).

(vi) *No Information Rate (NIR)* and P-Value [Acc > NIR]: The "no-information rate (NIR)" is the largest proportion of observed classes. This means that given a "rare" binary outcome Y_i with $p(Y_i) \approx 0.02$, the majority class has a probability approximately equal to 98%. A model whose classification accuracy is say 90% and the NIR is 98% tells us that if we just pick the majority class, we will be correct 98% of the time. A hypothesis test is also computed to evaluate whether the overall accuracy rate is greater than the rate of the majority class.

(vii) *AUC-ROC* is used as a measure of performance to classify the binary class variable Y_i. The AUC-ROC is a probability curve, and it represents the degree or measure of separability. This means that it gives us a measure of how a model is capable of distinguishing between classes. For example the higher the AUC, the better the model is at predicting 0s and 1s correctly. The AUC-ROC is a function of *sensitivity* and *specificity*.

(viii) *95% CI coverage*: This is the percentage of the *1000* data sets in which the estimated 95 percent confidence interval included the true treatment effect.

3.4 Results and Discussion

The results for Monte Carlo simulations from samples of size $N = 1000$ are presented first and thereafter, we present the results from smaller samples of size $N = 500$ and larger samples of size $N = 2000$. Several critical discoveries were made about the performance of the different methods used in this paper in estimating propensity scores and average treatment effects. We note that logistic regression can perform better with proper selection of interactions than the simple main effects-only model considered in this study.

This study was conducted to assess the performance of logistic regression, deep neural network (DNN with three hidden layers), PropensityNet and the autoencoder in estimating propensity scores, and consequently average treatment effects. The aim was to determine whether deep learning methods can be successfully used to estimate propensity scores. Also, the study proposed the DNN algorithm that was designed to improve the performance of PropensityNet, and show that it performs better than PropensityNet and logistic regression in estimating propensity scores and reducing *bias* when estimating average treatment effects. The study extended the work by Setoguchi et al. (2008); Lee et al. (2010); Ramachandra (2018); Cannas and Arpino (2019) by incorporating nonparametric statistical tests such as the Cohen's Kappa, and the No Information Rate. Hypotheses tests were also performed to evaluate whether the overall accuracy rate was greater than the rate of the majority class (NIR) for each model. The performance of these propensity score methods was tested under different sample size conditions.

3.4.1 Simulations of $N = 1000$ sample sizes

3.4.2 Simulations of $N = 500$

The average performance metrics values across all scenarios for the different models for samples of size $N = 500$ were largely similar to those for samples of size $N = 1000$.

3.4.3 Simulations of $N = 2000$

The values of the average performance metrics for all scenarios for the different models for samples of size $N = 2000$ were largely similar to those for samples

Table 3.3: Performance metrics for logistic regression (LR); deep neural network (DNN); PropensityNet (PN), and autoencoder (AE) for sample size *N = 1000.*

		Scenario							
Metrics	**Model**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**Average**
ASAMD	LR	0.051	0.061	0.088	0.069	0.089	0.082	0.088	**0.075**
	DNN	0.402	0.439	0.408	0.429	0.445	0.441	0.438	0.429
	PN	0.251	0.371	0.273	0.356	0.237	0.266	0.169	0.275
	AE	0.419	0.741	0.753	0.622	0.622	0.670	0.854	0.669
Absolute Bias	LR	0.005	0.043	0.043	0.043	0.043	0.043	0.043	0.038
	DNN	0.025	0.007	0.005	0.008	0.009	0.007	0.007	**0.009**
	PN	0.010	0.006	0.016	0.008	0.013	0.013	0.018	0.012
	AE	0.047	0.015	0.033	0.011	0.011	0.039	0.051	0.030
Standard Error	LR	0.027	0.027	0.027	0.027	0.027	0.027	0.027	0.027
	DNN	0.045	0.049	0.049	0.049	0.049	0.048	0.048	0.048
	PN	0.019	0.019	0.020	0.019	0.020	0.019	0.019	**0.019**
	AE	0.135	0.357	0.379	0.269	0.269	0.312	0.522	0.320
Accuracy (%)	LR	64.40	28.53	28.53	28.53	28.53	28.53	28.53	33.66
	DNN	99.09	99.82	99.81	99.82	99.80	99.81	99.81	**99.71**
	PN	73.87	97.20	55.75	98.53	59.91	86.00	58.80	75.72
	AE	96.30	98.10	98.10	98.10	98.10	98.10	98.10	97.84
AUC-ROC (%)	LR	27.91	27.91	27.91	27.91	27.91	27.91	27.91	27.91
	DNN	99.76	99.59	96.64	97.16	96.93	97.23	96.20	**97.64**
	PN	50.18	65.74	53.55	58.81	46.34	56.19	51.81	54.66
	AE	55.28	55.10	54.48	54.81	55.04	55.18	55.32	55.03
Cohen' Kappa	LR	0.046	0.018	0.018	0.018	0.018	0.018	0.018	0.022
	DNN	0.819	0.925	0.919	0.923	0.918	0.918	0.918	**0.906**
	PN	0.001	0.003	0.003	0.000	0.001	0.011	0.003	0.003
	AE	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
95% Coverage	LR	90.60	90.20	90.80	88.60	88.00	76.00	51.10	82.19
	DNN	96.10	95.10	96.40	94.30	94.50	95.30	94.80	95.21
	PN	100.00	100.00	100.00	100.00	100.00	100.00	100.00	**100.00**
	AE	100.00	100.00	100.00	100.00	100.00	100.00	100.00	**100.00**
p-Value [Acc > NIR]	LR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	DNN	0.010	0.009	0.009	0.009	0.009	0.011	0.009	**0.009**
	PN	0.945	0.919	0.946	0.960	0.952	0.932	0.959	0.945
	AE	0.544	0.561	0.561	0.561	0.561	0.561	0.561	0.558

of size $N = 1000$.

ASAMD is an excellent measure to assess covariate balance because it can effectively predict the bias of the average effect of treatment (Stuart et al., 2013). Cannas and Arpino (2019) suggested as a "rule of thumb" a more stringent criterion for obtaining ASAMD values lower than 0.1 in order to achieve covariate balance. Using Table 3.3, the average ASAMD for DNN, autoencoder, and PropensityNet are not acceptable as they are all greater than 0.1 in all scenarios. This means that the deep learning models did not achieve covariate balance. On the other hand, logistic regression achieved covariate

Table 3.4: Performance metrics for logistic regression (LR); deep neural network (DNN); PropensityNet (PN), and autoencoder (AE) for sample size *N = 500.*

		Scenario							
Metrics	**Model**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**Average**
ASAMD	LR	0.184	0.069	0.073	0.073	0.073	0.073	0.073	**0.088**
	DNN	0.428	0.384	0.391	0.393	0.389	0.393	0.392	0.396
	PN	0.228	0.311	0.362	0.211	0.313	0.373	0.172	0.282
	AE	0.515	0.468	0.637	0.561	0.561	0.501	0.386	0.518
Absolute Bias	LR	0.014	0.042	0.042	0.042	0.042	0.042	0.042	0.038
	DNN	0.014	0.026	0.026	0.025	0.026	0.027	0.027	**0.024**
	PN	0.042	0.026	0.024	0.053	0.040	0.038	0.036	0.037
	AE	0.090	0.041	0.075	0.078	0.078	0.075	0.042	0.069
Standard Error	LR	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039
	DNN	0.049	0.048	0.043	0.044	0.043	0.044	0.044	0.045
	PN	0.028	0.027	0.027	0.027	0.027	0.027	0.027	**0.027**
	AE	0.201	0.155	0.287	0.237	0.237	0.206	0.147	0.210
Accuracy(%)	LR	52.80	33.87	33.87	33.87	33.87	33.87	33.87	36.57
	DNN	99.30	99.61	99.30	99.02	99.30	99.09	99.09	**99.24**
	PN	71.97	90.93	98.10	63.20	90.13	96.53	50.40	80.18
	AE	97.20	98.40	98.40	98.40	98.40	98.40	98.40	98.23
AUC-ROC(%)	LR	37.89	33.60	33.60	33.60	33.60	33.60	33.60	34.22
	DNN	99.76	99.59	96.64	97.16	96.93	97.23	96.20	**97.64**
	PN	55.64	53.03	56.95	46.43	52.25	62.73	49.50	53.79
	AE	66.25	60.28	59.39	60.10	60.40	60.54	60.54	61.07
Cohen' Kappa	LR	0.025	0.016	0.016	0.016	0.016	0.016	0.016	0.017
	DNN	0.849	0.861	0.732	0.749	0.737	0.745	0.734	**0.772**
	PN	0.014	0.026	0.000	0.008	0.031	0.018	0.005	0.014
	AE	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
95% Coverage	LR	88.80	89.50	90.60	84.90	77.80	63.40	57.60	78.94
	DNN	97.80	99.80	99.40	99.60	99.30	99.80	99.40	99.30
	PN	100.00	100.00	100.00	100.00	100.00	100.00	100.00	**100.00**
	AE	100.00	100.00	100.00	100.00	100.00	100.00	100.00	**100.00**
p-Value [Acc > NIR]	LR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	DNN	0.040	0.041	0.049	0.037	0.045	0.045	0.043	**0.043**
	PN	0.944	0.951	0.953	0.958	0.969	0.921	0.967	0.952
	AE	0.579	0.598	0.599	0.600	0.598	0.599	0.600	0.596

balance, as it produced mean ASAMD values that ranged from 0.051 - 0.089, which were all less than 0.1.

Propensity score weighting is an important preprocessing technique used in order to achieve covariate balance. Achieving covariate balance justifies *ignorability* on the observed covariates, which in turn allows for a valid causal inference to be made. Although logistic regression achieved covariate balance for all the different sample sizes (Tables 3.3-3.5), and across all scenarios A-G, its absolute biasses were consistently higher than those of the DNN and

Table 3.5: Performance metrics for logistic regression (LR); deep neural network (DNN); PropensityNet (PN), and autoencoder (AE) for sample size *N = 2000.*

		Scenario							
Metrics	**Model**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**Average**
ASAMD	LR	0.163	0.038	0.041	0.041	0.041	0.041	0.041	**0.058**
	DNN	0.361	0.383	0.389	0.376	0.408	0.400	0.403	0.389
	PN	0.299	0.299	0.302	0.304	0.307	0.303	0.301	0.302
	AE	0.386	0.450	0.487	0.534	0.534	0.451	0.506	0.478
Absolute Bias	LR	0.009	0.042	0.042	0.042	0.042	0.042	0.042	0.038
	DNN	0.014	0.016	0.017	0.016	0.017	0.018	0.017	0.017
	PN	0.013	0.011	0.011	0.011	0.012	0.011	0.011	**0.011**
	AE	0.169	0.065	0.079	0.112	0.112	0.120	0.078	0.105
Standard Error	LR	0.020	0.019	0.019	0.019	0.019	0.019	0.019	0.019
	DNN	0.029	0.034	0.034	0.034	0.034	0.034	0.034	0.033
	PN	0.015	0.013	0.013	0.013	0.013	0.013	0.013	**0.014**
	AE	0.069	0.159	0.206	0.251	0.251	0.155	0.210	0.186
Accuracy (%)	LR	73.20	39.27	39.27	39.27	39.27	39.27	39.27	44.11
	DNN	99.35	99.67	99.54	99.68	99.64	99.50	99.46	**99.55**
	PN	65.87	65.82	64.70	64.70	65.48	66.43	65.21	65.46
	AE	96.35	98.10	98.10	98.10	98.10	98.10	98.10	97.85
AUC-ROC	LR	41.22	24.65	24.65	24.65	24.65	24.65	24.65	27.01
	DNN	99.45	99.50	99.44	99.72	99.63	99.51	99.44	**99.53**
	PN	56.34	56.64	56.68	56.54	56.72	57.00	56.77	56.67
	AE	50.51	39.45	40.68	40.69	40.68	40.68	40.67	41.91
Cohen' Kappa	LR	0.057	0.036	0.036	0.036	0.036	0.036	0.036	0.039
	DNN	0.175	0.479	0.475	0.473	0.469	0.481	0.463	**0.431**
	PN	0.000	0.000	0.001	0.001	0.001	0.001	0.001	0.001
	AE	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
95% Coverage	LR	88.90	89.50	90.60	87.20	72.30	64.70	33.50	75.24
	DNN	96.10	95.10	96.40	94.30	94.50	95.30	94.80	95.21
	PN	96.10	96.10	96.10	96.70	96.40	96.40	95.30	96.16
	AE	100.00	100.00	100.00	100.00	100.00	100.00	100.00	**100.00**
p-Value [Acc > NIR]	LR	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	DNN	0.002	0.001	0.002	0.002	0.008	0.008	0.006	**0.004**
	PN	0.910	0.867	0.878	0.880	0.900	0.902	0.884	0.889
	AE	0.561	0.550	0.550	0.548	0.548	0.549	0.548	0.551

PropensityNet. This important finding may suggest that achieving covariate balance may not be enough to lower bias, or it may be that the ASAMD does not adequately measure covariate balance. This finding is supported by Lee et al. (2010); Linden and Yarnold (2017); Cannas and Arpino (2019).

Table 3.3 shows that the absolute bias for the logistic regression for scenario A was generally acceptable and low (0.005, 95% CI =90.6%). Scenario A represents additivity and linearity (main effects only). However, as the scenarios became more nonadditive and nonlinear (scenarios B-G), the performance of

logistic regression was poor. For example, with moderate nonadditivity and nonlinearity (scenario G), LR produced a average absolute bias of 0.043 and a 95% CI coverage of 51.1% (scenario G). These results show that the LR propensity score model overestimated the true causal effect of the treatment as the data became more non-additive and nonlinear. Table 3.3 shows that the DNN and PropensityNet had low absolute biases averaging (across all scenarios) 0.009, and 0.012, respectively and high 95% CI coverage. The DNN displayed the lowest bias as nonadditivity, and nonlinearity in the data was increased. Thus, the DNN performed better at reducing absolute bias compared to PropensityNet, logistic regression, and the autoencoder.

As supported by Macukow (2016), three hidden layers are sufficient for the proposed DNN. Increasing the number of hidden layers does not inevitably improve performance in terms of reducing absolute bias. Also, the DNN achieved superior and stable performance using the Adam stochastic optimiser as opposed to PropensityNet which was unstable when it was trained using the Adadelta stochastic optimiser. This finding shows that standard stochastic optimisation methods may exhibit instability during model training. Therefore, before implementing deep learning models, it is important to choose a stochastic optimiser that offers better stability and performance during model training (Asi and Duchi, 2019).

The standard errors for the logistic regression, DNN, PropensityNet were comparable on average with standard errors averaging 0.027, 0.048, and 0.020 across all scenarios respectively. This shows that these models did not produce standard error estimates that were substantially different from each other. The low standard errors for logistic regression came at the expense of relatively poor 95 percent CI coverage which averaged 82.2% across all scenarios compared to the DNN and PropensityNet which averaged 95.2% and 100.0% respectively. Despite achieving 100.0% 95 percent CI coverage, the autoencoder tended to produce the largest standard errors on average (0.320) compared to the other methods as shown in Table 3.3.

Table 3.3 shows that DNN gave superior classification accuracy (99.71%) on average across all scenarios compared to logistic regression (33.66%), PropensityNet (75.72%) and the autoencoder (97.84%). This means that the DNN and the autoencoder were able to accurately classify the rare binary outcome variable Y_i compared to the other three models. Logistic regression gave the

poorest classification accuracy followed by PropensityNet, which was moderate. The accuracy performance metric can be a useful measure if we have the same amount of samples per class, but if we have an imbalanced set of samples it is not useful at all. Therefore, we considered other measures such as the AUC-ROC, which gives the performance of a model, while addressing the issue of class imbalance to evaluate the performance of our models.

Based on the AUC-ROC, the DNN performed better than the other models across all scenarios. The average AUC-ROC across the different scenarios was 27.9% for logistic regression, 97.6% for DNN, 54.6% for PropensityNet and 55.03% for the autoencoder, as shown in Table 3.3. The AUC-ROC results revealed that the DNN returned a better classification compared to other algorithms. The AUC-ROC for logistic regression was very poor, and those for PropensityNet and the autoencoder were unacceptable. The AUC-ROC gives a better measure of accuracy because it is a function of sensitivity and specificity. The AUC-ROC curve is insensitive to differences in class proportions.

In addition to evaluating our models using classification accuracy and AUC-ROC we also considered Cohen's Kappa. The average Cohen's Kappa values for logistic regression, DNN, PropensityNet, and the autoencoder were 0.022, 0.906, 0.003, and 0.000, respectively, for all scenarios (Table 3.3). The Cohen's Kappa statistic for the DNN indicates that there was substantial agreement between the actual classes and the predicted classes (Landis and Koch, 1977a). Thus, DNN can handle very well unbalanced class problems. This means that the DNN was doing a good job of predicting propensity scores and also classifying 0s and 1s. Cohen's Kappa statistic values for logistic regression, PropensityNet, and the autoencoder did not offer any agreement between the actual classes and the predicted classes. These models are not capable of distinguishing between classes, i.e., they are not good at predicting 0s and 1s correctly.

Table 3.3 shows that on average the p-value [Acc $>$ NIR] for DNN (0.009 $<$ 0.05) is statistically significant at a 5% level of significance. This means that the average classification accuracy for DNN (99.71%) across all scenarios is significantly greater than the NIR (98.53%). Thus, DNN is a useful model for predicting propensity scores that are used to calculate average treatment effects. The p-value [Acc $>$ NIR] for logistic regression, PropensityNet, and the autoencoder were not significant as they were all greater than 0.05. With

the no-information rate in mind, we now see that the accuracy of the logistic regression model was very poor or bad. We note that a good model is one where the no information rate is significantly less than accuracy. It is important to check whether the accuracy is significantly greater than the no information rate to determine whether the model is actually doing anything useful for the particular outcome it claims to predict.

3.4.4 Case Study

Due to the fundamental problem of causal inference, finding observational data sets that have the ground-truth ATE (or ITE) is a challenge in practice. The data set used as an example in this paper has an estimate of the population ATE. The case study example comes from the Atlantic Causal Inference Conference (ACIC) Data Challenge 2019 Yao et al. (2020). The challenge was to estimate the average treatment effect (ATE) using a quasi-real world data set (Yao et al., 2020). Some of the covariates used in the ACIC 2019 data challenge were derived from simulations as well as from publicly available data sets. The link to the **R** code for the data generation processes is available in Yao et al. (2020). Various challenges of estimation of the average treatment effects were incorporated into the processes to generate the binary treatment assignment as well as the binary or continuous outcomes. These challenges include violations of the positivity assumption, different proportions of true confounders among the observed covariates, treatment effect heterogeneity, and nonlinearity of the response surface. The data sets consist of 3200 low-dimensional data sets, and 3200 high-dimensional data sets. We randomly selected a low-dimensional data set of size $N = 5735$ with 26 covariates and the true $\boldsymbol{ATE = 2.5274}$. The data set chosen has six binary variables, one categorical variable with four levels, 14 continuous variables, and five integer variables. The data set is unbalanced with a less frequent binary outcome Y_i with $P(Y_i) \approx 0.12$ and a binary treatment variable A_i with $P(A_i) \approx 0.46$. The results (Table 3.6) show that the average treatment effect estimates obtained from the DNN had the least biased causal effect estimates compared to the other models.

Also, the ASAMD results for the DNN, PropensityNet, and the autoencoder were greater than 0.1, indicating that these models did not achieve covariate balance. The DNN gave a much better classification accuracy (88.84%). The AUC-ROC for the DNN was between 80%-90% and is considered excellent

Table 3.6: Case study results for logistic regression (LR); deep neural network (DNN); PropensityNet (PN) and autoencoder (AE).

	Model			
Metrics	**LR**	**DNN**	**PN**	**AE**
ASAMD	**0.080**	0.532	0.445	0.451
Absolute Bias	0.203	**0.097**	0.253	0.265
Standard Error	**0.338**	0.677	0.415	0.597
Accuracy (%)	40.17	**88.44**	74.60	87.90
AUC-ROC (%)	35.46	**86.92**	54.84	53.50
Cohen' Kappa	0.128	**0.419**	0.013	0.000
95% coverage	0.000	**100.0**	100.0	100.0
p-Value [Acc > NIR]	1.000	**0.010**	1.000	0.543

(Mandrekar, 2010). The AUC-ROC for PropensityNet, and the autoencoder were between 50%-60% suggesting the models were poor at classifying 0s and 1s. Table 3.6 shows that Cohen's Kappa for DNN is between 0.40 and 0.60, indicating moderate agreement between the actual and predicted classes. There was no agreement between the actual and predicted classes for PropensityNet and the autoencoder as they had very low Cohen's Kappa values. Furthermore, DNN accuracy (88.84%) was significantly higher than the No Information Rate (NIR) (87.26%) as the p-value [Acc > NIR] was significant ($0.010 < 0.05$).

The results of the case study show that the DNN is a useful model for estimating propensity scores. The DNN produced the best classification accuracy, Cohen's Kappa and AUC- ROC and significantly reduced bias in cases where the data may be complex compared to logistic regression in practice. The ACIC 2019 data set used in this paper was complex because challenges such as nonlinearity of the response surface, treatment effect heterogeneity, varying proportion of true confounders among the observed covariates, and near violations of the positivity assumption were incorporated into the data generation process of the dichotomous treatment assignment, and the binary or continuous outcomes.

3.5 Conclusion

In conclusion, our simulation results show that using deep neural networks models (DNN) with three hidden layers offers a number of advantages over

logistic regression, PropensityNet and the autoencoder in propensity score estimation. DNN can significantly improve 95% CI coverage and also significantly reduced bias over a range of sample sizes from $N = 500$ to $N = 2000$, and scenarios A−G, compared to logistic regression. The DNN also has excellent predictive performance in modelling rare binary outcomes. The DNN proved to be a useful model in predicting propensity scores, as it produced excellent values for the accuracy, AUC-ROC, Cohen's Kappa and significant p-values[Acc > NIR]. The PropensityNet with **five** fully connected layers performed poorly compared to the DNN with three hidden layers. Furthermore, the results show that a deep neural network with three hidden layers can be used in place of logistic regression to estimate propensity scores, since logistic regression is prone to misspecification errors that can result in imprecise estimates of propensity scores. The literature on using deep neural networks to estimate propensity scores using class-membership probabilities is still limited. This study has shown that with the correct configuration, deep learning methods can be employed to reduce or eliminate the reliance on logistic regression assumptions regarding variable selection, the functional form, distribution of variables and specification of interactions. Also, DNN performed better than logistic regression in the estimation of the propensity score and the reduction of bias when estimating the average treatment effects. Thus, deep learning models, such as the DNN can be used in situations where the objective is to reduce absolute bias in the causal effects.

We strongly recommend that further research should focus on applying the deep neural network (DNN) method in more real-life situations in estimating propensity scores. More research should be carried out using DNN to compare the performance of propensity score matching and propensity score weighting when there are multiple treatment groups. The results of such research will further inform us regarding the advantages and disadvantages of deep learning methods in situations where we have different propensity score analysis techniques (matching vs weighting) and more than two treatment groups. To the best of our knowledge, the application of propensity scores derived from deep learning methods to estimate average treatment effects has not been explored for studies with multiple treatment groups.

CHAPTER 4

A Machine Learning Evaluation of the Effects of South Africa's COVID-19 Lockdown Measures on Population Mobility

---

ALBERT WHATA [1,*] and CHARLES CHIMEDZA [2]

[1] School of Natural and Applied Sciences, Sol Plaatje University

[2] School of Statistics and Actuarial Science, University of the Witwatersrand

Published in: *Machine Learning and Knowledge Extraction*, (2021) **3(2)**: 481-506.

Statement of Contributions of Joint Authorship

Albert Whata **(Candidate)**
Conducted the research, writing and compilation of manuscript);

Charles Chimedza **(Supervisor)**

Supervised, edited and coauthor of the manuscript.

This chapter is an exact copy of the journal paper (Whata and Chimedza, 2021b) referred to above, and available at `https://www.mdpi.com/2504-4990/3/2/25`**.**

ABSTRACT

Following the declaration by the World Health Organization (WHO) on 11 March 2020 that the global COVID-19 outbreak had become a pandemic, South Africa implemented a full lockdown from 27 March 2020 for 21 days. Full lockdown was implemented after the publication of the National Disaster Regulations (NDR) gazette on March 18, 2020. The regulations included lockdowns, public health measures, movement restrictions, social distancing measures, and social and economic measures. We developed a hybrid model that consists of a long-short term memory autoencoder (LSTMAE) and the kernel quantile estimator (KQE) algorithm to detect change-points. Thereafter, we used the Bayesian structural times series models (BSTSM) to estimate the causal effect of the lockdown measures. The LSTMAE and KQE, successfully detected the change-point that resulted from the full lockdown that was imposed on March 27, 2020. Additionally, we quantified the causal effect of the full lockdown measure on population mobility in residential places, workplaces, transit stations, parks, grocery and pharmacy, and retail and recreation. The mobility of the population in grocery and pharmacy stores decreased significantly by 17137.04% (p-value = 0.001 $<$ 0.05). Population mobility at transit stations, retail and recreation, workplaces, parks, and residential places decreased significantly by 998.59% (p-value = 0.001 $<$ 0.05), 1277.36% (p-value = 0.001 $<$ 0.05), 2175.86% (p-value = 0.001 $<$ 0.05), 70.00% (p-value = 0.001 $<$ 0.05), and 22.73% (p-value = 0.001 $<$ 0.05), respectively. Therefore, the level 5 full lockdown imposed on 27 March 2020 had a causal effect on population mobility in these categories of places.

4.1 Introduction

On 11 March 2020, the World Health Organization (WHO) declared that the global COVID-19 outbreak had become a pandemic (World Health Organization, 2020). Consequently, the government of South Africa declared a national

state of disaster on 15 March 2020 (South Africa. Dept. of Co-operative Governance and Traditional Affairs., 2020). When the outbreak worsened, the government ordered all South Africans to a full lockdown. Full lockdown was effective for 21 days from March 26, 2020. The full lockdown was implemented after the publication of the National Disaster Regulations (NDR) Gazette on March 18, 2020 (Department of Co-operative Governance and Traditional Affairs, 2020). The regulations or measures contained in the gazette were applicable for the duration of the full lockdown. These drastic regulations or measures imposed on the public included lockdowns, public health measures, movement restrictions, social distancing, and social and economic measures. For example, the nationwide lockdown, which was initially set for 21 days ending April 16, 2020, required that everyone except those providing essential services stayed at home. People were only allowed to leave their homes for urgent food shopping and medical treatments. The lockdown measure was imposed to fundamentally disrupt the chain of transmission of the corona virus and to stop the spread of the virus, thereby saving South African lives. Although the lockdown was viewed as the best response from a public health perspective, the economic impact was devastating for ordinary South African households and businesses (Ajam, 2020).

In this paper, we seek to identify change-points (Dehning et al., 2020) using the Google COVID-19 Community Mobility Reports (Google LLC., 2020) and the South African government COVID-19 measures contained in the Government Measures data set provided by ACAPS (ACAPS., 2020). In addition, we study how these government interventions affect population mobility in areas including workplaces, residential, transit stations, parks, grocery and pharmacy, and retail and recreation. The aim is to quantify the causal effects of the South African government interventions on population movements in these areas. The literature on change-points informed part of our approach in evaluating the causal effects of the government measures. For example, Taylor (2000) indicates that for historical data, control charts have traditionally been used to detect single change points. The authors mention that single change-point methods have applied classical statistical thresholding algorithms based on the mean changes. In addition, other statistical methods estimate the probability of a change-point occurring by utilizing Bayesian priors that incorporate time-dependent information.

We developed a hybrid model that consists of a long-short term memory autoencoder (LSTMAE) and the kernel quantile estimator (KQE) algorithm to detect change-points. We used the LSTMAE algorithm because it has been shown to perform better in anomaly or intrusion detection (Li et al., 2019; Farahnakian and Heikkonen, 2018; Sovilj et al., 2020; Zhang et al., 2019a; Tan et al., 2019). There are some advantages to using LSTMs. For example, LSTMs can be used in sequences of varying lengths (Singh, 2017) without making any assumptions about the number of previous points that are needed to make predictions. LSTMs are structured to exploit temporal dependencies in sequential data and do not assume any functional form between outcome variables and regressors or explanatory variables (Poulos, 2017).

LSTMs are a variant of recurrent neural networks (RNNs). RNNs are a very powerful tool in deep learning (Tran et al., 2019). According to Murad and Pyun (2017), RNNs outperform conventional machine learning methods that include k-nearest neighbours (KNN) and support vector machines (SVM) because they contain "memory" that captures past information regarding what has been calculated and they can also learn long-range patterns. On the other hand, we use the KQE to determine a threshold that is used to detect anomalies in the reconstruction errors obtained from the LSTMAE. The KQE is desirable because we do not want to assume a parametric form for the distribution of the reconstruction errors (Sheather and Marron, 1990). This means that the KQE offers flexibility over parametric estimators, as we can choose from several classes of functions where we assume the reconstruction errors to belong. In addition, the KQE expresses the univariate distribution of reconstruction errors as a finite mixture and thus gives a smooth distribution from which to estimate quantiles (Siloko et al., 2019).

After detecting the change-point(s) we then create a Bayesian structural time series model (BSTSM), to predict a counterfactual and then measure the causal effect of the South African government interventions such as a lockdown (change-point) on population mobility. The BSTSM is implemented in the R package, CausalImpact (Brodersen et al., 2015) using the COVID-19 Community Mobility Reports by Google (Google LLC., 2020). A study of the causal effect of interventions will provide information on the effect of government measures and, we hope, will assist those making critical decisions to combat COVID-19 or any other possible future pandemic.

Change-point detection entails finding the location in a sequence of observations where the statistical properties change (Killick and Eckley, 2014). Detecting change-points is important in many different application areas. Several supervised and unsupervised techniques that can be used for change-point detection in time series data were surveyed by (Aminikhanghahi and Cook, 2017). Change-point detection has primarily been used to model and predict time series in several application areas such as climatology (Gallagher et al., 2013), bioinformatic applications (Muggeo et al., 2008), finance (Pepelyshev and Polunchenko, 2015), medical imaging (Nika et al., 2014), speech (Tahmasbi and Rezaei, 2008) and image analysis (Radke et al., 2005).

Change-point analysis can be employed to evaluate the effect of an intervention using synthetic control methods (SCM) for comparative case studies (Bouttell et al., 2018). According to Abadie et al. (2015), the use of SCM for comparative case studies involves comparing units that are subjected to an event or intervention of interest with one or more units that are not exposed. This means that comparative case studies are only possible when some units are exposed to an intervention whilst others are not exposed. Thus, change-points can be used to separate units that are exposed to an intervention and units that are not exposed. When investigating the effect of a policy or intervention, identifying change-points in each data set on interventions is very important. This is because the change-point analysis should verify that a change-point has indeed occurred at the time or point of intervention. Thus, at a change-point, we can estimate the average treatment effect of a change-point or intervention. The challenge in change-point analysis is to come up with an algorithm that automatically detects changes in the properties of sequences to allow us to make the appropriate decisions. This is because change-points in a sequence can be described as "rare events", like anomalies that make it harder for the classification problem to detect it, as the data set will be heavily imbalanced. The points are changes only at that temporal context and not as independent points. Therefore, this problem is difficult to solve using general classification algorithms. Because LSTMs have memory within their structure, they are better suited to capture patterns inside the sequences. The behavioural change in the sequence at any temporal context will also have patterns among them. Thus, LSTMs make a reasonable option to solve the change-point detection problem, as they have the capability to learn the patterns in sequences.

While associated means or variances can be obtained, we specifically focus on detecting the positions and number of the change-points. After detecting and verifying the existence of change-points, a model is fit that can predict the counterfactual using the pre-intervention time series and then compare the predicted (counterfactual) to the actual times that are recorded after the intervention. The BSTSM that can be implemented in CausalImpact R package (Brodersen et al., 2015) are then used to estimate the average treatment effect of an intervention (change-point).

4.2 Review of Literature

In this section, a review of the literature and related work is presented.

4.2.1 Causal Inference and the Counterfactual Approach

Causal inference has been studied in Statistics (Reiter, 2000; Rubin, 2003), Econometrics (Heckman, 2008) and Biostatistics (Egleston et al., 2007). These studies have focused mostly on a setup where there is a binary treatment, and the Rubin Causal Model (RCM) or the potential outcomes framework is often used (Holland, 1986). The basic element of causal inference is that each unit in a large population is characterised by the potential outcomes $Y_i(0)$ and $Y_i(1)$ (Athey et al., 2017). In addition, the difference, $Y_i(1) - Y_i(0)$, gives the unit-level, treatment effect. If we let $Z \in \{0, 1\}$ be a binary indicator for treatment, with $Z_i = 0$ if unit i received the control and $Z_i = 1$ if unit i received the active treatment, then $Y_i(1)$ is the outcome if unit i receives the active treatment and $Y_i(0)$ is the outcome if unit i receives the control. Note that $Y_i(1)$ and $Y_i(0)$ can never be observed at the same time on the same unit. Holland (1986) refers to this as the fundamental problem of causal inference. Because we cannot realise $Y_i(0)$ and $Y_i(1)$ at the same time on the same unit, Holland (1986) states that the average causal effect then becomes the typical measure of a causal effect. Calculation of the average causal effect involves exposing some units in the population to treatment 1, providing information about $E(Y_i(1))$ and some units to treatment 0, providing information about $E(Y_i(0))$. Since both outcomes are observable, the average treatment effect is estimated by: $\tau = E(Y_i(1)) - E(Y_i(0))$.

There are basically two important assumptions linked to causality that need to

be considered, and these are the endogeneity (Antonakis et al., 2014) and the Ignorable Treatment Assignment assumptions (Austin, 2011). Endogeneity occurs when the error term *(e)* is correlated with a regressor *(x)* (Antonakis et al., 2014). A variety of conditions can lead to violations of this assumption, but one important case occurs when key variables are excluded from the model. This means that there exist some other variables not included in the model that are correlated with both the dependent and the independent variable(s). Olmos and Govindasamy (2015) also point out that not taking omitted variables into account will create biased treatment effects in observational/quasiexperimental designs, thereby affecting the estimation of accurate causal effects. Therefore, it is important to be able to articulate all the reasons why a participant is assigned to a particular treatment. Failure to identify all the reasons will result in an endogeneity problem.

The counterfactual approach is important in causal inference and analysis; it imagines that individuals may occupy multiple causal states, and each has multiple potential outcomes, one for each causal state (Messeri, 2016). The counterfactual framework relies on the assumption of randomisation conditional on the covariates ("unconfoundedness") stated below (Athey et al., 2016b):

$$(Y_i(0), Y_i(1)) \perp Z_i | X_i \tag{4.1}$$

The assumption states that the pair of counterfactual outcomes, $(Y_i(0), Y_i(1))$, is independent of Z_i (treatment variable) given the covariates X_i (Olmos and Govindasamy, 2015). This assumption is also known as the Ignorable Treatment Assignment Assumption (Austin, 2011). Consequently, causal models generally include this assumption.

4.2.2 Related Work

The causal effects of the COVID-19 restrictions imposed by different federal states in Germany were investigated by Mastakouri and Schölkopf (2020). Their causal analysis was aimed at contributing to the larger effort of scientists to understand how the Covid-19 pandemic was spreading, as well as to investigate the causal role of political interventions such as social distancing. The Difference-in-differences method was used to identify causal effects of COVID-19 policies by Goodman-Bacon and Marcus (2020). They pro-

posed a difference-in-differences (DD) design to estimate causal effects in the COVID-19 context because governments implement certain policies differently. A DD design compares the results before and after a given COVID-19 related policy, to how the outcomes change in an area that did not implement the policy. Care must be taken when carrying out DD analysis because of the common trends assumption violations, which may appear in COVID-19 contexts (Goodman-Bacon and Marcus, 2020). DD designs may be affected by time lags between exposure to SARS-CoV-2 and recorded infections, as well as variations in person-to-person transmission, and the possibility that the effects of policies may be different over time.

The maximum likelihood approach that is implemented in the change-point R package (Killick and Eckley, 2014) was used to detect change points in population mobility trends in India (Ramachandra, Vikas and Sun, Haoqiao, 2020). The change-points were caused by the Covid-19 interventions imposed by the government of India. We extend the work by Ramachandra, Vikas and Sun, Haoqiao (2020) and propose a hybrid model that utilises an LSTM autoencoder and a kernel quantile estimator (Sheather and Marron, 1990; Tran et al., 2019) to automatically detect change-points. In addition, we used the BSTSM to estimate the causal effect of a change-point. According to Brodersen et al. (2015), the following assumptions must be met when using the BSTSM to estimate the treatment effects: (i) there exists a time series that is not affected by the intervention, i.e., the control; (ii) the relationship between time series affected by the intervention, i.e., the response and the control are stable during the post-intervention period. When these assumptions are met, the BSTSM is used to construct a time series model, perform posterior inferences on the counterfactual, and then return a treatment effect for a given response and control time series. Parametric models such as the difference-in-difference may have restrictive assumptions, which may make them harder to implement (Wijeyakulasuriya et al., 2020). Therefore, we make the contributions listed below.

(i) We develop a hybrid model that consists of a long-short term memory autoencoder (LSTMAE) and the kernel quantile estimator (KQE) algorithm to automatically detect change points from a time series or a sequence of values,

(ii) We compare the change points detected by our proposed model, the long-

short term memory autoencoder (LSTMAE) that is combined with a kernel quantile estimator (LSTME and KQE) to the maximum likelihood algorithm, Bayesian analysis models and linear regression models

(iii) We estimate the causal effect of a change-point or intervention using the Bayesian structural time series model (BSTSM) that has fewer assumptions.

The rest of this paper is organized as follows: Section 4.3 describes briefly the materials and methods used, Section 4.4 describes the experiments, Section 4.5 presents the results and analysis, 4.6 provides a discussion of the results, and Section 4.7 concludes the paper.

4.3 Materials and Methods

4.3.1 Data

The publicly available data sets used in this study are the ACAPS COVID-19 Government Measures data set (ACAPS., 2020) and the Google COVID-19 Community Mobility Reports (Google LLC., 2020). Comprehensive information reported by countries, that details the different governments' interventions to control the spread of COVID-19 is captured in the ACAPS COVID-19 Government Measures data set. The data set reports on the following categories of interventions; lockdowns, movement restrictions, social distancing, social and economic measures, and public health measures. In addition, government measures of different severity are included. Google COVID-19 Community Mobility Reports show graphs of population mobility trends over time, across six different categories of places such as groceries and pharmacies, transit stations, retail and recreation, workplaces, parks, and residential. These reports compare the changes in the number of visits and duration of stay with baseline days. The baseline days are based on the normal movement values for a particular weekday expressed as the median. This median value is taken over five weeks (January 3, 2020, to February 6, 2020) (`https://ourworldindata.org/covid-mobility-trends`). Thus, population mobility measures the number of visits and duration of stay for a particular day relative to a baseline value for that day of the week from January 3, 2020, to February 6, 2020. This means that measuring the changes for a

particular day and comparing them with a normal weekday is important since people's routines during weekdays differ from their routines during weekends. However, we do not consider changes in population mobility that are a result of seasonal variations because population mobility to grocery and pharmacy, as well as retail and recreation places usually increases during month-ends and paydays. In this paper, we are only interested in changes that can be explained by the pandemic and government interventions, and not changes that reflect seasonal movements. Consequently, we use South Africa's government interventions and community mobility reports as a case study and detect all change points between February 15, 2020 and July 31, 2020. Thereafter, we restrict ourselves to change-points that correspond to national lockdowns. We then evaluate the effects of these national lockdowns on population mobility in all six categories of places.

4.3.2 Long Short-Term Memory Networks

Long short-term memory (LSTM) networks which were introduced by Hochreiter and Schmidhuber (1997) can learn long-term dependencies through recurrently connected subnets known as memory blocks. LSTMs are a special kind of RNN that can detect change-points. A recurrent neural network attempts to model a time or sequence-dependent variable (Goodfellow et al., 2016). The most important function of an LSTM is to forget irrelevant parts of the previous state, selectively update a current state, and then output certain parts of the present state that are relevant to future states. This solves the vanishing gradient problem (Hochreiter and Schmidhuber, 1997) common in RNNs by updating a state and then propagating forward some parts of the state that are relevant to future states. Therefore, LSTMs become much more efficient than RNNs, as there is no extended chain of backpropagation as seen in RNNs. More details on the LSTM can be found in Goodfellow et al. (2016); Hochreiter and Schmidhuber (1997); Chen et al. (2020a). The idea of using LSTMs to detect change-points is to build an LSTM encoder–decoder structure, and then apply it to sequential data to reconstruct the input data.

4.3.3 Change-Point Detection

Given a time series, $x_{1:n} = \{x_1, \ldots, x_n\}$, a single change-point is said to occur when the properties of $\{x_1, \ldots, x_\tau\}$ and $\{x_{\tau+1}, \ldots, x_n\}$ at a time or at a point,

$\tau \in \{1, \ldots, N-1\}$ are statistically different in the mean, variance, or regression structure (Rohrbeck, 2013). In addition, the single change-point ideas can be used to detect multiple change-points. Therefore, if there are several change-points, m, and corresponding locations, $\tau_{1:m} = (\tau_1, \ldots, \tau_m)$, then the location of each change-point is an integer between 1 and $n-1$. If $\tau_0 = 0$ and $\tau_{m+1} = n$, then for ordered change-points: $\tau_i < \tau_j \Leftrightarrow i < j$, $m+1$ segments are created by splitting the m change-points. The i-th segment will contain the data $y_{(\tau_{i-1}+1):\tau_i}$. In addition, a set of parameters summarises each segment. For example, the set of parameters $\{\theta_i, \phi_i\}$ is associated with the i-th segment, where ϕ_i are nuisance parameters that should be evaluated when estimating the parameters, θ_i that contain the changes. Thereafter, we determine the number of segments that represent the data and evaluate the location as well as the number of change-points that are related to each segment.

4.3.4 Detecting Change-Points Using LSTMAE and KQE

Figure 4.1 shows a schematic representation of the proposed hybrid LSTM autoencoder (LSTMAE) neural network model to detect change-points. In Figure 4.1, we show an LSTMAE that is organised into an architecture called an Encoder–Decoder (Tan et al., 2019) LSTM. The LSTMAE can support input sequences of variable length and predict or output sequences of variable length. The motivation for using the LSTM autoencoder model in detecting change-points is that we want to use the weights obtained from the "normal" sequence to represent the training data well. The same weights are then used on the test data, and the prediction errors are then used to identify change-points.

We let $\{x_1, \ldots, x_n\}$, be a time series or sequence of data. We refer to this sequence as the normal sequence. Thereafter, we train the autoencoder on the normal data so that we reconstruct a new sample. The input is compressed by the encoder into a latent representation, and then the decoder reconstructs the input from the latent space representation. We compare the input and the reconstructed output to calculate the prediction or reconstruction errors, $e_i = ||\hat{x}_i - x_i||$, and obtain a vector of errors $\{e_1, \ldots, e_n\}$, for $i \in (1, \ldots, n)$. Our change-point detection technique uses these reconstruction errors. The distribution of these errors is assumed to be a multivariate Gaussian distribution (Tran et al., 2019). The authors argue that the assumption may be difficult

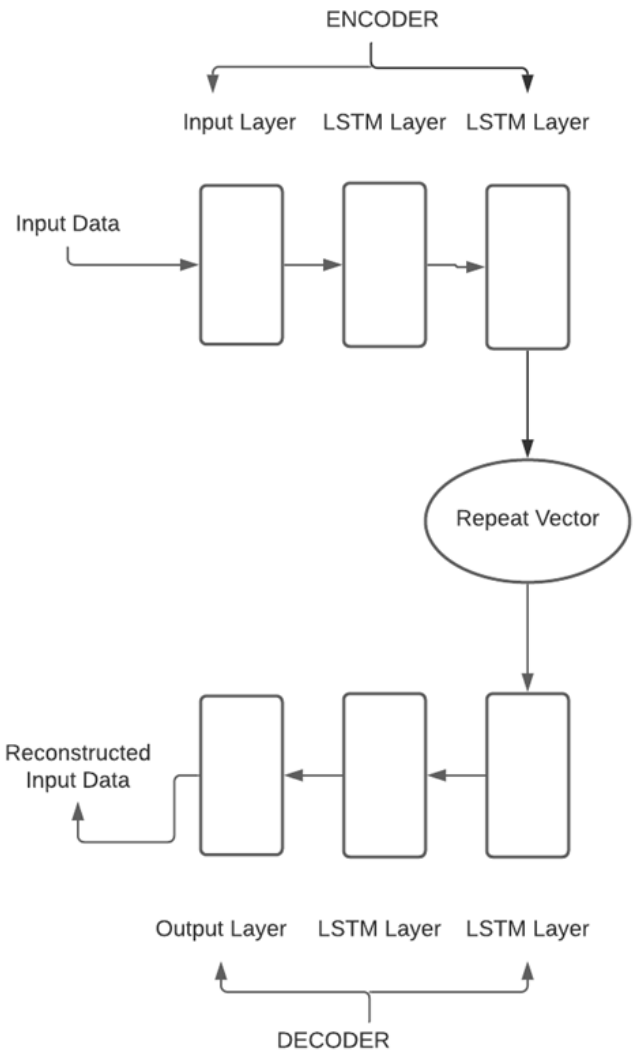


Figure 4.1: Schematic representation of the long-short term memory (LSTM-MAE) network.

to satisfy in practice because the distribution of the reconstruction errors is often not known. In this paper a nonparametric method is proposed that uses the kernel quantile estimator (Sheather and Marron, 1990; Tran et al., 2019) to estimate the threshold τ that is then used to detect an anomaly using the reconstruction errors. To estimate τ we consider the ordered error or reconstruction vectors $\{e_1 \leq e_2 \leq \cdots \leq e_n\}$ from a sample of size m. Given a density function, K, that is symmetric about zero, the threshold τ can be described as a kernel quantile estimator that can be evaluated using (Equation

4.2) (Sheather and Marron, 1990):

$$\tau_p = \sum_{i=1}^{n} \left[\int_{\frac{i-1}{n}}^{\frac{i}{n}} \frac{1}{h} K\left(\frac{t-p}{h}\right) dt \right] e_{(i)}, \tag{4.2}$$

where the bandwidth, $h > 0$, is an important parameter that regulates the extent of the smoothing used in a sample of size m, and p: $0 < p < 1$. (Tran et al., 2019) point out that there are several versions for the approximation of τ_p and the choice of K has little effect on the performance of the estimation. The kernel function that will be used in this paper is the widely used Gaussian of zero mean and unit variance (Guidoum, 2015), and it is given by:

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-u^2}{2}\right). \tag{4.3}$$

The assumption of the distribution of the reconstruction errors is not required when using a quantile kernel estimator (Guidoum, 2015). Moreover, the smoothness of the density estimate is significantly influenced by the bandwidth, in contrast to K, which does not have any influence on the smoothness of the density estimate. Good results can be obtained by using different functions, as K is not very sensitive to the shape of the estimator (Guidoum, 2015). However, choosing an efficient methodology for computing h for an observed data sample is very important in practice (Guidoum, 2015). This is because the bandwidth significantly affects the shape of the corresponding estimator. For example, an under-smoothed estimator will result from using a small bandwidth h and an over-smoothed estimator further away from the function to be estimated will result from using a large bandwidth h. Sheather and Marron (1990); Tran et al. (2019) apply an asymptotically optimised bandwidth h_{opt}, which is expressed as:

$$h_{opt} = \left(\frac{p(1-p)}{m+1}\right)^{\frac{1}{2}} \tag{4.4}$$

We use the value of τ_p to detect a change-point in a sequence or time series. At another time-step t, a reconstruction error is calculated by using $||\hat{x_t} - x_t||$, for a new observation, x_t, reconstruction $\hat{x_t}$ of x_t, that is predicted from the trained model. A candidate change-point, x_t is identified if $e_t > \tau_p$. After detecting a candidate change-point and its position, the causal effect of the

change-point or intervention on population mobility is then evaluated.

There are other methods that are used to detect change points such as; 1. the maximum likelihood approach (Killick and Eckley, 2014). We implement the likelihood ratio approach to detect change-points by using the change-point R package (Killick and Eckley, 2014), 2. Bayesian analysis (Barry and Hartigan, 1993), which is implemented in R using the bcp package (Erdman and Emerson, 2007), and 3. the structural changes in linear regression models (Zeileis et al., 2001), which are implemented using the algorithm, **breakpoints()** that is available in the R package **strucchange** (Erdman and Emerson, 2007).

4.3.5 Estimating Causal Effects Using Bayesian Structural time series Models

After detecting the change-points, we use the BSTSM to assess the effect of an intervention (Brodersen et al., 2015). In this case, the difference between actual times series and the counterfactual time series estimates after the intervention is used to assess the causal effect of an intervention. This difference gives the semiparametric Bayesian posterior distribution for the causal effect. The counterfactual is estimated at the point where there is a change-point because at that point the time series before and after the change-point differ in the mean or variance. The BSTSM framework uses the available prior knowledge about the model parameters to determine the counterfactual. Additionally, the framework also uses state-space time series models that include linear regressions of contemporaneous predictors (Brodersen et al., 2015).

Structural time series models, for example, the BSTSM are state-space models which can be represented by the following equations (Brodersen et al., 2015):

$$y_t = Z_t^T \alpha_t + e_t, \tag{4.5}$$

$$\alpha_{t+1} = T_t \alpha_t + R_t \eta_t, \tag{4.6}$$

where $e_t \sim N(0, \sigma_t^2)$ and $\eta_t \sim N(0, Q_t)$ are independent of all other unknowns. The observation equation that links the observed data y_t to a latent d-dimensional state vector α_t is shown in (10). The state equation that regulates the progression of the state vector α_t over time is shown in (11).

Following Brodersen et al. (2015), we consider y_t as a scalar quantity, Z_t as a d-dimensional output vector, T_t as a $d\ x\ d$ transition matrix, R_t as a $d\ x\ q$ control matrix, e_t as a scalar observation error with noise variance α_t, and η_t as a q-dimensional system error with a $q\ x\ q$ state-diffusion matrix Q_t, where $q \leq d$. The BSTSM framework is used to learn these parameters and thereafter the Markov chain Monte Carlo (MCMC) technique, and a Gibbs sampler are employed to perform posterior inference. An estimate of the causal effect is calculated as the difference between the counterfactual (predicted) and the actual response during the post-intervention period.

The BSTSM use the state-space models defined in Equations 4.5 and 4.6 as well as flexible Bayesian priors to fit a time series model pre-intervention (Brodersen et al., 2015). The counterfactual is then predicted using the fit model. Estimation of causal effect and statistical significance tests of an intervention using BSTSM can be done in R using the **CausalImpact R** package (Brodersen et al., 2015). The vignette is from (`https://cran.r-project.org/web/packages/CausalImpact/vignettes/CausalImpact.html`), and the paper by Brodersen et al. (2015) give detailed discussions of BSTSM. We perform Bayesian structural time series causal inference using the experiments described below.

4.4 Experiments

We used the ACAPS data set (ACAPS., 2020) and the Google COVID-19 Community Mobility Reports (Google LLC., 2020) data set. We implemented our proposed model (LSTMAE) described in Section 4.3.4 to identify change-points in all the six categories of places. The concept of using LSTMAE for change-point detection was taken from successful applications of LSTME in anomaly detection (Farahnakian and Heikkonen, 2018, 2019; Kherlenchimeg and Nakaya, 2018; Nguimbous et al., 2019). Python 3 was used to create and train our LSTMAE neural network model. TensorFlow (Abadi et al., 2016) was used as our back end, and Keras (Chollet et al., 2017) was used as our core model development library. Subsequently, data sets for training and testing our LSTMAE were defined. The data were split into a first part, which is the "normal" data or training data, without any change-point. This data set spans from February 15, 2020, to March 26, 2020. These dates include the point where the South African government declared a "national state of

disaster" on March 15, 2020. The test data were from March 27, 2020, to April 30, 2020, including the point where the government ordered all South Africans into a national lockdown for 21 days. For each of the 6 categories of places, we plotted the whole data set from February 15, 2020, to March 26, 2020, to visually check for the existence of any possible change-points.

We normalised and reshaped the data into a suitable input format for the LSTMAE neural network. LSTM cells expect a three-dimensional tensor as input. The LSTMAE neural network architecture used is shown in Figure 4.1. The first set of layers called the encoder creates the compressed representation of the input data. Subsequently, the compressed representational vector is distributed throughout the decoder time steps by a repeat vector layer. The reconstructed input data are produced by the final output layer of the decoder. The efficient Adam optimiser (Kingma and Ba, 2014) is used for training the model. The mean absolute error (MAE) is used as the loss function. The model is trained for 500 epochs.

Using the kernel quantile estimator (Equation 4.2), we determine a threshold value τ_p for identifying change-points. Reconstruction errors are calculated in the training data set and in the test data set to determine where the error values exceed the threshold τ_p and thus, detect a change-point m. Once a value of m has been positively identified as a candidate change-point that represents a government intervention, the **CausalImpact R** package is used to evaluate the average causal effect of a government intervention on our outcome variable. **CausalImpact R** (Brodersen et al., 2015) is also used to test the statistical significance of the average causal effect.

We compare the change-points detected by using reconstruction errors from the LSTM autoencoder and the kernel quantile estimator to the change-points detected by the change-point R packages which detect changes in the mean or variance or both. This is done to determine whether our proposed deep learning approach detects the same change-points as detected by other methods available in practice.

4.5 Results and Analysis

4.5.1 Change-point Detection Using Different Algorithms

Figure 4.2 shows the time series mobility trends for residential areas, transit stations, parks, workplaces, grocery and pharmacy, and retail and recreation.

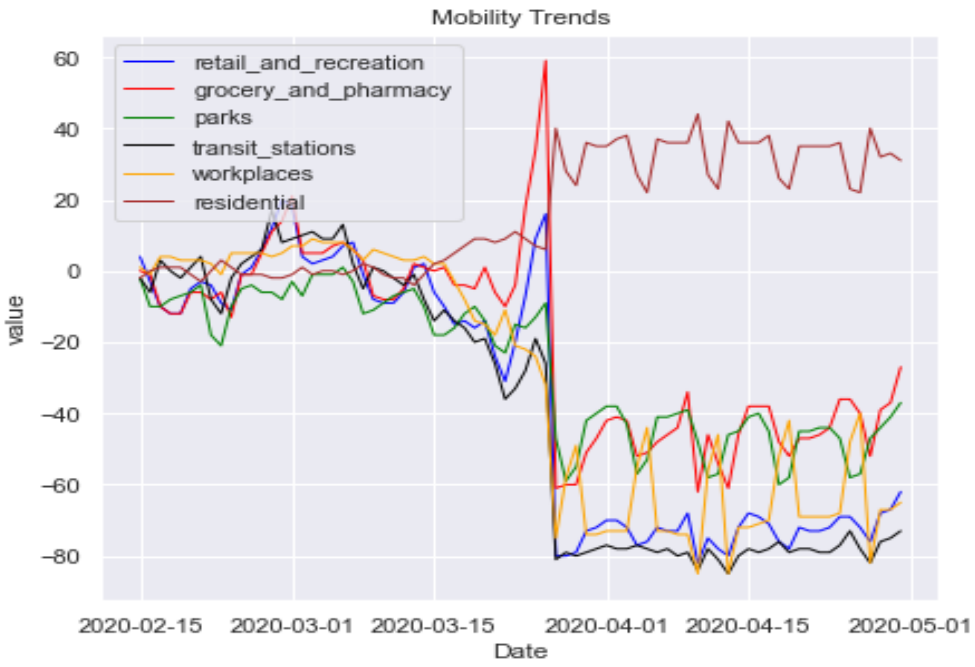


Figure 4.2: Plots of the data sets for the 6 categories of places.

The plots show that the mobility trends before and after the first intervention, a full lockdown imposed by the government of South Africa on March 27, 2020, are indeed different, indicating that March 27, 2020, is a candidate change-point.

The LSTMAE and KQE algorithm (Section 4.3.4) was trained for each of the 6 categories of places in order to detect the change-points and their positions, using the community mobility data set described in Section 4.3.1. The data set is based on the COVID-19 Community Mobility Reports by Google (Google LLC., 2020) data set and constitutes the changes in the number of visits and duration of stay corresponding to all the days from February 15, 2020, to April 30, 2020, inclusive. The LSTMAE and KQE algorithm was trained using the procedure outlined in Section 4.4. Figures 4.3–4.8 below show the change-

points that were detected using the LSTMAE and KQE algorithm, for each of the six categories of places.

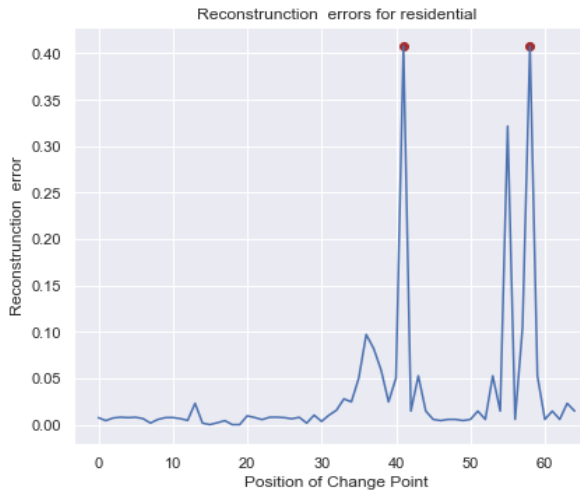


Figure 4.3: Reconstruction errors for residential.

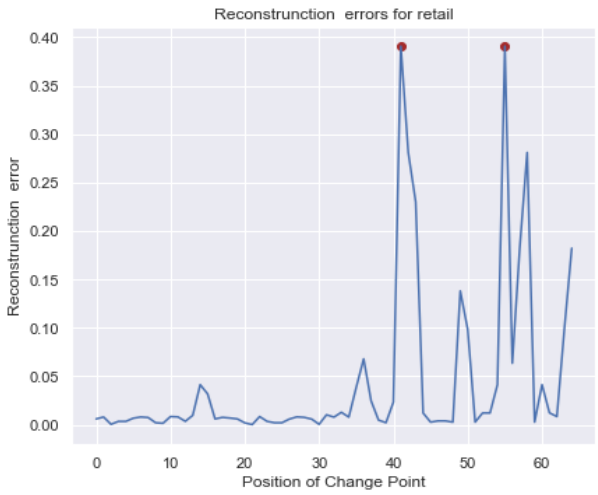


Figure 4.4: Reconstruction errors for grocery.

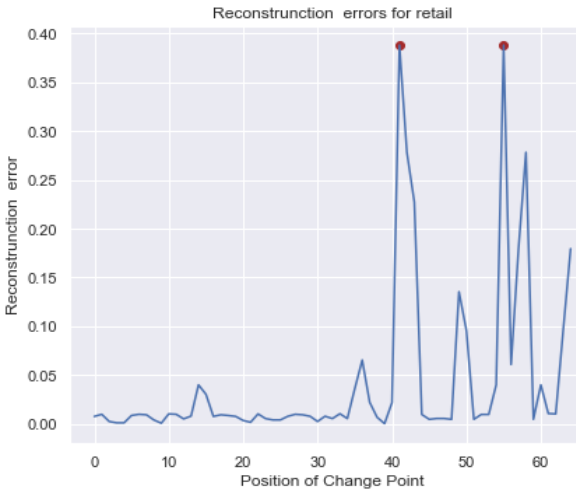


Figure 4.5: Reconstruction errors for retail.

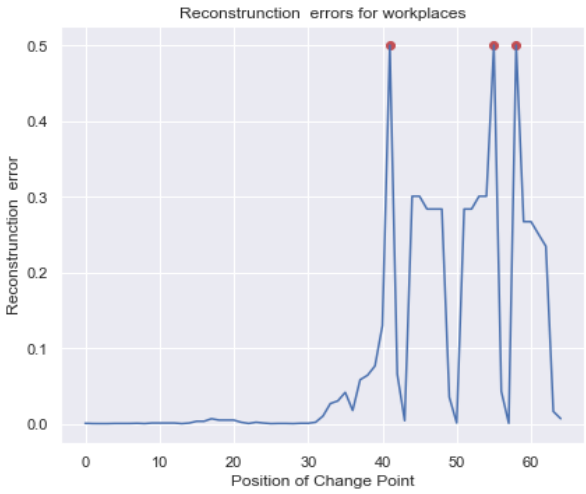


Figure 4.6: Reconstruction errors for workplaces.

The horizontal axis shows the positions/locations of the change-points detected by the hybrid LSTMAE model. Thereafter, the detected change-points are matched with the exact dates of the interventions. February 15, 2020, corresponds to position 1 and April 19, 2020, corresponds to position 65. The

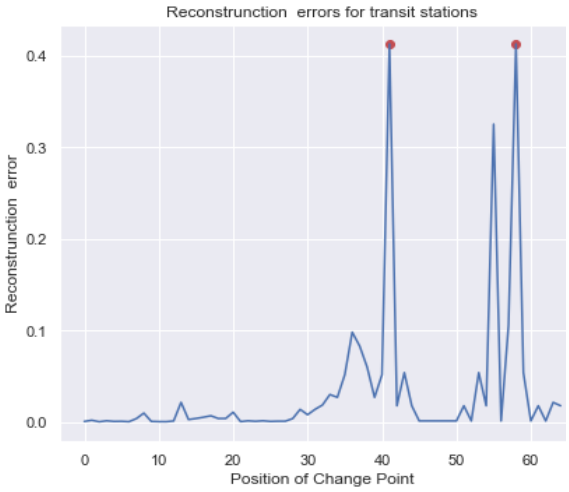


Figure 4.7: Reconstruction errors for transit.

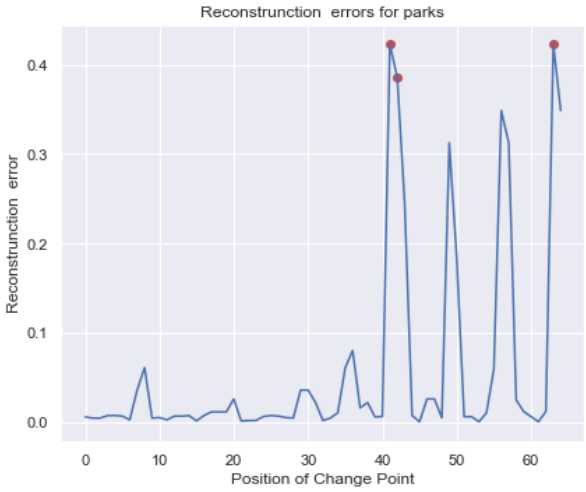


Figure 4.8: Reconstruction errors for parks.

change-points are shown as dots at the points where the reconstruction errors are highest. Figures 4.3, 4.5–4.8 all contain the change-point detected at location 42 (March 27, 2020). For grocery and pharmacy, the change-point was detected at location 41, that is, a day before the start of the lockdown. The change-points at locations 41 and 42 coincided with the first major intervention, a full lockdown (lockdown level 5) that was imposed by the government of South Africa on March 27, 2020. This shows that our hybrid model accurately detected a known intervention (change-point) on March 27, 2020.

A comparison of the change-points that were detected by LSTMAE and KQE and the commonly used **R** packages namely: change-point, Bayesian change-point (bcp), strucchange (breakpoints), from 15 February 2020 to April 19, 2020 is shown in Table 4.1.

We note that the change-point at location 42 is detected as an "anomaly" by the LSTMAE and KQE algorithm. This means that our proposed algorithm was able to detect a change in population mobility at location 42. This change-point corresponds to the first major full lockdown that was imposed on March 27, 2020, for an initial period of 21 days to curb the spread of COVID-19. For grocery and pharmacy, our proposed algorithm detected a change in mobility at location 41 on March 26, 2020. The increase in population mobility on this date was a result of people stocking up on grocery and pharmacy supplies in preparation for the full shutdown.The *change-point*, *Bayesian*

Table 4.1: A comparison of the date of occurrence and location of the change-points that were detected by the different algorithms between February 15, 2020 and April 19, 2020 inclusive.

	Method			
Category of place	*LSTMAE +KQE*	*change-point*	*bcp*	*strucchange*
Grocery & pharmacy	26/03/2020 (41), 10/04/2020 (56)	26/03/2020 (41)	23/03/2020 (38), 25/03/2020 (40), 26/03/2020 (41)	26/02/2020 (12), 26/03/2020 (41)
Parks	27/04/2020 (42), 18/04/2020 (64)	26/03/2020 (41)	26/03/2020 (41)	26/02/2020 (12), 26/03/2020 (41)
Retail & recreation	27/04/2020(42), 10/04/2020 (56), 13/04/2020 (59)	26/03/2020 (41)	24/03/2020 (39), 26/03/2020(41)	24/02/2020 (11), 07/03/2020 (22), 26/03/2020 (41)
Residential	27/04/2020(42), 10/04/2020 (56), 13/04/2020 (59)	26/03/2020 (41)	26/03/2020 (41)	15/03/2020 (30), 26/03/2020 (41)
Workplaces	27/04/2020(42), 10/04/2020 (56), 13/04/2020 (59)	26/03/2020 (41)	26/03/2020 (41)	15/03/2020 (30), 26/03/2020 (41)
Transit stations	27/04/2020(42), 10/04/2020 (56), 13/04/2020 (59)	26/03/2020 (41)	09/03/2020 (25), 20/03/2020 (35), 26/03/2020 (41)	25/02/2020 (11), 14/03/2020 (29), 26/03/2020 (41)

change-point (bcp), and *strucchange* (breakpoints) methods detected a change-point at location 41 on March 26, 2020 as shown in Table 4.1. These methods measure changes in the statistical properties before and after a change-point. This means that the statistical properties of the observations from location 1 (February 15, 2020) to location 41 (March 26, 2020) and the statistical properties of the observations from location 43 (March 28, 2020) to location 65 (April 19, 2020) are different in the mean, therefore making location 42 on March 27, 2020, a change-point.

In addition, our proposed model, the LSTMAE and KQE algorithm, detected more change-points after March 27, 2020. For example, a change-point was detected at location 56 (April 10, 2020) for retail and recreation, grocery and pharmacy, and workplaces. It also detected a change-point at location 59 (April 13, 2020) for workplaces, and transit stations, a change-point at location 64 (April 18, 2020) for parks. The change-points detected by LSTMAE + KQE at locations 56 and 59 were detected as "anomalies" that indicate unusual mobility around those dates. This increase in mobility at positions 56 (10 April 2020) and 59 (13 April 2020) could be a result of the Easter holidays for 2020. This may indicate that movement restrictions were not effectively monitored during the Easter holidays, as there was an increase in population movements on Easter Friday and Easter Monday.

We observe that the Bayesian change-point (bcp) method detected change-points at locations 38 (March 23, 2020), 40 (March 25, 2020), 41 (March 26, 2020), for grocery and pharmacy and 39 (March 24, 2020), 41 (March 26, 2020) for retail and recreation, as shown in Table 4.1. These change-

points were as a result of increased population mobility as people were stocking up on food items and other essential commodities in preparation for the full lockdown on March 27, 2020. The *strucchange* (breakpoints) method detected change-points at location 11 (February 24, 2020) for transit stations, and retail and recreation, location 12 (February 25, 2020) for parks, and grocery and pharmacy. The R package, *strucchange* (breakpoints) also detected a change-point at location 22 (March 7, 2020) for retail and recreation. For locations 11 and 12, the change-points can be attributed to the fact that most people get paid their salaries or wages on the 25th of each month in South Africa. Therefore, increased population mobility is a result of people moving around to access their salaries. The change-point at location 22 may be as a result of people moving around to access the South African Social Security Agency (SASSA) grants that are normally paid around that time. It is interesting to note that, at locations 11, 12 and 22, the LSTMAE and KQE algorithm did not detect any anomaly or change-point in population mobility across all the category of places at locations. This means that our proposed model did not detect any anomalous behaviour in the population movements from February 15, 2020, to March 26.

4.5.2 Change-Point Detection Using Different data sets

To check for the robustness of the LSTMAE and KQE algorithm, change-point, Bayesian change-point (bcp), and strucchange (breakpoints), in detecting possible change points beyond April 19, 2020, we applied these methods to two additional nonoverlapping data sets from April 20, 2020, to May 18, 2020. as well as from May 19, 2020, to June 19, 2020. The locations of the change-points that were detected by these methods from April 20, 2020, to May 18, 2020, are shown in Table 4.2. The results show that LSTMAE and KQE detected a change point at location 73 on April 26, 2020, for residential, workplaces, and transit stations. Most people receive their salaries or wages on the 25th of each month in South Africa. The increased population mobility is, therefore, a result of people moving around to access their salaries. The LSTMAE and KQE, *change-point* and *bcp* algorithms detected a change point at location 76 (30 April 2020) or location 77 (01 May 2020) (Table 4.2).

The change points at locations 76 and 77 coincided with the gradual ease of the full lockdown by the South African government (South African Government,

Table 4.2: A comparison of the date of occurrence and location of the change-points that were detected by the different algorithms between April 20, 2020 and May 18, 2020 inclusive.

	Method			
Category of place	*LSTMAE +KQE*	*change-point*	*bcp*	*strucchange*
Retail & recreation	01/05/2020 (77)	30/04/2020 (76)	30/04/2020 (76)	27/04/2020 (73), 03/05/2020 (79)
Grocery & pharmacy	30/04/2020 (76)	30/04/2020 (76)	30/04/2020 (76)	30/04/2020 (76)
Residential	27/04/2020 (73), 08/05/2020 (84)	30/04/2020 (76)	30/04/2020 (76)	30/04/2020 (76)
Workplaces	27/04/2020 (73), 01/05/2020 (77)	30/04/2020 (76)	30/04/2020 (76)	30/04/2020 (76), 08/05/2020 (84)
Parks	01/05/2020 (77)	30/04/2020 (76)	30/04/2020 (76)	27/04/2020 (73), 03/05/2020 (79)
Transit stations	27/04/2020 (73), 01/05/2020 (77)	30/04/2020 (76)	30/04/2020 (76)	30/04/2020 (76)

2020). The easing of the full lockdown was done to allow economic activities to recover gradually. Thus, from May 1, 2020, the government of South Africa adopted a deliberate, risk-adjusted and careful approach to the easing of the lockdown restrictions. The country moved from level 5, a full lockdown, to level 4 lockdown with fewer restrictions than those imposed under level 5. Our proposed model and the strucchange method detected a change point at location 84 (8 May 2020). At this point, the South African government implemented a staggered return to work plan (South Africa. Department. Public Service and Administration, 2020). Individuals were required to apply for permits to return to work. Working hours were also reviewed so that not all employees would leave or return to work at the same time. Table 4.3 shows the locations of the change-points that were detected by the different methods between May 19, 2020, to June 19, 2020.

Table 4.3: A comparison of the date of occurrence and location of the change-points that were detected by the different algorithms between May 19, 2020 to June 19, 2020 inclusive.

	Method			
Category of place	*LSTMAE +KQE*	*change-point*	*bcp*	*strucchange*
Workplaces	31/05/2020 (107)	28/05/2020 (104)	24/05/2020 (100), 28/05/2020 (104)	28/05/2020 (104) , 19/06/2020 (126)
Parks	30/05/2020 (106), 31/05/2020 (107)	30/05/2020 (106)	24/05/2020 (100), 30/05/2020 (106)	30/05/2020 (106), 14/06/2020 (121)
Transit stations	30/05/2020 (106)	28/05/2020 (104)	24/05/2020 (100)	25/05/2020 (101), 27/05/2020 (103)
Retail & recreation	30/05/2020 (106)	28/05/2020 (104)	24/05/2020 (100)	24/05/2020 (100), 27/05/2020 (103)
Grocery & pharmacy	30/05/2020 (106)	30/05/2020 (106)	30/05/2020 (106)	25/05/2020 (101), 26/05/2020 (102)
Residential	31/05/2020 (107), 06/06/2020 (113), 16/06/2020 (123)	28/05/2020 (104)	28/05/2020 (104)	26/05/2020 (102), 28/05/2020 (104), 19/06/2020 (126)

There are change-points that were detected at locations 100 (May 24, 2020), 101 (May 25, 2020), 102 (May 26, 2020), 103 (May 27, 2020), 104 (May 28, 2020) by some of the methods as shown in Table 4.3. These change-points fall on the dates that most people in South Africa receive their salaries. Our proposed model detected change-points at location 106 (May 30, 2020) for

grocery and pharmacy, parks, transit stations, and retail and recreation, and at location 107 (May 31, 2020) for workplaces, parks, and residential places. These change-points were a result of the transition from level 4 lockdown to level 3 lockdown on June 1, 2020. The government took a differentiated approach when dealing with hot-spot areas that had high rates of COVID-19 transmission and infections. The LSTMAE and KQE algorithm detected a change-point at location 113 (June 6, 2020) for residential places which coincided with the payments of SASSA grants. The model detected a change point at location 123 (June 16, 2020) for residential places. June 16, 2020, is a public holiday (Youth Day) that is recognised to commemorate the Soweto Uprising, which took place on 16 June, 1976. Our proposed model was able to detect the change in population mobility on this day as people moved within residential places socialising and commemorating the day.

4.5.3 Evaluating the Effect of the Full Lockdown Level 5 Effective March 27, 2020, on Population Mobility

A full lockdown level 5 was implemented by the government of South Africa from midnight of March 26, 2020 to April 16, 2020. There are 5 levels of the lockdown process, where level 5 is the full lockdown imposed on March 27, 2020 and Level 1 is when the country is essentially functioning normally. Under lockdown level 5, people were prohibited from leaving their homes, except for strict reasons (aside from essential workers in the response). People who broke the lockdown rules were either detained and/or fined as punishment for breaking the rules. We used the BSTSM described in Section 4.3.5 to perform time series causal inference on the six categories of places. The analysis was carried out to infer the causal effect of the South African government's COVID-19 lockdowns on population mobility in these six categories of places. As noted in Section 4.3.1, we are restricting our analysis to changes that can be explained by the COVID-19 pandemic and the government's interventions, and not changes that reflect seasonal movements. Figures 4.9–4.11 show the effect of the national full lockdown level 5 that was imposed by the South African government on March 27, 2020.

The top panels in the graphs, show the counterfactual predictions represented by the dashed line and the corresponding confidence interval for the counterfactual (the shaded part). The solid line represents the actual values observed

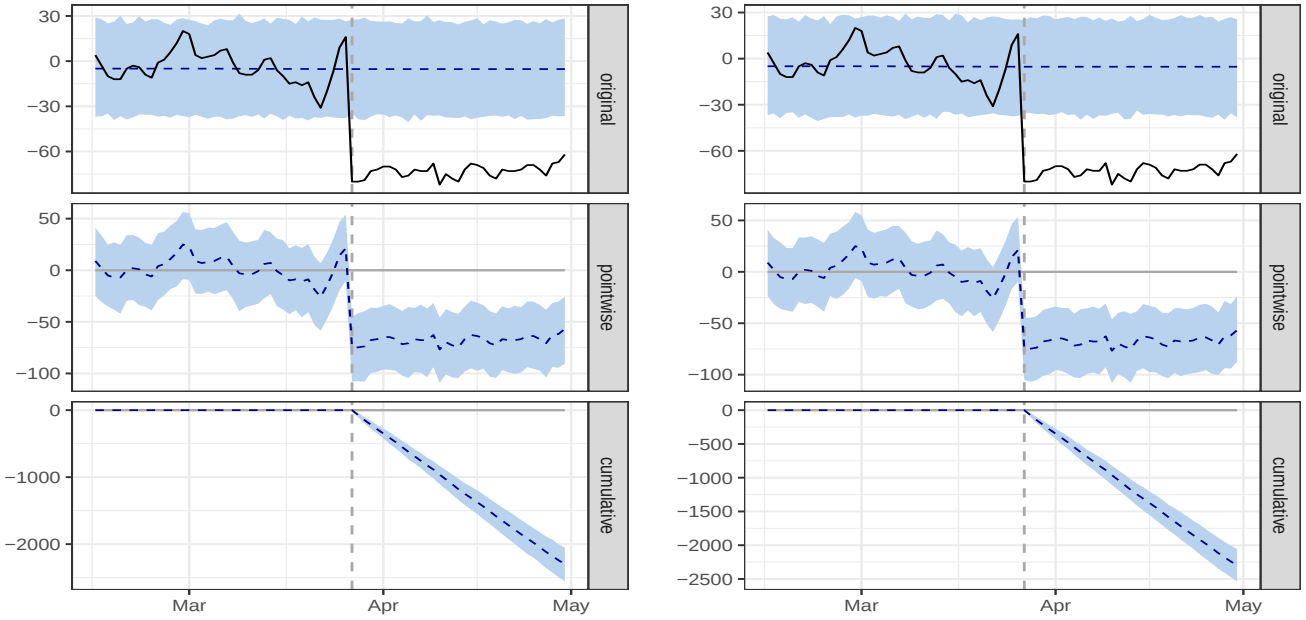


Figure 4.9: Left: Effect of the national full lockdown level 5 on grocery and pharmacy. Right: Effect of the national full lockdown level 5 on retail and recreation.

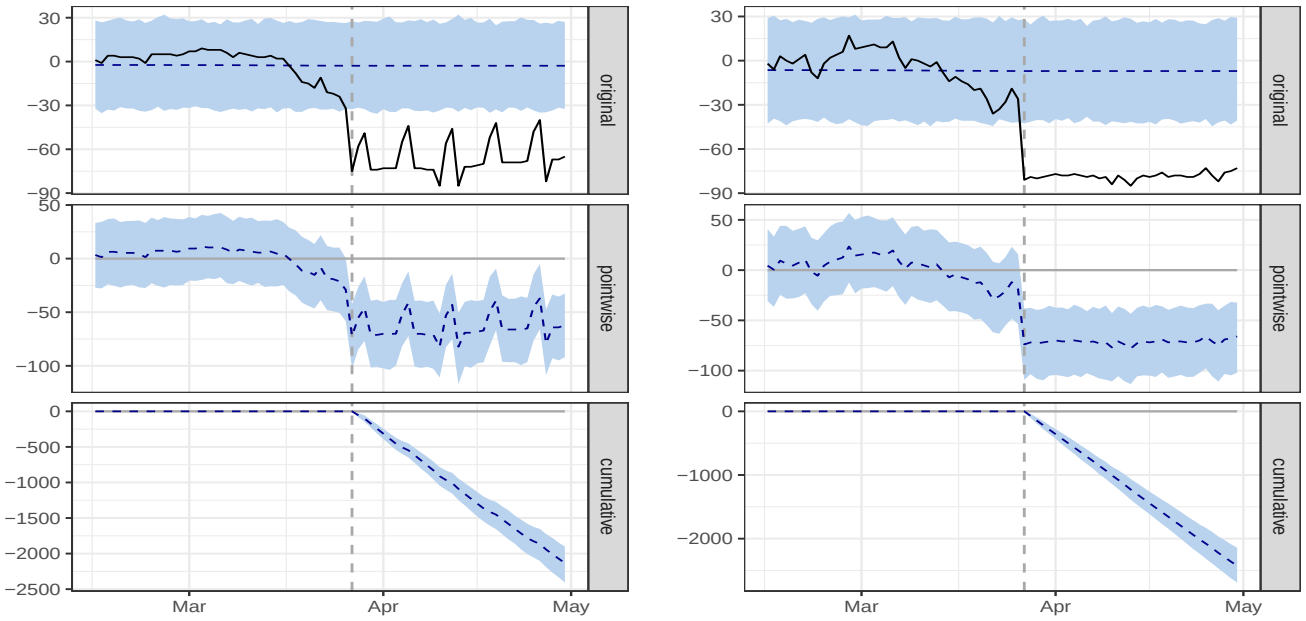


Figure 4.10: Left: Effect of the national full lockdown level 5 on workplaces. Right: Effect of the national full lockdown level 5 on transit stations.

after the intervention i.e., from the 27 March 2020 when the government ordered a full lockdown, to 30 April 2020. The difference between the actual population mobility and the counterfactual predictions of population mobility,

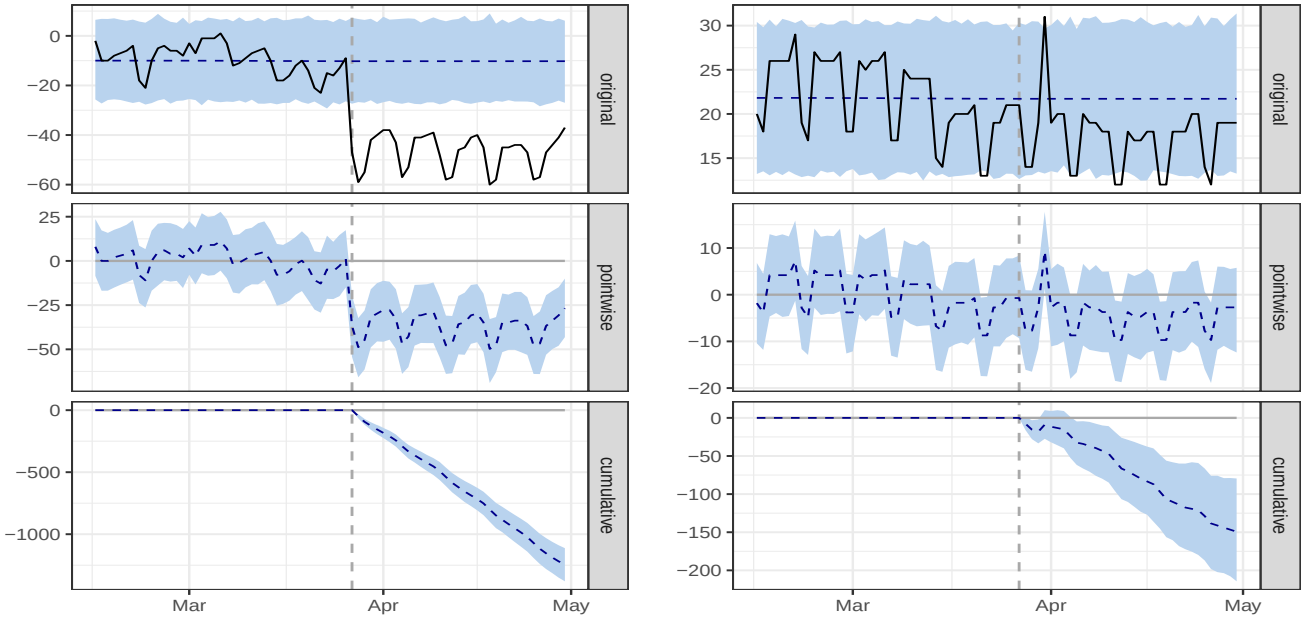


Figure 4.11: Left: Effect of the national full lockdown level 5 on parks. Right: Effect of the national full lockdown level 5 on residential places.

which represents the estimated treatment effect of the full lockdown is shown in the middle panel. The bottom panel shows a way of visualising the effects of the interventions by using a cumulative effect plot. The plot shows the cumulative treatment effect up to that day.

A visual inspection of the graphs clearly shows a change-point in the population mobility data in all the categories of places. The full lockdown imposed by the government of South Africa on March 27, 2020, resulted in much lower movements of people in the categorised places than before the full lockdown. The estimates of the causal effect of the full lockdown imposed on March 27, 2020, for each category of places are shown in Table 4.4.

Table 4.4: Causal effect of lockdown level 5 implemented 27 March 2020 for each category of places.

Category of Place	**Actual**	**Predicted**	**Causal Effect Estimate**	**95% CI**	**Relative Effect**	**95% CI**	**Bayesian one-sided *p*-values**
grocery and pharmacy	-46	0.27	-46.27	[-54, -39]	-17137.04%	[-19850%, -14306%]	0.001
retail and recreation	-73	-5.3	-67.7	[-75, -60]	-1277.36%	[-1417%, -1136%]	0.001
Workplaces	-66	-2.9	-63.1	[-70, -55]	-2175.86%	[-2452%, -1928%]	0.001
Parks	-47	-10	-37	[-40, -33]	-370.00%	[-395%, -319%]	0.001
Transit Stations	-78	-7.1	-70.9	[-80, -63]	-998.59%	[-1130%, -893%]	0.001
Residential	17	22	-5	[-6.4, -2.4]	-22.73%	[-30%, -11%]	0.001

The actual column shows the average (across time) population mobility during

the pre-intervention period (February 15, 2020 to March 26, 2020). The predicted column shows the predicted counterfactual during the post-intervention period, which indicates how the population movements would have behaved without the lockdown in place. For example, during the post-lockdown period, the population mobility for grocery and pharmacy was approximately equal to an average actual value of -46. However, an average predicted or counterfactual value of 0.27 would have been obtained in the absence of an intervention. The causal effect estimate column is the estimated average causal effect of the lockdown. An estimate of the causal effect of the lockdown on the response variable is found by subtracting the predicted (counterfactual) average value from the actual average value. Therefore, for grocery and pharmacy, the causal effect of the lockdown on population mobility is -46.27, with a 95% posterior confidence interval of $[-54, -39]$. These results show that there was a decrease in population mobility in places of grocery and pharmacy locations after the lockdown compared to the baseline days. Since the 95% posterior confidence interval does not include 0, we conclude that the lockdown imposed on March 27 2020 had a causal effect on population mobility in grocery and pharmacy places. The mobility of the population in supermarkets and pharmacy decreased by 17 137.04%. The 95% interval of this percentage is [−19,850%, −14,306%] with a Bayesian one-sided p-value $= 0.001 < 0.05$. This means that the probability of obtaining the causal effect by chance is very small. Thus, the causal effect is statistically significant.

Similarly, Table 4.4 shows that, the population movements at transit stations, retail and recreation, workplaces, and parks had decreased and were all significant at 5% level of significance The results show that there are smaller changes in population mobility for residential places compared to the other categories. Data for residential places show how time spent at home changes. On the contrary, other places show how the total number of visitors changes (`https://ourworldindata.org/covid-mobility-trends`).

4.5.4 Evaluating the Effect of Lockdown Level 4 Effective 1 May 2020 on Population Mobility

On May 1 2020, the full lockdown level 5 imposed on March 27 2020 was gradually eased. South Africa began a measured and phased recovery of economic activity. The country implemented a risk-adjusted strategy through which the

government took a thoughtful and careful approach to the ease the lockdown restrictions imposed on March 27 2020. We evaluated whether our proposed algorithm was able to detect the change or transition from level 5 full lockdown to Level 4 lockdown. Under lockdown level 4, movement restrictions were eased and all South Africans were required to wear a face mask whenever they left their homes. Some businesses could resume operations under specific conditions. However, the government encouraged businesses to implement work-from-home strategies where possible. Some activities, such as walking, jogging, and cycling, were allowed between 9 am and 6 am. Figures 4.12–4.14 show the effects of easing the lockdown from level 5 to level 4. A visual inspection of the graphs clearly shows that the easing of the full lockdown level 5 to level 4 on May 1, 2020, resulted in an increase in the movements of people in most places, except for residential places that showed no change in population movements. The causal effect estimates of the transition from lockdown level 5 to lockdown level 4 are shown in 4.5.

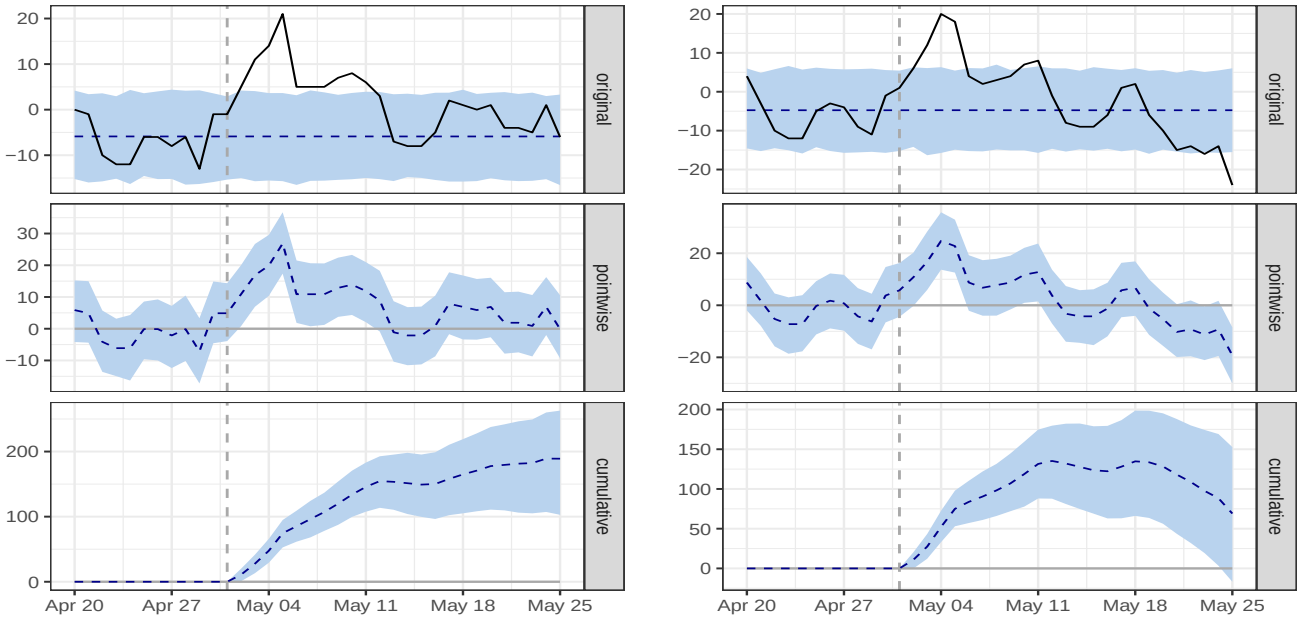


Figure 4.12: Left: Effect of the national full lockdown level 4 on grocery and pharmacy. Right: Effect of the national full lockdown level 4 on retail and recreation.

Population movements increased in retail and recreation places, transit stations, grocery and pharmacy places, and workplaces. The causal effect estimates were all significant at a 5% level of significance. Therefore, we conclude

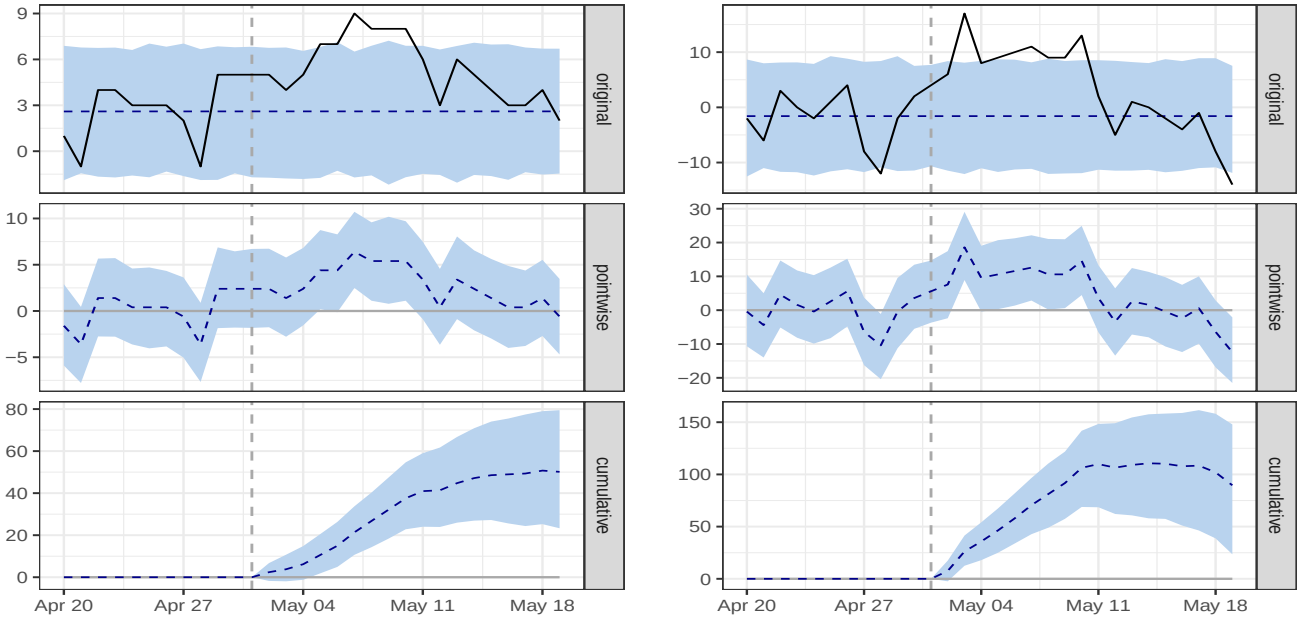


Figure 4.13: Left: Effect of the national full lockdown level 4 on workplaces. Right: Effect of the national full lockdown level 4 on transit stations.

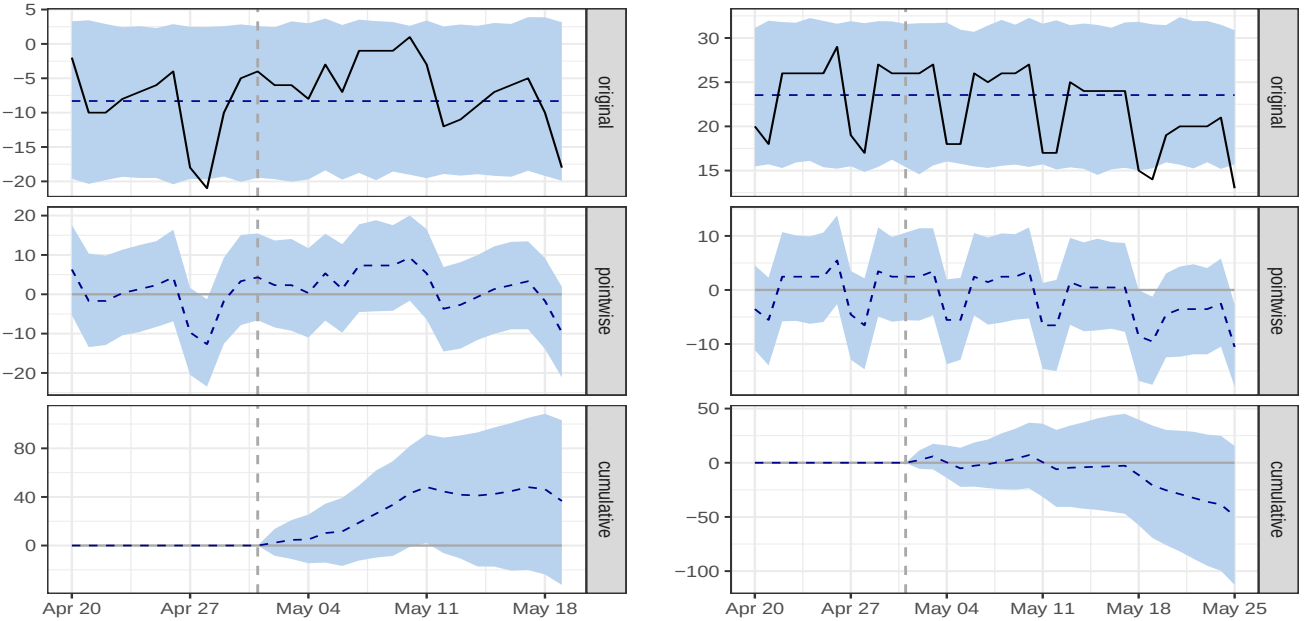


Figure 4.14: Left: Effect of the national full lockdown level 4 on parks. Right: Effect of the national full lockdown level 4 on residential places.

that the change from level 5 full lockdown to level 4 on 1 May 2020 influenced population mobility in these categories of places. The results show that, for residential places and parks, the population movements under lockdown level 4 did not significantly change from the population movements under lockdown

Table 4.5: Causal effect of lockdown level 4 implemented May 1, 2020 for each category of places.

Category of Place	Actual	Predicted	Causal Effect Estimate	95% CI	Relative Effect	95% CI	Bayesian one-sided p-values
grocery and pharmacy	3.6	-5.9	9.5	[6.1, 13]	161.02%	[103%, 217%]	0.001
retail and recreation	2.7	-4.8	7.5	[3.5, 11]	156.25%	[72%, 232%]	0.001
Workplaces	5.4	2.6	2.8	[1.3, 4.4]	107.69%	[50%, 170%]	0.001
Parks	-6.3	-8.3	2	[-1.8, 5.7]	24.10%	[-69%, 22%]	0.158
Transit Stations	3.4	-1.6	5	[1.3, 8.2]	312.50%	[82%, 518%]	0.001
Residential	22	24	-2	[-4.7, 0.65]	8.33%	[-20%, 2.8%]	0.063

level 5. For example, under lockdown level 4, public parks, nature reserves and beaches remained closed, hence the insignificant change in population mobility.

4.5.5 Evaluating the Effect of Lockdown Level 3 Effective 01 June 2020 on Population Mobility

On June 1, 2020, South Africa was moved from lockdown level 4 to lockdown level 3. The government took a differentiated approach to deal with COVID-19 hotspot areas that had far higher levels of infection and transmission. Some of the measures taken by the government, included allowing wholesale and retail trades (including stores, spaza shops, and informal traders) to fully open. Additionally, universities could safely accommodate no more than a third of the student population on campus. Under lockdown level 3, all manufacturing, mining, construction, financial services, professional and business services, information technology, communications, government services, and media services could operate subject to hygiene and social distancing measures. Figures 4.15–4.17 show the effect of the change from lockdown level 4 to lockdown level 3. A visual inspection of the graphs clearly shows that the ease of the full lockdown level 4 to level 3 on June 1, 2020 resulted in an increase in people's movements in most places, except residential places that did not show any change in population movements.

The causal effect estimates of the changeover from lockdown level 4 to lockdown level 3 is shown in Table 4.6. There was a significant increase in the number of visitors to places like grocery and pharmacy, retail and recreation, workplaces, and transit stations. Therefore, we conclude that the changeover from lockdown level 4 to level 3 on June 1, 2020, influenced population mobility in these categories of places. However, for residential places and parks, population movements did not change significantly from mobility trends witnessed

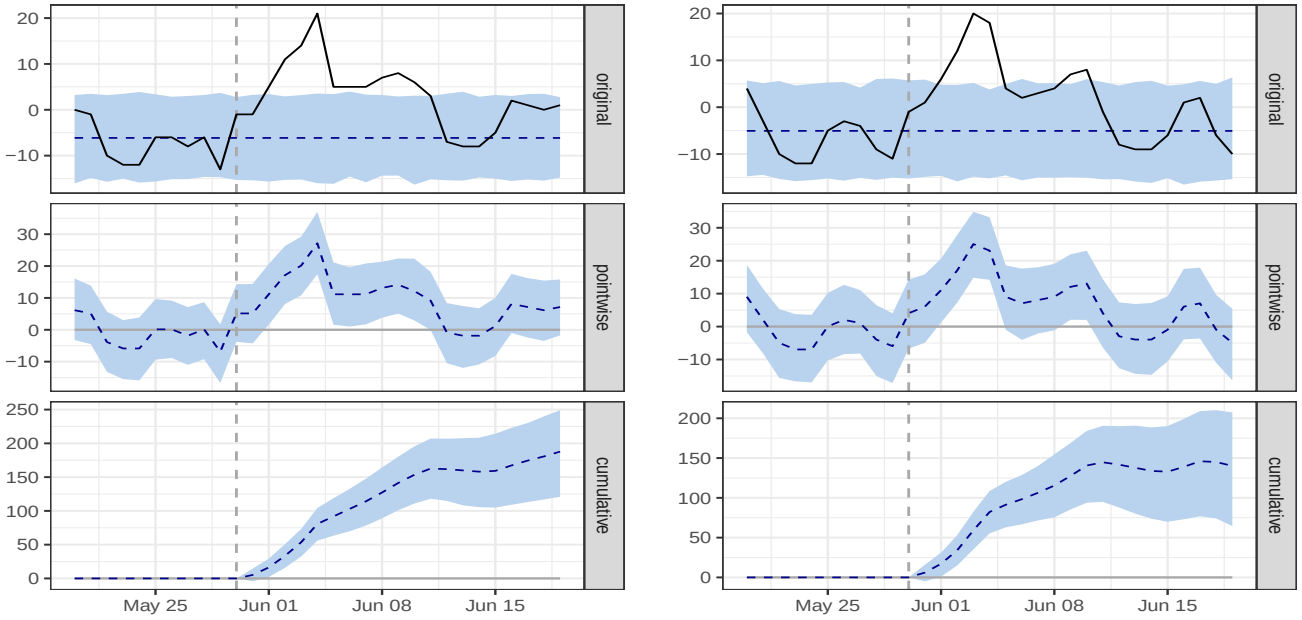


Figure 4.15: Left: Effect of the national full lockdown level 5 on grocery and pharmacy. Right: Effect of the national full lockdown level 5 on retail and recreation.

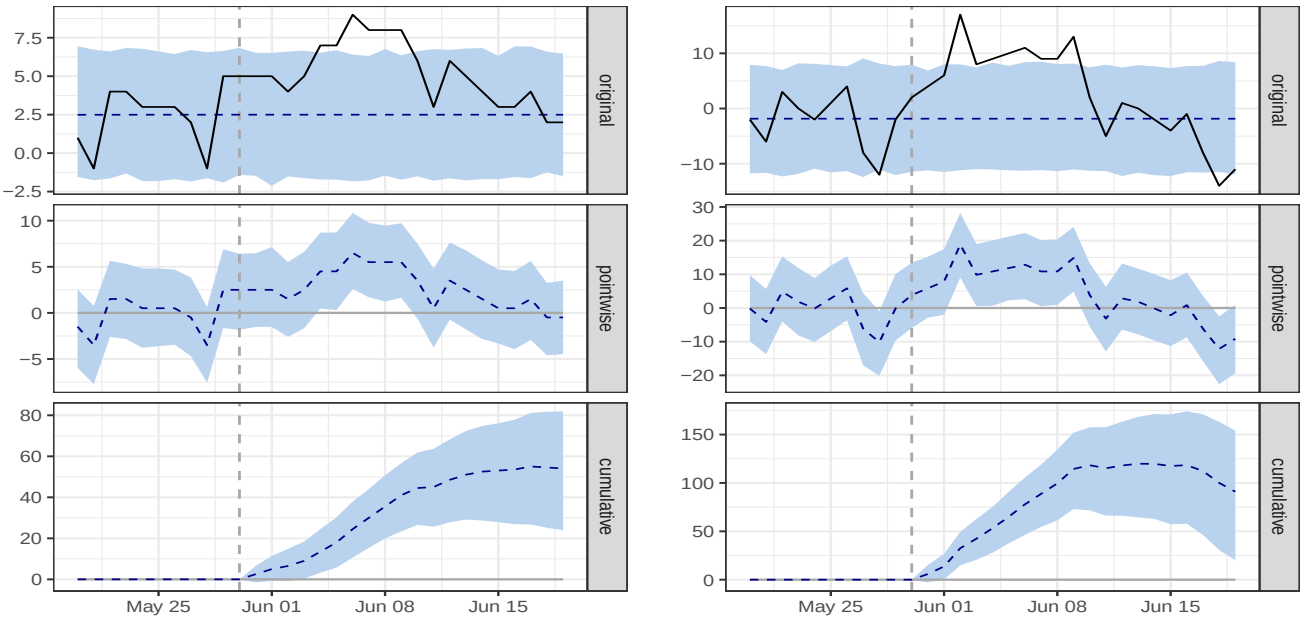


Figure 4.16: Left: Effect of the national full lockdown level 3 on workplaces. Right: Effect of the national full lockdown level 3 on transit stations.

under lockdown level 4.

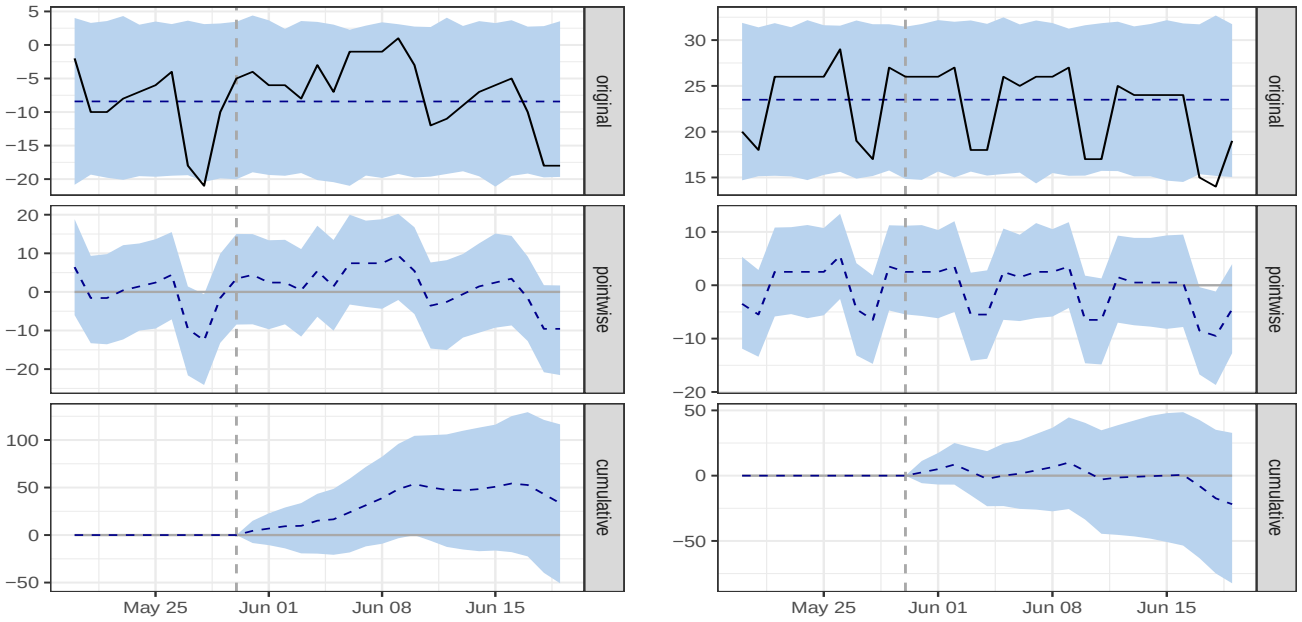


Figure 4.17: Left: Effect of the national full lockdown level 3 on parks. Right: Effect of the national full lockdown level 3 on residential places.

Table 4.6: Causal effect of lockdown level 3 implemented June 1, 2020 for each category of places.

Category of Place	**Actual**	**Predicted**	**Causal Effect Estimate**	**95% CI**	**Relative Effect**	**95% CI**	**Bayesian one-sided *p*-values**
grocery & pharmacy	3.2	-6.1	9.3	[6.3, 13]	152.46%	[103%, 205%]	0.001
retail & recreation	1.9	-5.1	7	[3.2, 10]	137.25%	[64%, 205%]	0.001
Workplaces	5.2	2.5	2.7	[1.2, 4.1]	108.00%	[48%, 164%]	0.001
Parks	-6.8	-8.4	1.6	[-2.5, 5.8]	19.05%	[-69%, 30%]	0.209
Transit Stations	2.7	-1.8	4.5	[1, 7.7]	250.00%	[55%, 417%]	0.007
Residential	22	23	-1	[-4.1, 1.6]	4.35%	[-18%, 7%]	0.213

4.6 Discussion

In this paper we have, 1. successfully developed a model that integrates a long short-term memory network (LSTM) and a kernel quintile estimator (KQE) to detect change-points in time series or sequential data; 2. developed a non-parametric that does not require advanced knowledge of the true number of change-points; 3. developed a model that can detect abrupt changes such as lockdowns that are sufficiently "large" regardless of the noise levels in the data and the size of the data. This is crucial to avoid having several false positives.

We have used a data point reconstruction error, which is the error between the original value of the data point and its low-dimensional reconstruction,

to detect a change point as an anomaly. Change-point detection methods that detect changes in parameters such as the mean or variance do not detect isolated abnormal points such as anomalies and should be supplemented with a Shewhart control chart (Taylor, 2000). Our algorithm addresses this shortcoming as the change-points are detected as anomalies in the time series and the algorithm does not depend on the statistical properties such as the mean before or after a change-point. The key factor of the performance of reconstruction-based methods is the threshold, which represents the value of the reconstruction error where a data point is labelled as an anomaly or change-point. Thus, we do not estimate changes in the mean process or changes in the mean and/or variance of a classical model. However, work has been done in the past to detect change-points in model parameters (Eckley et al., 2011; Niekum et al., 2014; Aminikhanghahi and Cook, 2017; Gao et al., 2020). If the underlying functional form is specified correctly, the parametric techniques become efficient. Our proposed model does not make strong assumptions about a specific functional form; thus, the model can freely learn any functional form from the training data.

Our method, the LSMAE and KQE was successful in determining the number and exact time of the major change-points in the population mobility during the period from February 15, 2020, to March 31, 2020. The proposed model successfully detected the change-point as a result of the full lockdown level 5 that was imposed by the South African on March 27, 2020. We used other data sets beyond April 30, 2020, to determine if our model was able to capture other different levels of the lockdown. Using a data set from April 20, 2020, to May 25, 2020, our model successfully detected the change from lockdown level 5 to lockdown level 4 on May 1, 2020. We used another data set from May 20, 2020, to June 19, 2020, and our proposed model successfully detected the changeover from lockdown level 4 to level 3. This means that our model was successful in capturing some of the interventions (in this case lockdowns) that were imposed by the South African government from February 15, 2020 to June 19, 2020.

In this paper, an approach to inferring the causal effect of COVID-19 interventions has been proposed. The approach uses a hybrid model that incorporates an LSTM autoencoder and a kernel quantile estimator to detect change-points that are then used to infer the causal effects of the COVID-19 interventions.

We implemented our model to detect change points using time series data on population mobility trends before and after an intervention. We used BSTSM which are implemented in the CausalImpact R package (Brodersen et al., 2015) to predict the counterfactual. The causal effect was estimated as the difference between the observed population mobility (before the intervention) and the population mobility that would have been observed had the intervention not occurred (counterfactual).

The lockdown imposed by the South African government on 27 March 2020 caused a significant decrease in activities in all categorised places, as shown in Table 4.4. These findings about the causal effects of the lockdown add to emerging evidence that interventions such as lockdowns significantly reduce mobility (Aloi et al., 2020; Bonaccorsi et al., 2020; Nian et al., 2020). These findings suggest that the causal effects of interventions on population mobility should be carefully considered before taking measures that can severely affect people's livelihoods and the economy. For example, Bonaccorsi et al. (2020) found out that lockdowns disproportionately affect the poor in a country. Atalan (2020) states that the measures taken by countries against the spread of COVID-19 often bring along unprecedented economic hardships. The change from lockdown level 5 to lockdown levels 4 and 3 caused a significant increase in activities at transit stations, grocery and pharmacy, retail and recreation, and workplaces, except for residential places and parks, which did not show significant changes during the transitions, as shown in Tables 4.5 and 4.6. The parks remained closed during lockdown level 4 and only a limited number of open access national parks could open under lockdown level 3. For residential places, the insignificant change in population mobility is because people spend equally more time at home even on workdays.

Making inferences about the effect of COVID-19 interventions is a crucial process that must be done in a timely manner. This is because understanding the effects of such measures can inform policymakers to make the right decisions. If policymakers think that imposing interventions has little effect, they may be faced with a situation where infections may rise again (Goodman-Bacon and Marcus, 2020). On the other hand, if policymakers believe that the interventions they impose can significantly slow down the spread of COVID-19, then the interventions can be maintained for longer periods. However, this damages economic and social recovery, and it is vital to strike a balance between the

potential positive effects of population mobility restrictions on public health and the potential negative social and economic impacts.

4.6.1 Limitations

A possible limitation of our proposed approach is that it was not evaluated in high-dimensional settings (the so-called curse of dimensionality). Therefore, we were unable to determine its accuracy in high-dimensional settings. Another limitation is that the ACAPS COVID-19 Government Measures data set (ACAPS., 2020) used in this paper only contained a description of the measures taken by the South African government from March 10, 2020, to July 7, 2020. This means that our analysis only covered the three lockdown measures (levels 5, 4 and 3) as they fall within the period of March 10, 2020, to July 7, 2020. Google COVID-19 community mobility reports do not provide the actual number of people and duration of stay values, as well as the median values. They only show how the number of people or duration of stay has changed relative to the median, which is a limitation.

For future work, we would like to evaluate our method in (1) high-dimensional settings, (2) detecting multiple change-points in multivariate time series or genomic sequences, (3) identifying possible mutations in SARS CoV-2 genomic sequences and evaluate the causal effect of the identified mutations, and (4) finding the relationship between population mobility and the rate of transmission of the virus.

4.7 Conclusions

We have used a data-driven counterfactual approach to evaluate interventions that guide governments in controlling the spread of COVID-19. The paper has made two very important contributions. Firstly, we used a deep learning approach coupled with a kernel quantile estimator to successfully detect change-points in time series data. Secondly, we performed a careful causal analysis to learn about the effects of different government interventions. The findings show that the complete lockdown imposed on 27 March 2020 to contain the spread of COVID-19 affected the mobility of the population and significantly reduced economic activities in transit stations, grocery and pharmacy, retail and recreation, workplaces and parks. The findings show that people generally

stayed home during the lockdown. Currently, there is no study available in South Africa that incorporates change-point analysis of population mobility trends and causal inference to quantify the effects of an intervention such as a complete lockdown on population movements.

CHAPTER 5

Credibility of Causal Estimates from Regression Discontinuity Designs with Multiple Assignment Variables

---

ALBERT WHATA [1,*] and CHARLES CHIMEDZA [2]

[1] School of Natural and Applied Sciences, Sol Plaatje University

[2] School of Statistics and Actuarial Science, University of the Witwatersrand

Journal paper under review: *Stats — An Open Access Journal from MDPI*:

Statement of Contributions of Joint Authorship

Albert Whata **(Candidate)**

Conducted the research, writing and compilation of manuscript);

Charles Chimedza **(Supervisor)**

Supervised, edited and coauthor of the manuscript.

This Chapter is an exact copy of the journal paper referred to above.

ABSTRACT

In this paper, we determine treatment effects when treatment assignment is based on two or more cut-off points of covariates rather than on one cut-off point of one assignment variable, using methods that are referred to as multivariate regression discontinuity designs (MRDD). One major finding of this paper is the discovery of new evidence that both matric points and household income have a huge impact on the probability of eligibility for funding from the National Student Financial Aid Scheme (NSFAS) to study for a bachelor's degree program at universities in South Africa. This evidence will inform policymakers and educational practitioners on the effects of matric points and household income on eligibility for NSFAS funding. The availability of the NSFAS grant has a huge impact on students' decisions to attend university or seek other opportunities elsewhere. Using the analytical results of the frontier MRDD, barely scoring matric points greater than or equal to 25 points compared to scoring matric points less than 25 for students whose household income is less than R350 000 ($\approx$ US\$25 00), increases the probability of eligibility for NSFAS funding by a significant percentage point of 3.75 (p-value $= 0.0001 < 0.05$) percentage points. Therefore, we have shown that the frontier MRDD can be used to determine the causal effects of barely meeting the requirements of one assignment variable, among the subjects that either meet or fail to meet the requirements of the other assignment variable.

5.1 Introduction

Multivariate regression discontinuity designs (MRDD) pose challenges that are distinct from those identified in traditional RDD (Wong et al., 2013). Traditional RDD studies focus on units that are assigned to a treatment based on a single cut-off point and a single continuous assignment variable (Lee and Lemieux, 2010). In reality, units are usually assigned to a treatment based on more than one continuous assignment variable (Cheng, 2016). Thus, treatment effects may be estimated across a multi-dimensional cut-off frontier, as opposed to a single point on the assignment variable, using methods referred to as multivariate regression discontinuity designs (MRDD) (Wong et al., 2013; Papay et al., 2011; Cheng, 2016). For example, using the frontier approach, Wong et al. (2013) showed that the MRDD treatment effect estimate, τ_{MRDD} may be decomposed into a weighted average of two univariate RDD effects, τ_{x_1} at the X_1-cut-off and τ_{x_2} at the X_2 -cut-off. The term *frontier* means that the average treatment effect estimates for MRDD are only for subpopulations of units located at the cut-off frontier as opposed to the average treatment effect for the overall population under study. Like in the standard RDD, the causal estimates obtained in MRDD have limited external validity, as the causal estimates are only identified for observations in the immediate vicinity of the cut-off scores (Papay et al., 2011). This paper studies the credibility of estimates of the MRDD. It is important that after estimating τ_{MRD}, a researcher checks the plausibility of the assumptions of the estimates of MRDD. With more credible estimates, inference about causality can reduce the reliance of causal estimates on the following modelling assumptions (Reardon and Robinson, 2012):

1. the cut-off scores determining treatment assignment are exogenously set;
2. potential outcomes are continuous functions of the assignment scores at the cut-off scores and
3. the functional form of the model is correctly specified.

Reardon and Robinson (2012) suggest extending the assumption checking for the RDD context to the MRDD. Assumptions will be assessed as they apply to the frontier regression discontinuity using *supplementary analysis* techniques (Athey and Imbens, 2017). Primary analyses in MRDD focus on estimating

τ_{MRD}. In contrast, supplementary analyses seek to shed more light on τ_{MRD} from the primary analyses. These supplementary analyses use the fact that assumptions behind the identification strategy often have implications for the data beyond those used in the primary analyses (Athey and Imbens, 2017). There are basically four approaches that can be employed for carrying out supplementary analyses, and these are: the McCrary test (McCrary, 2008), placebo analysis, robustness and sensitivity, checking for discontinuity of the assignment variable at the cut-off point.

This paper uses two assignment variables to study the effects of an educational funding intervention in South Africa, the National Student Financial Aid Scheme (NSFAS). The eligibility criteria for NSFAS funding are based on a student's family annual total household income and the aggregate points they score on their matriculation certificate. In South Africa, a matriculation certificate is a school-leaving diploma or national senior certificate (NSC) that is awarded after completing Grade 12. The total number of scores achieved on seven subjects on the matriculation certificate are referred to in this paper as the matric points. The role that NSFAS funding plays in access to post-secondary education for students from disadvantaged backgrounds in South Africa cannot be ignored. Therefore, one of the vital objectives of this research is to evaluate the causal effect of household income and matric points on the probability of eligibility for NSFAS funding. To the best of our knowledge, no such studies have been carried out to date in South Africa using a multivariate regression discontinuity design to quantify the causal effect of household income and matric points on eligibility for NSFAS funding. The other goal of this paper is to provide analyses beyond the primary analyses that yield causal estimates by assessing the credibility of causal estimates through supplementary analyses.

5.2 Literature Review

5.2.1 Multivariate Regression Discontinuity Design

The basic setup of RDD is that we are interested in the causal effect of a binary treatment or program denoted by W_i, in the presence of an exogenous variable (forcing variable) denoted by X_i. There are different RDD methods proposed in the literature that allow us to estimate average treatment effects.

s Athey and Imbens (2017), Imbens and Wooldridge (2009), Lee and Lemieux (2010), Wong et al. (2013), and Papay et al. (2011) provide detailed overviews of these methods. The RDD is framed in the context of the potential outcomes framework i.e., for a given unit i, there exist two potential outcomes, $Y_i(1)$ and $Y_i(0)$, and the causal effect is simply the difference, $Y_i(1)$-$Y_i(0)$. The fundamental problem of causal inference states that we cannot observe $Y_i(1)$ and $Y_i(0)$ at the same time. Because of this problem we usually focus on the average effects of the treatment, that is, averages of $Y_i(1)$-$Y_i(0)$ over all subpopulations rather than on unit-level effects (Lee and Lemieux, 2010). Authors Lee and Lemieux (2010) gave a compilation of over sixty studies, showing where RDD designs have been applied in many different contexts. Matsudaira (2008) estimates the effect of a mandatory summer school program assigned to students who fail to score higher than a preset cut-off in both math and reading exams. Keele and Titiunik (2015) developed a special kind of regression discontinuity design in which the assignment variable is geographic. Their approach, the geographic regression discontinuity (GRD) design, was similar to a standard RD but with two running variables.

The multivariate regression discontinuity designs (MRDD) present opportunities for obtaining unbiased estimates of treatment effects using the same thinking as the single assignment variable RD designs. The MRDD is an extension of the traditional RDD, except that the treatment effects are estimated for multiple cut-offs, as opposed to a single cut-off point. Studies have been carried out with multivariate assignment variables and cut-off points for treatment assignment. For example, Papay et al. (2011) gave an example of teachers who received a bonus for improving student scores in both Mathematics and English Language Arts (ELA). The authors point out that in some cases students must pass externally defined standards in several subjects to avoid summer school or to graduate from high school. Multivariate regression discontinuity designs (MRDD) have also been used in other disciplines, such as politics (Cattaneo et al., 2016). There are multiple strategies for estimating MRDD and these strategies present multiple possible estimands (τ_{MRDD}). Reardon and Robinson (2012) present five strategies for estimating treatment effects in MRDD, namely: binding-score RD, distance-based RD, frontier RD, the response surface RD, and fuzzy frontier RD. The authors mention that these four methods have their own advantages and disadvantages that depend

on the structure of the data with regard to the correlation between the two assignment variables as well as the locations of the respective cut-points. Reardon and Robinson (2012) indicate that fuzzy IV methods have limited practical applications, as they tend to yield imprecise estimates. Also, the response surface RD is sensitive to the functional form mis-specification and this could be a problem when it is implemented using local linear regression, which requires full functional form specification.

We employ the frontier RD as recommended by Reardon and Robinson (2012) to estimate the probability of eligibility for NSFAS funding using matric points and household income as assignment variables. The frontier approach is deemed to be straightforward to implement, as the modelling assumptions mentioned earlier are easily assessed by subsetting the data by frontier (Reardon and Robinson, 2012). In addition, because the frontier approach reduces the MRDD to at least one single-rating RDD, they are easily implemented using traditional RDD methods. We implement the frontier approach to obtain the primary analyses estimates. Thus, we adopt the approaches proposed by Papay et al. (2011) and Reardon and Robinson (2012) that use two assignment variables to assign individuals to a range of different treatment conditions. According to Papay et al. (2011), if there are J forcing variables such that $W_{ji} = \mathbb{1}(X_{ji} \geq c)$, $\forall\ j = 1,\ldots,J$, then there will be 2^J treatment conditions. For any combination of these 2^J treatment effects, the parameters of interest become the left and right side limits on either side of the cut-off. MRDD with two assignment variables defines four different treatment conditions.

For two assignment variables, X_1 and X_2 and respective cut-offs, c_1 and c_2, the treatment conditions W_{1i} and W_{2i} are defined as follows:

$$W_{1i} = \begin{cases} 1, & \text{if } X_{1i} \geq c_1 \\ 0, & X_{1i} < c_1 \end{cases} \quad \text{and} \quad W_{2i} = \begin{cases} 1, & \text{if } X_{2i} \geq c_2 \\ 0, & X_{2i} < c_2 \end{cases} \tag{5.1}$$

To qualify for NSFAS funding, a student must have a matric certificate with at least 25 matric points, and the total family household income should be at most R350 000. MP = 25 and INC = R350 000 are therefore the cut-off points for matric points and household income, respectively. Thus, as described by Papay et al. (2014), with two assignment variables, there are four treatment conditions that define four distinct regions in a two-dimensional space spanned by MP^c and INC^c, the centered variables of MP and INC

around their respective cut-off points.

1. Treatment 1: If the students score at least 25 matric points and the family income is greater than R350 000 (Region 1) : $W_{1i} = 1$ and $W_{2i} = 0$

2. Treatment 2: If the students score less than 25 matric points and the family income is greater than R350 000 (Region 2): $W_{1i} = 0$ and $W_{2i} = 0$

3. Treatment 3: If the students score less than 25 matric points and the family income is at most R350 000 (Region 3): $W_{1i} = 0$ and $W_{2i} = 1$

4. Treatment 4: If the students score at least 25 matric points and the family income is at most R350 000 (Region 4): $W_{1i} = 1$ and $W_{2i} = 1$

The parameters that we estimate then become the population conditional mean probabilities of being eligible for NSFAS funding for students in each treatment condition at the appropriate cut-off point. For example, the causal effect of being eligible for NSFAS funding instead of not being eligible for NSFAS funding, for a student scoring on the income (INC) cut-off is then the difference between (Lee and Lemieux, 2010):

$$u_l(c) = \lim_{x \downarrow 0^-} E[NSF_i | MP^C = x, INC^C = 0]$$

and

$$u_r(c) = \lim_{x \uparrow 0^+} E[NSF_i | MP^C = x, INC^C = 0] \tag{5.2}$$

5.2.2 Multivariate Assignment Variables: Estimation Strategies

Equation 5.2 gives the basic analytic strategy for estimating treatment effects using single or multivariate assignment variables. Observed data is used to estimate the limits of the average potential outcomes at the boundaries of two treatment assignment regions, and then take the difference of the estimated limits. According to Reardon and Robinson (2012), these limits must be estimated at the boundary of the observed data for each treatment condition and fit a regression model of the following form (assuming a sharp regression discontinuity).

$$Y_i = f(X_{1i}, X_{2i}, ..., X_{ji}, W_i) + \epsilon_i, \tag{5.3}$$

where $X_{1i},\ X_{2i}, ..., X_{ji} \in D \subset S$ and D is the domain of observations used to estimate the model, S is the domain of observations in the entire sample, and W_i is a zero / one treatment-assignment indicator.
For MRDD with two assignment variables, we have the following:

$$Y_i = f(X_{1i}, X_{2i}, W_i) + \epsilon_i, \tag{5.4}$$

where $X_{1i},\ X_{2i} \in D \subset S$ and D is the domain of observations used to estimate the model, S is the domain of observations in the full sample, and W_i is a zero/one treatment-assignment indicator (Porter et al., 2017). The authors point out that estimators of the MRDD differ in terms of:

(i) the specification of the function f;

(ii) the domain (D) of observations used in estimating the model.

5.3 Materials and Methods

5.3.1 Data

We focus on estimating the impact that matric points and household income have on the chance that a student qualifies for the NSFAS grant to support his or her bachelor's degree studies using simulated data for household income (INC) and matric points (MP). Thus, our assignment variables are MP and INC. The simulated income data will be generated using the log-normal distribution (Van der Berg, 2011; Cheng, 2016). To simulate income data for South Africa that follow a log-normal distribution, we estimate the mean and standard deviation of total household income from Table 5.1 of the average household incomes by population group in South Africa (`http://www.statssa.gov.za/publications/P0310/P03102014.pdf`).
In order to simulate the MP data we use a scaled, shifted, and truncated beta distribution proposed by Kane (2003) and Cheng (2016). Using Table 5.2, we consider seven matric subjects and generate matric points that range from 0 to 49. If a student does not achieve any of the subject pass levels that are shown in Table 5.2 in seven subjects, then the students will score a total of zero matric points. On the other hand, if a student achieves a subject pass level equal to seven in seven subjects, then the student will have a total of 49 matric points.

Table 5.1: Average income distribution (Source: *STATSSA: Living Conditions Survey 2.*

Population group of household	**Average income**	**% Households**	**Number of Households**
Black African	92983	80.41	18799.858
Coloured	172765	7.23	1690.374
Indian/Asian	271621	2.31	540.078
White	444446	10.04	2347.352
Total	**138168**		**23377.662**

Table 5.2: Subject Passes Level System.

Level	**Final Mark%**	**Achievement**
7	80-100%	Outstanding
6	70-79%	Meritorius
5	60-69%	Substantial
4	50-59%	Moderate
3	40-49%	Adequate
2	30-39%	Elementary
1	0-29%	Not Achieved-Fail

Different combinations of the scaled, shifted, and truncated beta distribution parameters α and β were employed in simulating the matric points data. We chose $\alpha = 5$, $\beta = 13$ as the preferred parameters for simulating matric points as they give us the density of matric points slightly positively skewed to the right as shown in Figure 5.1.

After simulating the income and matric points data, we use the data generation process in Equation 5.5, to generate estimates of the probabilities of eligibility for NSFAS funding $P(NSF_i = 1|INC^c, MP^c)$:

$$P(NSF_i = 1|INC^c, MP^c) = \beta_0 + \beta_1 INC^c + \beta_2 MP^c + INC^c * MP^c + \epsilon_i \quad (5.5)$$

We used simulated data because at the time of writing this article, consolidated data showing household income and matric points for each student were not available from NSFAS, and therefore we could not directly apply our approach to the income and matric points data from South African students in the real world. The simulated data for household income have a connection to the "real world" data because it is based on household incomes that were compiled per household population group by Statistics South Africa (Stats SA). In addition, the matric points were simulated using a scaled, shifted, and truncated beta

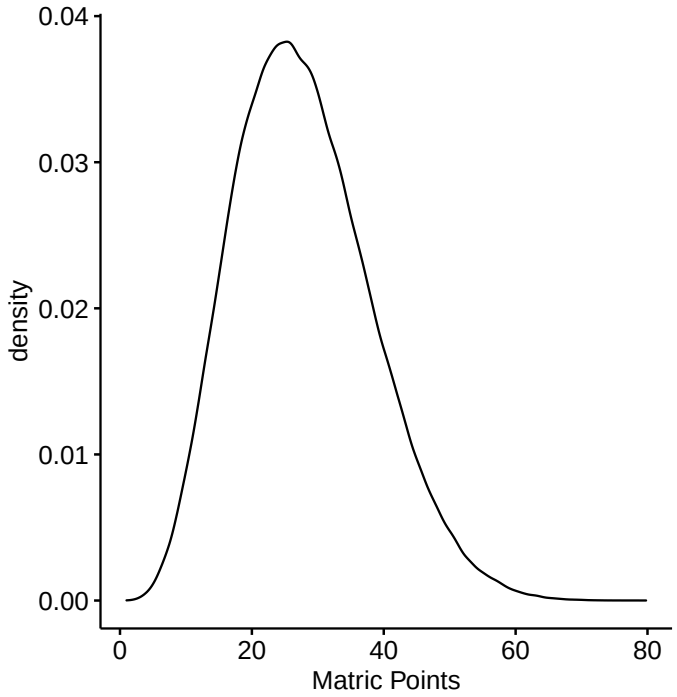


Figure 5.1: Density of Matric Points.

distribution that were employed to simulate grade point averages (GPAs) by authors such as Cheng (2016) and Kane (2003). In addition, Arthurs et al. (2019) mention that the beta distribution can also be used to simulate data on grades. We have used a simulation-based approach because it can provide us with different data-generating mechanisms so that we can evaluate different scenarios such as varying the sample size of simulated data sets or the variability of the assignment variables. Because of the unavailability of consolidated real-world household income data and matric points data, we used the graduate admission data (Acharya et al., 2019), to demonstrate the applicability of the frontier MRDD to a real-world data set as well as conduct supplementary analyses. The graduate admission data set describes the probability of admission to graduate master's programs in science and technology in the USA for Indian students.

5.3.2 Estimating Causal Effects Using the Frontier Regression Discontinuity Design(FRDD)

The FRDD approach analyses samples of the data based on status (i.e., above or below the cut-off score) on all but one of the assignment variables and then

models the discontinuity along the remaining assignment variable using the single assignment variable RDD methods (Reardon and Robinson, 2012). Focusing on the MP^c frontier method entails using sample points where $INC^c \geq 0$. Because the assignment variable is centered on its cut-point value, sample points where $INC^c \geq 0$ are used. For example, when estimating Treatment 1 versus Treatment 2 in Figure 5.2, we limit the sample only to individuals with $INC^c \geq 0$.

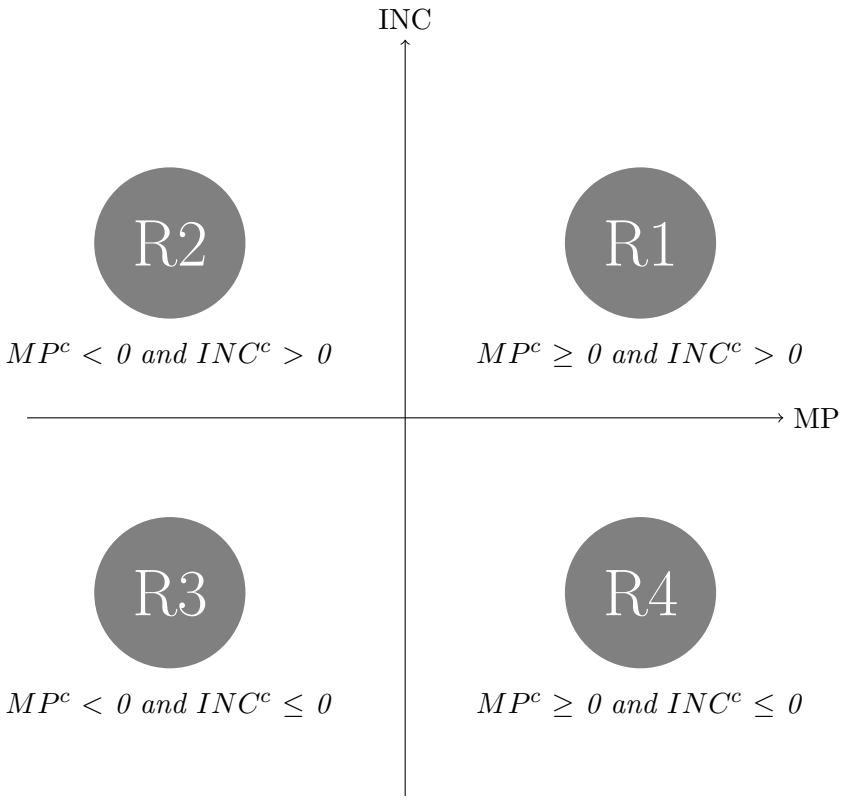


Figure 5.2: Treatment Regions R1 to R4.

Thereafter, we choose a sub-sample from the samples that lie within the optimal bandwidth on the left or right of MP^c. This sub-sample will be used in fitting linear probability models of the form:

$$P(NSF_i = 1|MP^c, T_{mp_i}) = \beta_0 + \beta_1 T_{mp_i} + \beta_2 MP_i^c + \beta_3 MP_i^c * T_{mp_i} + \epsilon_i, \quad (5.6)$$

or

$$P(NSF_i = 1|MP^c, T_{mp_i}, INC^c) = \beta_0 + \beta_1 T_{mp_i} + \beta_2 MP_i^c + \beta_3 MP_i^c * T_{mp_i} + \beta_4 INC_i^c + \epsilon_i, \quad (5.7)$$

Where $P(NSF_i = 1|MP^c, T_{mp_i})$ is the probability of eligibility for NSFAS

funding given MP^c and the treatment assignment variable for the matric points, T_{mp_i}. Equation 5.7 includes the covariate INC^c to improve the treatment effect, $\widehat{\beta}_1$'s precision. The equations are also adapted and expressed as functions of INC^c, T_{inc_i}, and MP^c as a covariate, to evaluate the effect of the income variable on the chance of eligibility for NSFAS funding. We use the linear probability model specification, rather than *logistic* or *probit*, to model Equations 5.6 and 5.7. According to Angrist and Pischke (2008) and Papay et al. (2014), the linear probability specification provides consistent and unbiased estimates of the fundamental trends for samples. In addition, the interpretation of the linear probability model becomes enormously simple. Also, Papay et al. (2014), compared the linear probability model and logistic regression and found that within a narrow and optimal bandwidth these two models produce identical results. Von Hippel (2015) suggested that if the probabilities that are being modelled are extreme, that is, close to 0 or 1, then a researcher would probably have to use logistic regression. However, if the probabilities that are being modelled are more moderate, i.e., between .20 and .80, or a little greater than 0.8, then the linear and logistic models fit about equally well, and the linear model is preferred over the logistic regression as it is easy to interpret. In addition, Angrist and Pischke (2008); Papay et al. (2014) state that the linear probability model becomes even more credible when it is applied within a narrow bandwidth using local linear regression analysis. In addition, the regression parameters are easy to interpret in terms of population differences in the probability of eligibility for NSFAS funding per unit difference in either matric points or household income. Deke (2014) highlights that the linear probability model yields treatment estimates that are just as accurate as those estimated by logistic regression. As we estimate the effect of matric points and household income on the probability of eligibility for NSFAS funding, the parameter of interest is the coefficient (β_1) of the treatment variable(s) and not the coefficients of the assignment variables. This makes the linear probability model an appropriate analytic procedure for estimating the effect of matric points and household income since the treatment status is a binary variable (Deke, 2014). Thus, the functional form concerns about the linear probability model may not necessarily apply because all that is required is to estimate the treatment effects as opposed to estimating the effect of the continuous assignment variables.

5.4 Experiments

To implement the frontier multivariate regression discontinuity design, we consider three variables, namely, the matric points (MP), household income (INC), and NSFAS funding (NSF). Students apply for NSFAS funding in grade 12 before their matric results are out and they do not know in advance whether or not they are going to achieve matric points that meet the requirement to study for a bachelor's degree. This makes MP suitable as an assignment variable as it cannot be manipulated precisely. MPs around the threshold then become as good as random. The other variable is the total household income (INC), which is a measure of the total income of members of the household. The variable NSF indicates the chance that a student is eligible for NSFAS funding to study for a bachelor's degree within one year after matriculation. Data is simulated that contain MP scores and INC values with cut-off points of 25 and R350 000 ($\approx$ US$25 000) respectively. We create continuous assignment variables (predictors) by centering the normalised MP and INC scores on the values of their respective cut-off points. The centered continuous predictors are labelled MP^c and INC^c. Thus, $MP^c = 0$ and $INC^c = 0$, indicate that the student achieved the minimum requirements for qualifying for NSFAS funding. Following the approach by Papay et al. (2014), we create dichotomous versions of the same predictors (MP and INC) labelled, T_{mp} and T_{inc} which are the treatment variables that indicate whether a student met the minimum passing standard (MP = 25 points) required to study for a bachelor's degree and also met the maximum income threshold (INC = R350 000) for NSFAS funding. A binary outcome variable Y_i was generated such that $Y_i = 1$ if $MP_i^c >= 0$ and $INC_i^c <= 0$, that is, a student receives NSFAS funding and 0 otherwise, that is, a student does not receive NSFAS funding. The probabilities (between 0 and 1) of eligibility for NSFAS funding are then estimated by fitting a logistic regression to Equation 5.8 adapted from Papay et al. (2011, 2014), thereby giving the outcome variable as a probability;

$$P(NSF_i = 1|MP^c, INC^c) = \hat{\gamma_0} + \hat{\gamma_1} MP_i^c + \hat{\gamma_2} INC_i^c + \hat{\gamma_3}(MP_i^c \times INC_i^c) + \epsilon_i \tag{5.8}$$

Under normal conditions, the full functional forms of the data generation models are usually not known (Reardon and Robinson, 2012). Therefore, choosing an estimation method and the best way to implement it could be challenging.

Thus, we took a systematic approach where we have relied on programmed algorithms to select the bandwidth that ensures that the chosen functional form approximates our simulated data well. Equations 5.6 and 5.7 were used to specify the linear functional forms using an optimal bandwidth that minimises the mean squared error of $\widehat{\beta}_1$. This bandwidth is chosen by using a nonparametric density estimation algorithm (Cattaneo et al., 2020). The causal effects are then estimated by fitting a linear probability model to Equations 5.6 and 5.7, the data that lie within the chosen optimal bandwidths and then derive the estimates of $\widehat{\beta}_1$. Equation 5.6, specifies the outcomes as a linear function of, say, MP^c, the treatment variable, T_{mp}, an interaction of MP^c and T_{mp}. Equation 5.7 adds INC^c as a baseline covariate. In addition, we investigate whether adding $INC^c \geq 0$ as a covariate improves the precision of the estimates of $\widehat{\beta}_1$.

Simulations were conducted based on the values of income, matric points, and Equations 5.6 and 5.7. MP^c and INC^c were generated as continuous variables. We introduced different values of σ ($\sigma = 0$, 0.05, 0.1 and 0.15) to control for the amount of variation in the assignment variables. The simulated data was initially analysed without adding more variation to the simulated assignment variables, this represented $\sigma = 0$. The variation in the assignment variables was gradually increased in steps of 0.05 from 0 to 0.15, and the models were rerun to give the new estimates of the parameters of interest. We will investigate whether or not increasing the variation in the assignment variables affects the treatment assignment mechanism, and thus, the treatment effect estimates. Also, we considered different samples of sizes 5 000, 10 000, and 20 000 to determine whether or not increasing the sample size affects the estimated treatment effects. 1 000 simulations were performed for each model, by varying the sample sizes and the variability σ that controls the amount of variation in the assignment variables. Overall, the simulation scenarios were composed of three different levels of sample sizes, four error variances, and four treatment regions (Figure 5.2) for each of Equations 5.6 and 5.7, giving a total of 96 scenarios. For each scenario, we generated 1 000 simulated data sets and used Equations 5.6 and 5.7 to estimate the coefficient β_1, which gives the estimate of the treatment effect. Because, the true relationship between a binary outcome and continuous explanatory variables is fundamentally not linear, the functional form of the linear probability model is generally not

correctly specified. We will show through supplementary analyses that within a narrow and optimal bandwidth, the linear specification model is still credible.

5.5 Results and Analysis

5.5.1 Estimation of the Causal Effects

The results obtained by fitting the simulated data in each treatment region (Figure 5.2) to Equation 5.6 are presented in Tables 5.3 and 5.4. Table 5.3 shows that for treatment regions R3 vs R4 the treatment coefficients β_1's are statistically significant (p-value $= 0.0119 < 0.05$) when $N = 20\ 000$ for $\sigma = 0$. Also, the coefficients are statistically significant (p-value $= 0.0349 < 0.05$) when $N = 20\ 000$ for $\sigma = 0.05$. These findings suggest that Equation 5.6 requires a larger sample size for it to start producing significant treatment effects. However, using Equation 5.7 that includes INC^c as a covariate yields significant treatment effects for all levels of σ and for all different sample sizes, as shown in Table 5.5. Therefore, we will report the results shown in Table 5.5 that are based on the statistically significant estimates obtained by fitting the data to Equation 5.7 (with INC^c as a covariate). Table 5.5 shows that when $\sigma = 0$, the treatment effects (β_1's) for R3 vs R4, are comparable and statistically significant for the different sample sizes. These results show that when $\sigma = 0$ and $N = 5\ 000$, 10 000 or 20 000, scoring matric points that are greater than or equal to 25 compared to scoring matric points that are less than 25, increases the probability of eligibility for NSFAS funding, i.e., $P(NSF_i = 1)$ by 3.75, 3.74 and 3.71 percentage points respectively for a unit increase in matric points, for students whose households income is less than or equal to R350 000 ($\approx$ US$25 000).

The results show strong evidence that scoring matric points that are greater than or equal to 25 and an income less than R350 000 ($\approx$ US$25 000) significantly increases the chance of eligibility for NSFAS funding. This makes achieving matric points greater than 25 and household income less than R350 000 important variables to be considered when awarding NSFAS funding. The estimated treatment effects decrease when the variability in the two assignment variables, MP^c and INC^c, is varied from $\sigma = 0$ to $\sigma = 0.15$ in steps of 0.05, while keeping the cut-offs constant as shown in Table 5.5 for R3 vs R4. Furthermore, the treatment effects are all still statistically significant at the 5%

level of significance. These results suggest that the causal estimates obtained by using the frontier approach may be sensitive to the level of variation in the assignment variables, because increasing the variation in the assignment variables may cause the observations to move away from the cut-off points, thereby decreasing the treatment effects.

As shown in Table 5.5 for R1 vs R2, when $\sigma = 0$ and $N = 10\,000$, scoring matric points that are greater than or equal to 25 compared to scoring matric points that are less than 25, decreases the probability of eligibility for NSFAS funding, i.e., $P(NSF_i = 1)$ by a significant 7.96 (p-value $= 0.0111 < 0.05$) percentage points for a unit increase in matric points for students whose household income is greater than or equal to R350 000 ($\approx$ US$25 000). These students who just meet the matric point threshold but with a household income that is just greater than R350 000 ($\approx$ US$25 000) do not receive NSFAS funding, and yet they are not different from those whose income is just below R350 000 ($\approx$ US$25 000). These students are referred to as the "missing middle" (Garrod and Wildschut, 2021). The authors define the "missing middle" as the students who come from households whose incomes are between R350 000 and R600 000. These students do not qualify for NSFAS funding, but at the same time they cannot afford to pay for their higher education.

Tables 5.4 and 5.6, explore the effects of having a household income greater than R350 000 compared to having an income less than or equal to R350 000 for students who score matric points greater than 25 or less than 25. These results show that all the β_1's are small and not statistically significant at 5% level of significance. The implications of these results are that when awarding NSFAS funding, one must first look at whether or not a student has met the matric points threshold and then consider the household income. Considering the matric points first, makes it easier to quantify the number of students that have qualified for university entry. Consequently, NSFAS will then be in a position to quantify those that automatically qualify for NSFAS funding and also, quantify the "missing middle".

Table 5.3: **Top**: Simulation results for fitting Equation 5.6 to compare R3 vs R4, which represents $MP^c < 0$ vs $MP^c \geq 0$ for $INC^c < 0$, and cut-off $c = 0$. **bottom**: Simulation results for fitting Equation 5.6 to compare R1 vs R2, which represents $MP^c < 0$ vs $MP^c \geq$ for $INC^c > 0$ and cut-off, $c = 0$.

R3 vs R4	N	h_l	h_r	β_1	$s.e$	p-value
	5 000	-0.1307	0.1248	0.0388	0.0196	0.1607
$\sigma = 0.00$	10 000	-0.1307	0.1248	0.0371	0.0139	0.0718
	20 000	-0.1307	0.1248	0.0371	0.0098	0.0119
	5 000	-0.1307	0.1248	0.0293	0.0186	0.2276
$\sigma = 0.05$	10 000	-0.1307	0.1248	0.0295	0.0131	0.1157
	20 000	-0.1307	0.1248	0.0293	0.0093	0.0349
	5 000	-0.1307	0.1248	0.0169	0.0165	0.3484
$\sigma = 0.10$	10 000	-0.1307	0.1248	0.0164	0.0117	0.2728
	20 000	-0.1307	0.1248	0.0161	0.0082	0.1631
	5 000	-0.1307	0.1248	0.0086	0.0146	0.4283
$\sigma = 0.15$	10 000	-0.1307	0.1248	0.0080	0.0103	0.4161
	20 000	-0.1307	0.1248	0.0082	0.0073	0.3301
R1 vs R2						
	5 000	-0.1922	0.2859	-0.0763	0.0407	0.1061
$\sigma = 0.00$	10 000	-0.1922	0.2859	-0.0790	0.0287	0.0171
	20 000	-0.1922	0.2859	-0.0786	0.0203	0.0008
	5 000	-0.1922	0.2859	-0.0680	0.0393	0.1388
$\sigma = 0.05$	10 000	-0.1922	0.2859	-0.0689	0.0276	0.0334
	20 000	-0.1922	0.2859	-0.0685	0.0194	0.0023
	5 000	-0.1922	0.2859	-0.0474	0.0373	0.2668
$\sigma = 0.10$	10 000	-0.1922	0.2859	-0.0475	0.0258	0.1139
	20 000	-0.1922	0.2859	-0.0468	0.0180	0.0274
	5 000	-0.1922	0.2859	-0.0272	0.0346	0.4550
$\sigma = 0.15$	10 000	-0.1922	0.2859	-0.0275	0.0243	0.3304
	20 000	-0.1922	0.2859	-0.0278	0.0171	0.1718

5.5.2 Supplementary Analyses

The following supplementary analyses are considered in this study, and they are based on the simulated data when $\sigma = 0$ and $N = 10\ 000$.

Table 5.4: **Top**: Simulation results for fitting Equation 5.6 to compare R2 vs R3, which represents $INC^c \leq 0$ vs $INC^c > 0$ for $MP^c \leq 0$, and cut-off $c = 0$. **bottom**: Simulation results for fitting Equation 5.6 to compare R1 vs R4, which represents $INC^c \leq 0$ vs $INC^c > 0$ for $MP^c > 0$, and cut-off $c = 0$.

R2 vs R3	$\boldsymbol{N}$	h_l	h_r	β_1	*s.e*	p-value
	5 000	-0.5097	0.5114	-0.0003	0.0040	0.6029
$\sigma = 0.00$	10 000	-0.5097	0.5114	-0.0003	0.0028	0.6055
	20 000	-0.5097	0.5114	-0.0004	0.0019	0.5977
	5 000	-0.5097	0.5114	-0.0004	0.0055	0.5924
$\sigma = 0.05$	10 000	-0.5097	0.5114	-0.0004	0.0038	0.5874
	20 000	-0.5097	0.5114	-0.0004	0.0027	0.5802
	5 000	-0.5097	0.5114	-0.0007	0.0098	0.5685
$\sigma = 0.10$	10 000	-0.5097	0.5114	-0.0004	0.0068	0.5667
	20 000	-0.5097	0.5114	-0.0004	0.0048	0.5695
	5 000	-0.5097	0.5114	-0.0002	0.0156	0.5554
$\sigma = 0.15$	10 000	-0.5097	0.5114	-0.0003	0.0108	0.5637
	20 000	-0.5097	0.5114	-0.0003	0.0076	0.5615
R1 vs R4						
	5 000	-0.4200	0.2970	-0.0014	0.1425	0.5046
$\sigma = 0.00$	10 000	-0.4200	0.2970	-0.0017	0.0996	0.4979
	20 000	-0.4200	0.2970	-0.0007	0.0699	0.4798
	5 000	-0.4200	0.2970	-0.0033	0.1365	0.4917
$\sigma = 0.05$	10 000	-0.4200	0.2970	-0.0008	0.0960	0.4957
	20 000	-0.4200	0.2970	0.0007	0.0675	0.4890
	5 000	-0.4200	0.2970	0.0041	0.1261	0.4851
$\sigma = 0.10$	10 000	-0.4200	0.2970	-0.0014	0.0884	0.4938
	20 000	-0.4200	0.2970	0.0044	0.0620	0.4953
	5 000	-0.4200	0.2970	0.0019	0.1134	0.4928
$\sigma = 0.15$	10 000	-0.4200	0.2970	-0.0005	0.0792	0.4970
	20 000	-0.4200	0.2970	0.0003	0.0558	0.4791

5.5.2.1 Checking for continuity of the conditional expectation of exogenous variables around the cut-off/threshold value

Graphical analysis is an integral part of any RDD design. It is assumed that the treatment effect or causal effect of interest is measured by the value of the

Table 5.5: **Top**: Simulation results for fitting Equation 5.7 to compare R3 vs R4, which represents $MP^c \leq 0$ vs $MP^c > 0$ for $INC^c <= 0$, and cut-off $c = 0$. **bottom**: Simulation results for fitting Equation 5.7 to compare R1 vs R2, which represents $MP^c \leq 0$ vs $MP^c > 0$ for $INC^c > 0$, and cut-off $c = 0$.

R3 vs R4	N	h_l	h_r	β_1	$s.e$	p-value
	5 000	-0.1307	0.1248	0.0375	0.0067	0.0001
σ= 0.00	10 000	-0.1307	0.1248	0.0374	0.0047	0.0000
	20 000	-0.1307	0.1248	0.0371	0.0033	0.0000
	5 000	-0.1307	0.1248	0.0296	0.0056	0.0002
$\sigma = 0.05$	10 000	-0.1307	0.1248	0.0292	0.0039	0.0000
	20 000	-0.1307	0.1248	0.0291	0.0028	0.0000
	5 000	-0.1307	0.1248	0.0163	0.0037	0.0019
$\sigma = 0.10$	10 000	-0.1307	0.1248	0.0162	0.0026	0.0000
	20 000	-0.1307	0.1248	0.0161	0.0018	0.0000
	5 000	-0.1307	0.1248	0.0080	0.0022	0.0084
$\sigma = 0.15$	10 000	-0.1307	0.1248	0.0080	0.0015	0.0001
	20 000	-0.1307	0.1248	0.0080	0.0011	0.0000
R1 vs R2						
	5 000	-0.1922	0.2859	-0.0778	0.0362	0.0843
$\sigma = 0.00$	10 000	-0.1922	0.2859	-0.0796	0.0256	0.0111
	20 000	-0.1922	0.2859	-0.0782	0.0181	0.0003
	5 000	-0.1922	0.2859	-0.0682	0.0341	0.1078
$\sigma = 0.05$	10 000	-0.1922	0.2859	-0.0692	0.0241	0.0200
	20 000	-0.1922	0.2859	-0.0685	0.0169	0.0012
	5 000	-0.1922	0.2859	-0.0469	0.0298	0.1961
$\sigma = 0.10$	10 000	-0.1922	0.2859	-0.0473	0.0210	0.0675
	20 000	-0.1922	0.2859	-0.0468	0.0147	0.0101
	5 000	-0.1922	0.2859	-0.0273	0.0249	0.3434
$\sigma = 0.15$	10 000	-0.1922	0.2859	-0.0275	0.0176	0.2041
	20 000	-0.1922	0.2859	-0.0278	0.0126	0.0746

discontinuity in the expected outcome at a particular cut-off point. This is a fundamental assumption that postulates that without an intervention, the outcome variable would have been continuous at the cut-off point. This means that any discontinuity in the outcome is credited only to the treatment or

Table 5.6: **Top**: Simulation results for fitting Equation 5.7 to compare R2 vs R3, which represents $INC^c \leq 0$ vs $INC^c > 0$ for $MP^c <= 0$, and cut-off $c = 0$. **bottom**: Simulation results for fitting Equation 5.7 to compare R1 vs R4, which represents $INC^c \leq 0$ vs $INC^c > 0$ for $MP^c > 0$, and cut-off $c = 0$.

R2 vs R3	N	h_l	h_r	β_1	$s.e$	p-value
	5 000	-0.5097	0.5114	-0.0004	0.0031	0.5879
$\sigma = 0.00$	10 000	-0.5097	0.5114	-0.0004	0.0021	0.6032
	20 000	-0.5097	0.5114	-0.0004	0.0015	0.5758
	5 000	-0.5097	0.5114	-0.0005	0.0041	0.5817
$\sigma = 0.05$	10 000	-0.5097	0.5114	-0.0004	0.0029	0.5836
	20 000	-0.5097	0.5114	-0.0004	0.0020	0.5763
	5 000	-0.5097	0.5114	-0.0004	0.0067	0.5835
$\sigma = 0.10$	10 000	-0.5097	0.5114	-0.0005	0.0046	0.5589
	20 000	-0.5097	0.5114	-0.0005	0.0033	0.5558
	5 000	-0.5097	0.5114	-0.0006	0.0093	0.5558
$\sigma = 0.15$	10 000	-0.5097	0.5114	-0.0008	0.0065	0.5368
	20 000	-0.5097	0.5114	-0.0001	0.0045	0.5427
R1 vs R4						
	5 000	-0.4200	0.2970	-0.0014	0.0842	0.5050
$\sigma = 0.00$	10 000	-0.4200	0.2970	-0.0016	0.0591	0.4956
	20 000	-0.4200	0.2970	-0.0007	0.0414	0.5179
	5 000	-0.4200	0.2970	-0.0009	0.0787	0.4877
$\sigma = 0.05$	10 000	-0.4200	0.2970	-0.0003	0.0552	0.5116
	20 000	-0.4200	0.2970	-0.0006	0.0389	0.5114
	5 000	-0.4200	0.2970	0.0011	0.0668	0.4903
$\sigma = 0.10$	10 000	-0.4200	0.2970	0.0005	0.0470	0.4963
	20 000	-0.4200	0.2970	0.0035	0.0330	0.5022
	5 000	-0.4200	0.2970	0.0037	0.0537	0.5147
$\sigma = 0.15$	10 000	-0.4200	0.2970	-0.0005	0.0377	0.4982
	20 000	-0.4200	0.2970	0.0011	0.0265	0.4886

exposure. Graphical analysis provides insights into the RDD results for causal analysis. For example, if the graph has a discontinuity, this would suggest that the intervention had a causal effect on the outcome, whereas if the graph is continuous, then it would suggest that there is no causal effect that can be

attributed to the intervention. Current literature on MRDD that we are aware of has limited discussion of a direct extension of conventional RDD graphs to MRDD. Cheng (2016), offers a *suboptimal* extension to MRDD, as well as used a different approach, the "slicing" and "sliding window" plots. In this study we deploy two-dimensional plots for each of the causal effects, as shown in Figures 5.3 and 5.4. The graphs compare two treatments at a time, yielding four plots in total (one for each pair of treatments being compared).
Using the function **"rdplot()"** in the R package **"rdrobust"** Calonico et al. (2017), the outcome variable is plotted as a function of the assignment variable, as shown in Figures 5.3 and 5.4. The graphs show that there are discontinuities at the cut-off points, thereby, providing evidence of a non-zero treatment effect. In addition, the graphs show that the relationship between the outcome variable and the assignment variables is approximately linear within a very small bandwidth of the assignment variables. Thus, the graphs show that the linear specification model is a plausible and credible model for estimating the treatments effects within a narrow bandwidth.

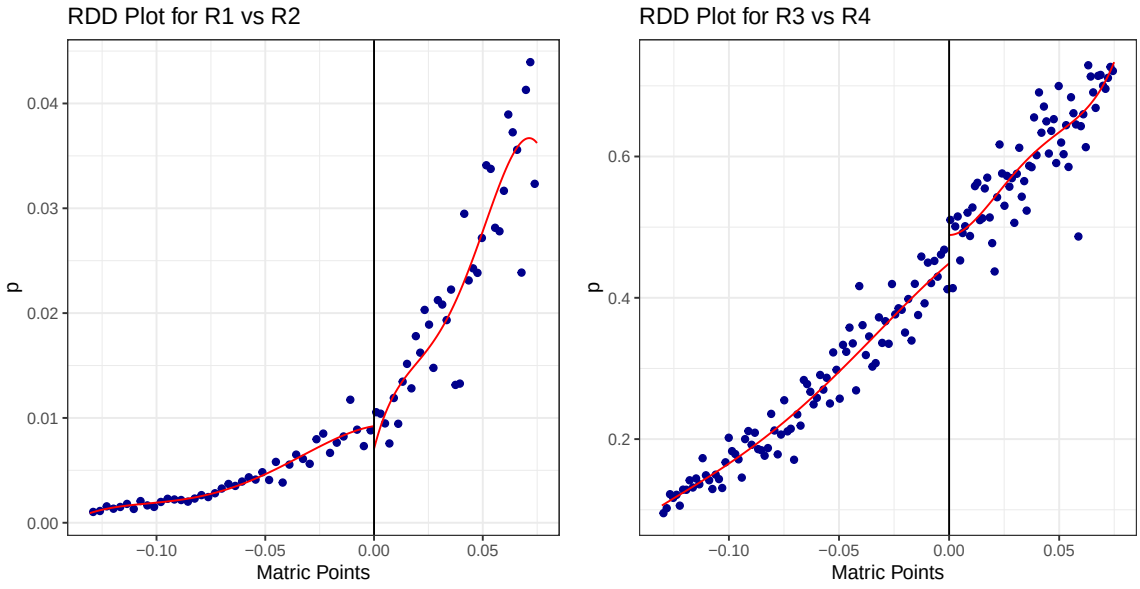


Figure 5.3: **Left**: **Causal effect 1:** Effect of $MP^c >= 0$ over $MP^c < 0$ for $INC^c > 0$. **Right**: **Causal effect 2:** Effect of $MP^c >= 0$ over $MP^c < 0$ for $INC^c <= 0$.

5.5.2.2 Manipulation testing using local polynomial density estimation

McCrary (2008) introduced the McCrary manipulation test to check for evi-

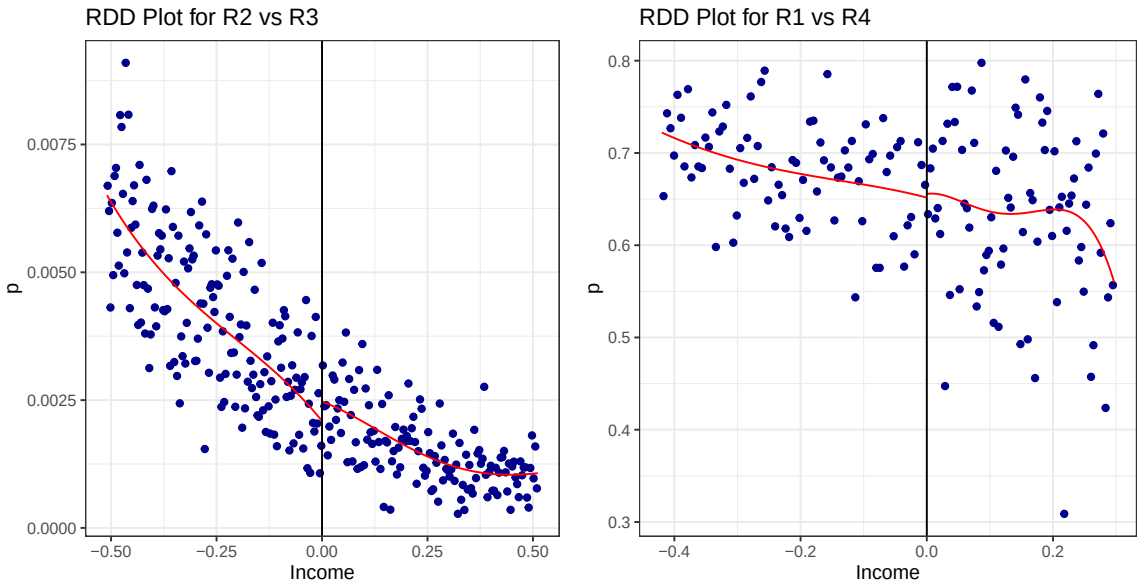


Figure 5.4: **Left**: **Causal effect 3:** Effect of $INC^c > 0$ over $INC^c <= 0$ for $MP^c < 0$ **Right**: **Causal effect 4:** Effect of $INC^c <= 0$ over $INC^c > 0$ for $MP^c >= 0$.

dence of manipulation near the income or matric cut-off points. However, in this study, we employ a newer method proposed by Cattaneo et al. (2018a) that uses local polynomial density estimators (Cattaneo et al., 2020) for manipulation testing. The method uses the **rddensity()** function in the R package **"rddensity"** to implement manipulation testing procedures. The test is robust to local polynomial order specifications and different bandwidths (Cattaneo et al., 2018a). Manipulation testing using local polynomial density estimation, tests the null hypothesis (H_0) of no discontinuity of density around the cut-off point versus the alternative hypothesis that the density is discontinuous around the cut-off points. A test statistic, $\mid T_q(h_l, h_r) \mid$ is computed for a given α level of significance (Cattaneo et al., 2020). H_0 is rejected if and only if $\mid T_q(h_l, h_r) \mid \geq \phi_{1-\alpha/2}$. Thus, H_0 indicates that there should not be any difference in the chance of eligibility for NSFAS funding for students on either side of the income or matric points cut-off points. Any differences should be attributed to the treatment effect only. The manipulation test checks whether this is actually true in our data. If students are able to choose their side of the cut-off points in order to influence the outcome, then we might worry that students on either side of the cut-off points are not comparable. The results of the manipulation tests for $\sigma = 0$ and $N = 10\ 000$ are shown in Table 5.7. For a given bandwidth, $T_q(h_l, h_r)$ is the final manipulation test statistic. For

Table 5.7: Examining Manipulation at the Income and Matric Points Cut-off Points.

Causal Effect	*Bandwidth*	$\mid T_q(h_l, h_r) \mid$	p-value
R3 vs R4	(-0.1307, 0.1248)	0.4809	0.6306
R1 vs R2	(-0.1922, 0.2859)	0.2300	0.8181
R2 vs R3	(-0.5097, 0.5114)	0.9122	0.3617
R1 vs R4	(-0.4200, 0.2970)	0.3571	0.721

$p < 0.05$ '*', $p < 0.01$'**', $p < 0.001$ '***'

example, for R3 vs R4, which represents $MP^c < 0$ vs $MP^c >= 0$ when $INC^c <= 0$ and cut-off, $c = 0$, $\mid T_q(-0.1307, 0.1248) \mid = 0.4809$, with a p-value = 0.6306. This means that we fail to reject H_0 and conclude that there is no statistical evidence of systematic manipulation of the matric points. Also, the manipulation test results for *R1 vs R2, R2 vs R3, and R1 vs R4* indicate that there is no evidence of manipulation of the assignment variables.

5.5.2.3 Sensitivity to Optimal Bandwidth Selection

We investigate whether the causal estimates critically dependent on a particular bandwidth choice, i.e., how sensitive the causal estimates are to outcomes of the units that are located very close to the cut-off points. Cattaneo et al. (2019) describe the implementation of a newer method that employs local polynomial methods to analyse the sensitivity of the causal estimates to the bandwidth choice. The method investigates sensitivity to bandwidth choice by removing or adding units at the end-points of the neighbourhood. This method is different from the continuity-based approach that changes the bandwidths that are used for local polynomial estimation. Also, the newer approach uses finite-sample methods to select a bandwidth that is closer to the cut-off points where the local randomization assumption is probably justified. The **rdwinselect()** command contained in the **"rdlocrand"** R package (Cattaneo et al., 2018b) is employed to check for sensitivity of the causal estimates to the bandwidth that is used. The **rdwinselect()** command automatically selects a window around the cut-off where the treatment can plausibly be assumed to have been *as-if* randomly assigned (Cattaneo et al., 2019). If the causal estimates critically depend on a particular bandwidth, then they are less *credible*. Table 5.8 shows the recommended windows around the cut-off points that were selected

for each of the treatment region comparisons;

Table 5.8: Minimum window around the cut-off points where the treatment can plausibly be assumed to have been randomly assigned.

Treatment Region	**Window**
R3 vs R4	(-0.0037, 0.0029)
R1 vs R2	(-0.0455, 0.0541)
R2 vs R3	(-0.095, 0.1336)
R1 vs R4	(-0.0973, 0.1237)

After choosing the appropriate minimum windows around the cut-off points using the **rdwinselect()** function that are shown in Table 8, different statistical tests such as the: difference-in-means (DM), Kolmogorov-Smirnov (KS), Wilcoxon rank sum (RS) can be implemented to test whether or not the treatment effects are different from zero or not. In this study, we implemented the difference-in-means (DM) of the outcomes on either side of the cut-off point using the **rdrandinf()** command in the **"rdlocrand"** R package. The DM procedure tests the Fisherian null hypothesis (Cattaneo et al., 2019), that the treatment effect is zero for all units. Cattaneo et al. (2019) points out that the interpretation of the difference-in-means test statistic in the Fisherian framework is different because it tests the sharp null hypothesis of no treatment effect, and it should not be interpreted as an estimated treatment effect. Its main purpose is to test the null hypotheses that are sharp (e.g., no treatment effect), not on point estimation. This is much stronger than testing the hypothesis that the average treatment effect (ATE) is zero. If the treatment effects are not statistically significant using the minimum window when in the first place we obtained statistically significant treatment effects using the optimal bandwidths and vice versa, then it indicates that the treatment effects are sensitive to the bandwidth used. Table 5.9 shows the DM results for each treatment region. The DM for the treatment regions R3 vs. R4 and R1 vs. R2 was significant at $\alpha = 0.05$ using the automatically determined minimum bandwidths. Table 5.5 shows that the treatment effects for R3 vs R4 and R1 vs R2 are all significant for $\sigma = 0$ and $N = 10\ 000$. These results show that causal estimates do not depend critically on a particular bandwidth. Furthermore, Table 5.9 shows that the DM for the treatment regions R2 vs R3, and R1 vs R4 are not statistically significant at $\alpha = 0.05$ using the automatically determined minimum bandwidths. The treatment effects for R2 vs R3, and R1

Table 5.9: Difference-in-means (DM) test statistics for the treatment regions under investigation.

Treatment Region	DM Test statistic (T)	p-value
R3 vs. R4	0.383	0.000
R1 vs. R2	0.138	0.000
R2 vs. R3	-0.003	0.103
R1 vs. R4	-0.078	0.132

vs R4 are also not statistically significant for $\sigma = 0$ and $N = 10\ 000$ as shown in Table 5.6. These results provide evidence that the causal estimates obtained in the simulation studies are *credible*, as they are very robust to bandwidths that are close to the cut-off points for income and matric points.

5.6 Case Study

5.6.1 Application of the MRDD using the Graduate Admission Data Set

Acharya et al. (2019) used graduate admission data to predict whether students from India can be admitted or not to their university of choice for a graduate master's program in science and technology in the USA. Instead of predicting the chance of admission, we estimate the causal effect of college admission variables on the chance of admission to a master's program in the United States using the frontier MRDD. As we implement a frontier MRDD with two assignment variables, feature importance curve plots are used to determine two important variables that influence the "chance of admit" the most. We use the Boruta algorithm (Kursa et al., 2010), which is designed as a wrapper around a random forest classification algorithm. This algorithm iteratively eliminates the statistically insignificant features. It is implemented using the Boruta R package (Kursa et al., 2010). Thus, the Boruta package is used to select the features that are important to the outcome variable, "chance of admit". The data used in this case study are available in Acharya et al. (2019). The graduate admission data set consists of the variables that are considered carefully by most universities' graduate admission committees, such as; graduate record examination (GRE) scores, test of English as a foreign language (TOEFL), cumulative grade point average (CGPA) (out of 10), statement of purpose

strength (SOP)(out of 5), university rating (out of 5) and chance of admit (out of 1). In India, the CGPA is calculated on a ten point scale, with the CGPA score of 10 being the maximum that can be attained.

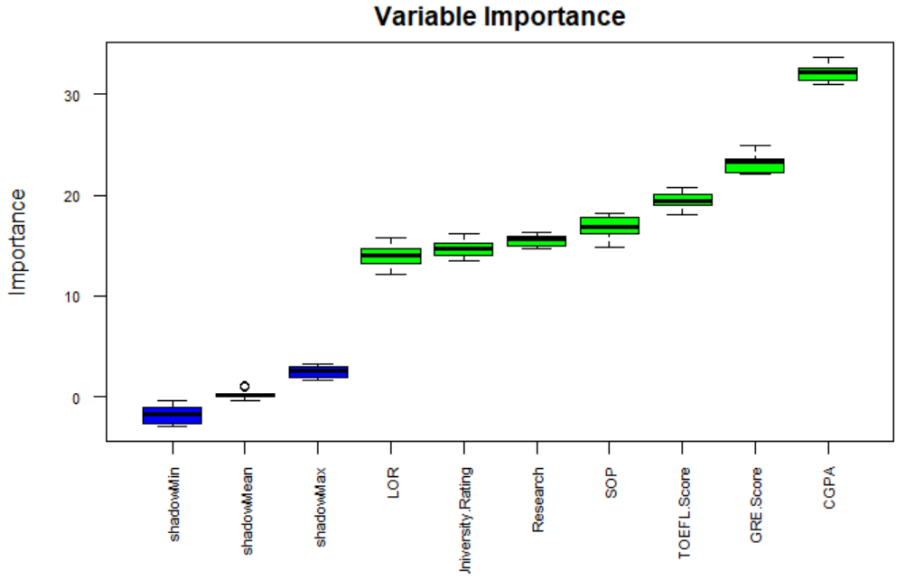


Figure 5.5: The Importance of each Graduate Admission Variable using Boruta.

Figure 5.5 shows that the chance of admission to a master's program is mainly dependent on the CGPA of the candidate followed by the GRE score (ie, the standardized test scores). Thus, CGPA and GRE will be used as assignment variables in the frontier MRDD analysis. The other variables, TOEFL, statement of purpose strength (out of 5), and university rating (out of 5) will be used as covariates to improve the precision of estimates of causal effects. We focus on engineering students and graduates from India who apply for master's programs in science and technology in the USA. Raghunathan (2010) recommends a total GRE score of 322 for a student to stand a higher chance of being admitted to a graduate program of choice. Thus, we set a GRE cut-off of 322. Similarly, Raghunathan (2010) highly recommends a GPA of $\approx$ 8.5 in order for a candidate not to be directly rejected due to poor academic performance. Thus, we use 8.5 and 322 as cut-off points for CGPA and GRE respectively.

5.6.2 Estimation of the Causal Effects of CGPA and GRE

To estimate the effect of scoring CGPA ≥ 8.5 over CGPA < 8.5 for students who have GRE < 322 or GRE ≥ 322, Equation 5.9 (similar to Equation 5.7) is fitted to the data.

$$P(Admit = 1|CGPA^c, T_{cgpa}, GRE^c, TOEF^c) = \beta_0 + \beta_1 T_{cgpa} + \beta_2 CGPA^c + \beta_3 CGPA^c * T_{cgpa} + \beta_4 GRE^c + \beta_5 TOEFL_i^c + \epsilon_i, \tag{5.9}$$

The following causal effects are determined using the assignment variables "CGPA" and "GRE" with respective cut-offs equal to 8.5 and 322. In addition, the variables are centred around their respective cut-off points to give $CGPA^c$ and GRE^c. To estimate the effect of GRE, $CGPA^c$ and T_{cgpa} variables are replaced by GRE^c and T_{gre} in Equation 5.9 respectively. The data used were of size, $N = 500$ at the time of writing this paper.

1. **Causal Effect 1**: $CGPA^c < 0$ vs $CGPA^c >= 0$ for $GRE^c >= 0$
2. **Causal Effect 2**: $CGPA^c < 0$ vs $CGPA^c >= 0$ for $GRE^C < 0$
3. **Causal Effect 3**: $GRE^c < 0$ vs $GRE^c >= 0$ for $CGPA^c >= 0$
4. **Causal Effect 4**: GRE^c <0 vs $GRE^c >= 0$ for $CGPA^c < 0$

The results (**Causal Effects 1-4**) of fitting to Equation 5.9 the graduate admission data set are shown in Table 5.10.

Table 5.10: The results of fitting to Equation 5.9 the graduate admission Data Set.

Causal Effect	h_l	h_r	β_1	***s.e***	***p*-value**
Causal Effect 1	-0.400	0.550	0.403	0.104	0.000
Causal Effect 2	-0.200	0.480	-0.035	0.016	0.025
Causal Effect 3	-24.0	18.0	0.607	0.028	0.000
Causal Effect 4	-23.0	13.0	0.118	0.018	0.000

Table 5.10 shows that **Causal Effects 1-4** are significant at a 5% level of significance. **Causal Effect 1** is equal to the effect of having a CGPA score that is greater than or equal to 8.5 compared to having a CGPA score that is less than 8.5 for all students whose GRE score is already greater than or equal

to 322. The results show that having a CGPA score greater than 8.5 compared to having a CGPA less than 8.5 increases the chance of admission to graduate school by a significant 40.3 percentage points (p-value = 0.000 < 0.05) for students from India who already have a GRE score that is greater than 322. Similarly, **Causal Effect 3** shows that having a GRE score that is greater than 322 compared to having a GRE score that is less than 322 increases the chance of graduate college admission by a significant 60.7 percentage points (p-value = 0.000 < 0.05) for students who already have a CGPA score greater than 8.5. The effect of scoring a CGPA score greater than 8.5 compared to scoring a CGPA less than 8.5 for students who already have a GRE score < 322 (**Causal Effects 2**), significantly reduces the chance of admission to a graduate master's program by ≈ 3.5 percentage points (p-value = 0.025 < 0.05). On the other hand, obtaining a GRE score > 322 compared to obtaining a GRE score < 322 (**Causal Effect 4**) increases the chance of college admission by ≈ 11.8 percentage points for students who already have CGPA < 8.5 for students who already have a CGPA score lower than 8.5. These results suggest that GRE and CGPA are very important obstacles that must be overcome if one is to have a realistic chance of being admitted to the graduate master's program. This means that for students from India intending to apply to study for graduate master's programs in science and technology in the USA, they must achieve competitive GRE and CGPA of at least 322 and at least 8.5 respectively.

Table 5.11: Examining manipulation at the CGPA and GRE cut-off points.

Causal Effect	h_l	h_r	$T_q(h_l, h_r)$	***p*-value**
Causal Effect 1	-0.400	0.550	-1.110	0.267
Causal Effect 2	-0.200	0.480	-0.701	0.483
Causal Effect 3	-24.0	18.0	0.921	0.357
Causal Effect 4	-23.0	13.0	1.505	0.133

Table 5.11 shows the results of the manipulation test of the CGPA and GRE scores. The results show that there is insufficient evidence to reject the null hypothesis (H_0) of no discontinuity of density around the cut-off points for **Causal Effects 1-4**. This suggests that there is insufficient evidence of manipulation of the CGPA and GRE scores. Thus, causal estimates are *credible*. Table 5.12 shows the difference-in-means (DM) tests for the Fisherian null hypothesis (Cattaneo et al., 2020), that the treatment effect is zero for all units.

We reject the Fisherian null hypothesis that the treatment effect is zero for **Causal Effects 1-4** and conclude that the causal effects that are all statistically significant at the 5% level of significance since the p-values of the DM statistics (T) shown in Table 5.12 are all less than *0.05*. This shows that causal estimates are not sensitive to the choice of bandwidth because the causal estimates shown in Table 5.10 are also statistically significant at the 5% level. This means that the frontier MRDD produced *credible* causal estimates.

Table 5.12: Tests for difference in means (DM) using the minimum window (bandwidth).

Causal Effect	Minimum Window (Bandwidth)	DM Test Statistic (T)	p-value
Causal Effect 1	(-0.4, 0.42)	0.347	0.001
Causal Effect 2	(-0.2, 0.19)	0.005	0.035
Causal Effect 3	(-2, 1)	0.665	0.000
Causal Effect 4	(-4, 5)	0.310	0.000

5.7 Discussion and Conclusions

5.7.1 Discussion

This paper has highlighted the importance of using an appropriate regression discontinuity approach when faced with the complexity caused by having more than one assignment variable. In this study, we have used the frontier MRDD because it is easy to implement. It reduces the assumptions checking process to a series of well defined single assignment variables RDD, whose methods are well defined in the literature (Reardon and Robinson, 2012). A data-driven bandwidth selection method was deployed in selecting an optimal bandwidth, thus eliminating the possibility of manipulating the results by choosing an arbitrary bandwidth. Researchers seeking to use local linear regression must ensure that there is a sufficient density of points around the cut-off points. The insufficient density of points at the threshold point may result in less credible causal estimates. density of points at the threshold point may result in less *credible* causal estimates. The simulation results produced significant causal estimates when the data were random with no induced variability in

the assignment variables. For example, according to our analytical results, barely scoring matric points greater than or equal to 25 points compared to scoring matric points less than 25 for students whose household income is less than R350 000, increases the probability of eligibility for NSFAS funding by 3.75 percentage points (Table 5.5 for $\sigma = 0$). When the level of variability in the two assignment variables, MP^c and INC^c increased from $\sigma = 0$ to $\sigma = 0.15$ in steps of 0.05, while keeping the cut-offs constant, the estimated treatment effects decreased respectively for each level of σ. This showed that the frontier MRDD approach was not suitable for handling data with high variability. Thus, for future work, we may consider developing methods that can handle variability in the assignment variables better. Furthermore, the frontier MRDD works very well when $\sigma = 0$, that is, when there is no induced variability in the assignment variables and the treatment effect estimates were comparable and credible for the different sample sizes.

We have not only reported the causal effects but also carried out supplementary analyses to assess the *credibility* of the primary causal effect estimates. We found evidence of a relationship between the outcome variable and the assignment variable(s) through graphical analysis. Figures 5.3 and 5.4, show "visual" evidence of disturbances or discontinuities in the expected smooth relationship between the outcome variable and the assignment variable(s). This gives credibility to the causal estimates as it indicates that the treatments effects actually exist at the cut-offs. The relationship between the probability of eligibility for NSFAS funding and either one of the assignment variables (income or matric points) is assumed to exhibit smooth functions. This assumption of smooth functions is based on the *continuity-based* framework that is used to explain the required identification assumptions intuitively, and also in developing causal effect estimates. In addition, we have assessed the existence of causal effects through an alternative causal framework that makes use of the local randomization framework (Cattaneo et al., 2015). The local randomisation framework is different from the *continuity-based* framework in that it formalises the idea of a local randomized experiment near the cut-off by embedding the RD design in a classical, Fisherian causal model, thus giving interpretation and justification to randomization inference and related classical experimental methods (Hill et al., 2017). Using the local randomisation framework, we analysed the units that are in a small window around the cut-

off points "as-if" they were randomly assigned to treatment or control. This enabled us to use statistical tools such as the difference-in-means (DM) to test the Fisherian null hypothesis, that the treatment effect is zero for all units. The Fisherian null hypothesis of no treatment effect was applied in both the simulation and the case study. For example, in the case study, we did not apply graphical analysis to visually inspect whether a treatment effect existed or not, but we rejected the Fisherian null hypothesis of no treatment effect, thus validating the presence of treatment effects for **Causal Effects 1-4**. The local randomisation assumption near the cut-off has allowed us to use a newer method proposed by Cattaneo et al. (2015) in assessing the credibility of causal estimates in a simulation study as well as in a real world data set. In addition, for the continuous assignment variables, MP^c and INC^c in the simulation study, we have used graphical analysis (or the continuity-based framework) to detect the presence of a discontinuity or treatment effect and then used the local randomization approach (Fisherian null hypothesis) as a robustness check.

To test for evidence of manipulation of the assignment variables, we used a newer intuitive and easy-to-implement nonparametric density estimator that is based on local polynomial techniques (Cattaneo et al., 2020). The authors indicate that the estimator is fully automatic and does not require any other transformation of the data. Students must not be able to influence their position relative to the cut-off scores. The results of the local polynomial density manipulation testing for both the simulation study (Table 5.7), and the case study (Table 5.11) show that the students were unable to manipulate or choose their preferred side of the assignment variable(s) cutoff point, thus making the causal estimates more credible. Furthermore, evidence of no manipulation holds because all students may not precisely manipulate their own matric points. This gives credibility to the causal effects estimated in both the simulation and the case study.

Sensitivity to bandwidth choices was evaluated by examining the robustness of our causal estimates to changing bandwidths. Table 5.9 shows that when the causal estimates were estimated using narrow windows around the cut-off points, they were all significant at the 5% level of significance. This suggests that our average causal estimate results remain largely robust to changing bandwidth choices.

The case study results show that the frontier approach can be implemented successfully using real world data. Using Indian graduate admission data for master's programs in science and technology studies in the USA, we obtained significant causal estimates (Table 5.10). Additionally, the results of the manipulation test (Table 5.11) indicated that there was insufficient evidence to reject the null hypothesis H_0 of no discontinuity. This suggests that there was no evidence of manipulation of the GRE or CGPA to gain admission to graduate master's programs for Indian students. Also, Table 5.12 shows that using the minimum window (bandwidth) around the cut-off points, the Fisherian null hypothesis of no treatment effect is rejected, thereby indicating that the causal estimates were credible and not sensitive to bandwidth choice.

5.7.2 Limitations

A limitation of the study was that consolidated data of household income and matric points were not available from NSFAS at the time of writing this paper and we could not directly apply our approach to real income and matric points data. However, we have shown that the approach we have adopted still works for MRDD with two assignment variables and can uncover valuable causal effects of interest. For future work, we strongly recommend that further research should focus on the determination of the minimum sample size required for two or more assignment variables. Furthermore, future simulation studies can be used to examine the effect of variations on the correlation and distribution of assignment variables. In the simulation studies we used samples of size 5 000, 10 000, and 20 000 and in the case study we used data of size 500. Determining the minimum sample required to obtain credible causal estimates will assist those wishing to apply MRDD. Determining the minimum sample size enables one to have enough density of data points around the cut-off point, and thus generate more *credible* estimates. In generating the outcome variable, the possibility of using different combinations of the data generation distributions could be explored. Future work could also look at studying the effect of NSFAS grants on subsequent enrollment rates for a bachelor's degree and not just the probability of eligibility for NSFAS funding. One could also examine whether or not NSFAS grants influence the choice of universities to attend or influence the number of years students spend studying for their bachelor's degree.

5.8 Conclusions

In the absence of randomised controlled experiments, regression discontinuity designs offer more opportunities to estimate causal effects by making use of the variation in the treatment assignment induced by a cut-off point. One major finding of this article is the discovery of new evidence that both matric points and income have a huge impact on the probability of getting NSFAS funding to study at any university in South Africa. This evidence will inform policy makers and educational experts about the effects of matric points and income on the chance of eligibility for NSFAS funding. The availability of the NSFAS grant has a huge impact on students' decisions to attend university or seek other opportunities elsewhere. In summary, this paper makes valuable contributions to the literature on multivariate regression discontinuity designs by conducting supplementary analyses that seek to add more *credibility* to the causal estimates obtained through primary analyses. If one is interested in determining causal effects of barely meeting the requirements of one assignment variable, among the subjects that either meet or fail to meet the requirements of the other assignment variable, then we strongly recommend the use of the *frontier* multivariate regression discontinuity design as it is easy to implement and incorporates discontinuities in multivariate assignment variables into single regression discontinuity designs along a number of frontiers of the treatment variables.

CHAPTER 6

Deep Learning for SARS COV-2 Genome Sequences

ALBERT WHATA [1,*] and CHARLES CHIMEDZA [2]

[1] School of Natural and Applied Sciences, Sol Plaatje University

[2] School of Statistics and Actuarial Science, University of the Witwatersrand

Published in: *IEEE Access*, (2021) **9**: 59597-59611. Impact factor: **3.745**

Statement of Contributions of Joint Authorship

Albert Whata **(Candidate)**

Conducted the research, writing and compilation of manuscript);

Charles Chimedza **(Supervisor)**

Supervised, edited and coauthor of the manuscript.

This Chapter is an exact copy of the journal paper (Whata and Chimedza, 2021a) referred to above, and available at `https://ieeexplore.ieee.org/document/9405995`**.**

ABSTRACT

The SARS-CoV-2 virus which originated in Wuhan, China has since spread throughout the world and is affecting millions of people. When there is a novel virus outbreak, it is crucial to quickly determine if the epidemic is a result of the novel virus or a well-known virus. We propose a deep learning algorithm that uses a convolutional neural network (CNN) as well as a bi-directional long short-term memory (Bi-LSTM) neural network, for the classification of the severe acute respiratory syndrome coronavirus 2 (SARS CoV-2) amongst coronaviruses. In addition, we classify whether a genome sequence contains candidate regulatory motifs or otherwise. Regulatory motifs bind to transcription factors. Transcription factors are responsible for the expression of genes. The experimental results show that at peak performance, the proposed convolutional neural network bi-directional long short-term memory (CNN-Bi-LSTM) model achieves a classification accuracy of 99.95%, area under curve receiver operating characteristic (AUC-ROC) of 100.00%, a specificity of 99.97%, the sensitivity of 99.97%, Cohen's Kappa equal to 0.9978, Mathews Correlation Coefficient (MCC) equal to 0.9978 for the classification of SARS CoV-2 amongst coronaviruses. Also, the CNN-Bi-LSTM correctly detects whether a sequence has candidate regulatory motifs or binding-sites with a classification accuracy of 99.76%, AUC ROC of 100.00%, a specificity of 99.76%, a sensitivity of 99.76%, MCC equal to 0.9980, and Cohen's Kappa of 0.9970 at peak performance. These results are encouraging enough to recognise deep learning algorithms as alternative avenues to detect SARS CoV-2 as well as detecting regulatory motifs in SARS CoV-2 genes.

6.1 Introduction

The SARS-CoV-2 virus which originated in Wuhan, China has since spread throughout all the provinces in China and the world and is affecting millions of people(Mackenzie and Smith, 2020). When there is a novel virus outbreak,

it is crucial to quickly determine if the epidemic is a result of the novel virus or a well-known virus. This means that the proper classification of novel viruses such as SARS-CoV-2 and detecting regulatory or transcription motifs in these viruses can assist scientists in deciding on the methods and measures that are suitable to identify the viruses, control their transmission rates and limit potential consequences that may be caused by these viruses.

The identification of SARS-CoV-2 can give misleading results because the virus is hard to differentiate from other viruses in the *Coronaviridae* family, due to the genetic similarities among the viruses in this family (Lopez-Rincon et al., 2021). This presents a challenge in that the detection of SARS CoV-2 viruses can yield false positives because of the presence of other viruses that are very similar to SARS CoV-2 (Metsky et al., 2020). Also, Metsky et al. (2020) state that those patients who are suspected to have SARS-CoV-2 may present symptoms that are sometimes similar to a different respiratory viral infection. Therefore, it is of paramount importance to accurately characterise the SARS CoV-2 virus from similar viruses to enhance patient diagnostics and also manage the outbreak of SARS CoV-2 virus.

SARS-CoV-2 is spreading fast due to the lack of accuracy in the detection tools that are currently used in practice (Lopez-Rincon et al., 2021). In addition, SARS-CoV-2 is a typical ribonucleic acid (RNA) virus that produces new mutations in a replication cycle of Coronavirus, with an average evolutionary rate of about 10^{-4} nucleotide substitutions per site each year (Lu et al., 2020). This has brought into question the current techniques that are used to detect SARS-CoV-2. The reverse transcription-quantitative real-time polymerase chain reaction (RT-qPCR) is a molecular tool that is widely used in detecting SARS CoV-2 in patients. The RT-qPCR technique combines RT-PCR with qPCR to enable the measurement of RNA levels through the use of cDNA in a qPCR reaction (Adams, 2020). According to Lopez-Rincon et al. (2021), RT-qPCR has used ORF1ab and N genes to identify SARS CoV 2. Also, RT-qPCR has been questioned by Yang et al. (2020) who report that the technique has achieved a negative rate of 17.8% when sputum samples were used in mild cases and 11.1% negative rate for severe cases. The techniques achieved negative rates of 26.7% and 27.0% in severe and mild cases respectively when applied on nasal swabs. In addition, the technique achieved negative rates of 40.0% and 38.7% in severe and mild cases respectively when

applied on throat swabs. These variations may be a result of the variations that are present in the RNA sequences of the viral species (Lopez-Rincon et al., 2021). Apart from giving false-negatives, the RT-qPCR technique can detect a small percentage of other similar coronaviruses that may be present in a simple which may hinder the positive identification of SARS CoV-2 (Lopez-Rincon et al., 2021). Furthermore, Zhao et al. (2020a) indicates that about 35.2% of 173 samples did not test positive when the technique was used. Also, Long et al. (8961) report that real-time RT-PCR may initially produce false-negative results, and they suggested that patients with typical computed tomography (CT) findings, but negative real-time RT-PCR should repeat the real-time RT-PCR to avoid misdiagnosis.

As mentioned earlier, SARS CoV-2 is like other viruses in the *Coronaviridae* family, and its identification can be difficult. Therefore, we will explore how deep learning methods can be used to accurately identify SARS CoV-2 from other coronaviruses. These methods can then be used to complement the existing molecular testing techniques to improve the detection rates of SARS CoV 2.

According to Dinka and Milkesa (2020), motifs are approximate short nucleotide sequences that occur repetitively in similar groups of sequences. The regulatory motifs are used to control the expression of genes, i.e., they are responsible for turning a gene on or off. Also, transcription factors (TFs) are proteins that attach to deoxyribonucleic acid (DNA). The main function of TFs is to convert or transcribe DNA into Ribonucleic acid (RNA). TFs attach themselves to DNA sequences and become responsible for turning on or off genes through a process called "gene expression". A particular TF binds to a specific site called a transcription factor binding site (TFBS), thus, regulates cell machinery (Hannenhalli, 2008).

It can be challenging in bioinformatics to identify regulatory motifs in DNA sequences (Bellora et al., 2007). This is because motifs are short sequences and their prediction usually results in several unacceptable false positives. In this paper, we will focus on regulatory motifs that are shared by the SARS CoV-2 genes in classifying whether a given sequence contains regulatory motifs for the SARS CoV-2 or not. Using deep learning, we focus on detecting nucleotides that are important in predicting whether a given sequence contains regulatory motifs for the SARS CoV-2 virus. The analysis of regulatory mo-

tifs is important for making improvements in medical treatment and gaining valuable knowledge about cell processes. For example, analysis of regulatory motifs may help better understand mutations that may affect the regulatory mechanism of gene expression.

We propose a hybrid deep learning algorithm that integrates a state-of-the-art CNN-Bi-LSTM to classify the SARS CoV 2 virus from other coronaviruses as well as classify whether a given sequence contains regulatory motifs for the SARS CoV-2 or not. This paper makes the following specific contributions:

1. Develop an alignment-free method for classifying SARS-CoV-2 gene sequences amongst coronaviruses' genes,

2. Develop a deep learning algorithm that can efficiently classify whether a SARS CoV-2 genome sequence contains candidate regulatory motifs and

3. Compare the classification performances of our proposed CNN-Bi-LSTM versus the CNN and CNN-LSTM.

6.1.1 Problem Statement

Detecting whether a given sequence contains regulatory motifs for the SARS-CoV-2 gene, as well as identification of SARS CoV-2 genes amongst coronaviruses, can be viewed as binary classification problems in that we have a data set $\mathcal{D}$ with N examples of input data together with their corresponding target classes: $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, and $\boldsymbol{X} \subset \mathbb{R}$ represents a feature space, which can be described as a matrix with dimensions, $4 \times N$. The length of the DNA sequence is, thus, represented by N. We consider a value $N = 100$ base pairs (bp) in this paper. Additionally, $\mathbf{Y}$ is a dichotomous variable in the standard space $\{0, 1\}$ (Zhang et al., 2020). As discussed earlier, there are four bases in DNA sequences namely: Adenine (A), Thymine (T), Guanine (G), and cytosine (C). These four base pairs form the sequence of base pairs {A, T, C, G} (Zhang et al., 2020). These base pairs can be characterised by one of the following one-hot vectors [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0] and [0, 0, 0, 1]. The SARS CoV-2 genes are like the other genes in the Coronavirus family (Lopez-Rincon et al., 2021), therefore, their classification can give rise to false results. Therefore, the major goal of this paper is to predict accurately SARS-CoV-2 gene sequences from amongst the coronaviruses' genes. Additionally, we

classify whether a genome sequence contains candidate regulatory/promoter motifs for SARS CoV-2 genes.

6.1.2 Related Work

Traditionally, the classification of genome sequences has used alignment-based techniques which include the Basic Local Alignment Search Tool (BLAST) (Altschul et al., 1990) and the Burrows-Wheeler Aligner (BWA) (Li and Durbin, 2009). Such techniques rely on annotating viral genes (Roux et al., 2019). Alignment-based methods such as BLAST have been successful in finding sequence similarities (Zielezinski et al., 186). However, in practice, these methods require heavy computational time when they are used to analyse thousands of complete genomes (Randhawa et al., 2391). Zielezinski et al. (186); Randhawa et al. (2020) mention that the alignments assume that the genes are homologous, i.e., they have the same continuous structure. However, in practice, this is not always the case.

Several alignment-free computational approaches (Zeng et al., 2016; Zou et al., 2019) have been used to predict deoxyribonucleic acid (DNA) protein binding. DeepFam which does not require the alignment of genes for predicting and modelling proteins was proposed by Seo et al. (2018). DeepFam uses a feed-forward convolution neural network. It achieved better accuracy and faster run-time for predicting binding proteins when compared to methods that required the alignment of sequences as well as those that did not require the alignment of sequences (Seo et al., 2018). Randhawa et al. (2020) proposed a Machine Learning with Digital Signal Processing-Graphical User Interface (MLDSP-GUI), which is an alignment-free tool for DNA sequence comparisons and analysis. The authors highlight that the tool was designed to address issues that are associated with the alignment of DNA sequences.

Our proposed model, CNN-Bi-LSTM is an alignment-free algorithm that consists of CNN layers followed by Bi-LSTM layers that capture the temporal effects in deoxyribonucleic acid (DNA) sequences (Zhang et al., 2020). DNA is made of nucleotide sequences whose function is to store information in all cells. Each nucleotide is made of sugar (Deoxyribose in DNA and Ribose in RNA), a base, and a phosphate. There are four bases in DNA sequences namely: Adenine (A), Thymine (T), Guanine (G), and cytosine (C). According to (Zhang et al., 2020), these four base pairs form the sequence of base pairs {A, T, C,

G}. We consider SARS CoV-2 gene sequences as patterns of letters made from the four nucleotides, A, T, G, and C, and then use one-hot vectors to represent these sequences in a similar way to text data. We adopt the procedure by Nguyen et al. (2016) to translate DNA sequences into sequences of words. For example, Nguyen et al. (2016) indicates that a dictionary of 64 words is formed when a word of size three nucleotides is used. This means that a one-hot vector of size 64 can represent every three-letter word. This method results in a sequence of words that can be represented by a two-dimensional matrix that encompasses information about the precise location of each base in the sequence. This numerical matrix is the input that is subsequently fed into a CNN. Additionally, one-hot vectors that are used in this paper to represent SARS CoV-2 gene sequences can conserve information about the position of each base in sequences (Nguyen et al., 2016).

The use of CNN is inspired by its successes in modelling DNA sequences. For example, Zhang et al. (2019b) mention that CNNs have outperformed machine learning algorithms that include support vector machines (SVM) or random forests in predicting protein binding based on DNA sequences. Also, CNNs have been successfully used in DeepSea (Zhou and Troyanskaya, 2015) to predict the chromatin effects sequence alterations with single nucleotide sensitivity. In addition, using patterns learned from experimental data, DeepBind has used CNN to discover specific DNA and RNA binding proteins (Zhang et al., 2019b). The use of the CNN as part of an algorithm that can classify SARS CoV-2 gene sequences is also inspired by its successes in text classification (Amin and Nadeem, 2018). Additionally, CNN has been used in topic categorisation (Johnson and Zhang, 2015), spam detection (Roy et al., 2020), and Twitter sentiment analysis (Jianqiang et al., 2018).

Nguyen et al. (2016) states that one-dimensional sequences of successive letters can be used to represent text data. Therefore, one-hot vectors that are fed as input into CNN can be used to represent text data. Johnson and Zhang (2015) recommend the use of one-hot vectors because the use of look-up tables that match each word in a word-vector is tantamount to using uni-grams information, whereas bi-grams and n-grams could be more discriminating in classifying samples. Thus, the use of one-hot vectors and concatenating word vectors of words that are close will include the n-gram information into text classification.

We use the CNN layers first to provide better input to the Bi-LSTM layers by generating filters that generalise sequence patterns (Zhang et al., 2020). The LSTM layers incorporate the long and short-term information that is present in DNA sequences (Zhang et al., 2015). The use of the Bi-LSTM layers is to ensure that we can utilise both past and future inputs i.e., DNA sequences at a given point in time. This means that the Bi-LSTM layer can make use of past and future DNA sequences by capturing the long-term relationships of a DNA sequence through the application of the forward LSTM as well as the backward LSTM. According to Zhang et al. (2020), the Bi-LSTM layer can characterise a probably very complex order in the DNA sequence in an efficient manner. Zhang et al. (2020) developed DeepSite for predicting DNA-protein binding. DeepSite has Bi-LSTM network layer(s) followed by CNN layer(s). Quang and Xie (2016) developed DanQ, similar to DeepSea, which is also uses CNN layers and Bi-LSTM layers for predicting the non-coding function at the start of a sequence. Our proposed model extends the work of Quang and Xie (2016) in classifying SARS CoV-2 gene sequences from amongst coronaviruses as well as identifying sequences that contain regulatory motifs for the SARS CoV-2. Our model reverses the order of appearance of the Bi-LSTM and CNN layers in DeepSea.

6.2 Materials and methods

We propose a CNN-Bi-LSTM to classify SARS CoV-2 virus amongst coronaviruses and predict the short regulatory motifs (i.e., DNA binding motifs) that are bound to the proteins (transcription-factors). Our model is different from DeepSite (Zhang et al., 2020) in that, we start with CNN layers that feed into Bi-LSTM layers. We employ the CNN-Bi-LSTM to extend the work by Lopez-Rincon et al. (2021) to classify accurately SARS CoV-2 genes. Also, the CNN-Bi-LSTM extends the work of Zou et al. (2019) to predict DNA binding motifs. In addition, combining CNN and Bi-LSTM layers is motivated by Sainath et al. (2015) who indicated that LSTMs performances can be improved by using CNN to provide better features to the LSTM.

6.2.1 Data sets

The data set for classifying SARS CoV-2 genes amongst coronaviruses are summarised in Table 6.1. The data set was obtained from the NCBI genes database on November 1, 2020.

Table 6.1: Data for classifying SARS CoV-2 genes amongst coronaviruses.

Virus gene	Class Label	Number of Samples
SARS CoV-2	1	34
Non-SARS CoV-2	0	295

All repeating sequences were removed resulting in 329 unique sequences. All the virus genes belonged to the Coronavirus (CoV) family. We attached a label of 1 if a gene was that of SARS CoV-2 gene and 0 otherwise. The data was unbalanced with 10.3% positive SARS CoV-2 samples and 89.7% negative samples.

6.2.2 Algorithms

6.2.2.1 Convolutional Neural Networks (CNN)

CNNs consist of a convolutional layer, a non-linearity layer, a max-pooling layer, and a fully connected layer (Nguyen et al., 2016). CNNs have achieved outstanding performance in image classification, computer vision, and natural language processing (NLP) (Albawi et al., 2017). Also, they have been applied to text problems that include spam detection, sentiment classification and topic categorisation (Minaee et al., 2020). Text classification seeks to automatically classify text documents into one or more known categories. Text data is represented as a one-dimensional sequence of successive letters as opposed to image data which is represented as two-dimensional matrices. Therefore, if we are to use text data as an input in CNNs, we change the one-dimensional sequences of letters into a matrix or 2D tensor (Johnson and Zhang, 2015).

DNA sequences have patterns of successive letters that do not have space in contrast to text data which has space between words. These sequences are made up of "words" from the four nucleotides, A, T, G, and C (Deza and Deza, 2009). The words formed by the sequences do not have any meaning.

Nguyen et al. (2016) indicates that DNA sequences can be characterised using one-hot vectors into 2D matrices that are, then, fed into the next layer which in this work is a CNN layer. We will adopt the one-hot vectors proposed by Johnson and Zhang (2015); Nguyen et al. (2016) to represent DNA sequences as 2D matrices.

A big argument for incorporating CNNs in our proposed model is that they are fast and efficient in terms of representation of text or sequences (Young et al., 2018). Thus, we use a deep learning algorithm that combines a CNN and Bi-LSTM to detect sequences with regulatory or transcription motifs and also for the classification of SARS-CoV-2 genes amongst other Coronavirus genes.

6.2.2.2 Long short-term memory network (LSTM)

Hochreiter and Schmidhuber (1997) introduced long short-term memory networks (LSTM) which are capable of learning long-term dependencies through recurrently connected memory blocks (subnets). Long short-term memory networks (LSTMs) are an example of recurrent neural networks (RNN) (Hochreiter and Schmidhuber, 1997). RNNs described in detail in Goodfellow et al. (2016) are deep neural networks that can process sequential data where outputs are dependent on the previous computations. However, RNNs are easily affected by the vanishing gradients problem (Le and Zuidema, 2016). Thus, RNNs become biased as they only deal with short-term data points. For time or sequence-dependent data, an RNN takes the output of a layer at time t and feeds it as part of the input of a layer at time $t + 1$. LSTM operates above the RNN and they add some memory components that assist in propagating the knowledge learned at a time t to the longer-term time-steps, (e.g, $t + 1$, $t + 2$,...). The most important function of an LSTM is to overlook insignificant parts of the preceding state, carefully update a current state, and then output only important parts of the current state that are required in future states. This solves the vanishing gradient problem in RNNs by updating a state then propagating forward important parts of that state that are pertinent to future states. Thus, LSTMs become far more efficient than RNNs as there is not an extended back-propagation chain often seen in RNNs (Hochreiter and Schmidhuber, 1997).

LSTMs use the input gate, forget gate, and output gate to release information

between the hidden state and the cell state. The structure of an LSTM cell is shown in Figure 6.1, where X_t: input vector, h_t: output of the current net-

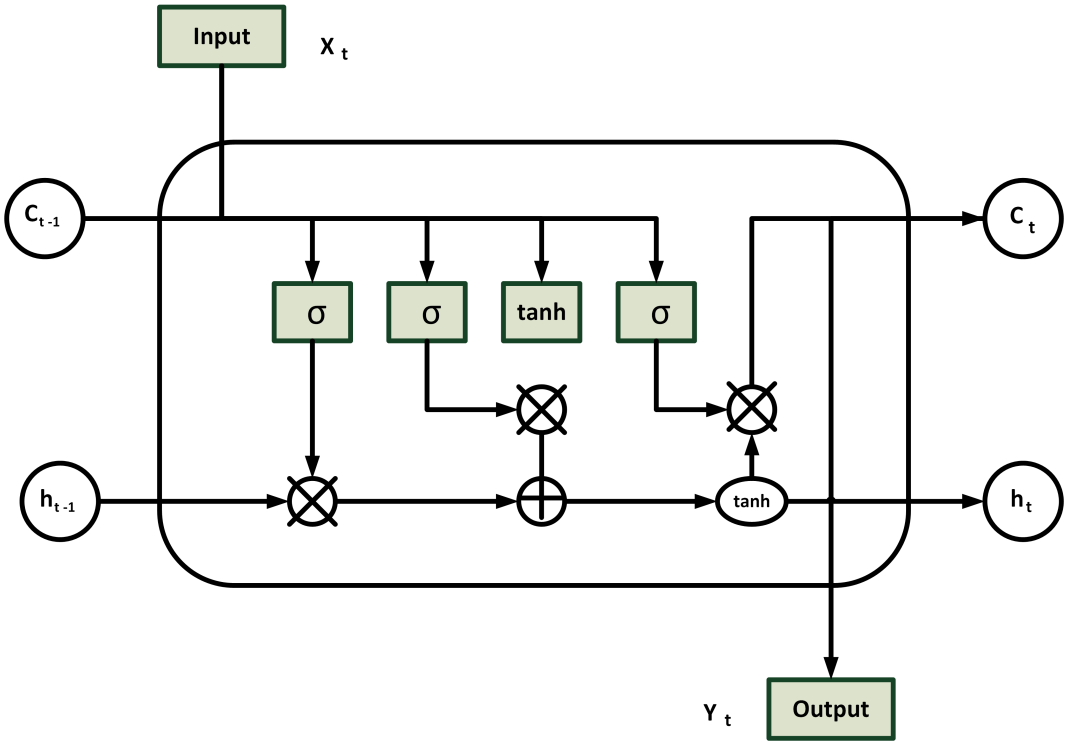


Figure 6.1: Schematic representation of a LSTM cell.

work, h_{t-1}: output from previous LSTM unit, C_{t-1}: a memory of the previous unit, C_t: a memory of the current unit, $\bigotimes$: element-wise multiplication, $\bigoplus$: element-wise summation and tanh: the hyperbolic tangent.

Figure 6.1 shows that an LSTM unit is made up of a cell, with a state C_t over time. The LSTM unit uses the following gates: input I_t, output, O_t and forget, f_t gates for modifying and adding memory in the cell. The flow of information into the cell as well as out of a cell is controlled by these three gates. Also, a cell emits h_t, an output signal after updating a gate. To update h_t, the sigmoid layer of an LSTM cell unit is initialised at the forget gate, f_t. Then, the LSTM cell unit determines the importance of C_{t-1}. Consequently, the sigmoid layer ("input gate layer") chooses the values to update. After that, a vector of new candidate values, $\tilde{C}_t$ is created using the tanh layer. $\tilde{C}_t$ may be appended to the state C_{t-1}, simultaneously, removing or forgetting some values. Moreover, multiplying C_{t-1} by f_t (without the removed or "forgotten values") and then adding $I_t \cdot \tilde{C}_t$ updates C_t. Thus, $I_t \cdot \tilde{C}_t$ is made up of the new candidate values multiplied by the input values of the current state. Lastly, the output

of the LSTM cell is computed by employing the third sigmoid level along with another tanh filter (Chen et al., 2020b). The following equations summarise the process of obtaining the output of the hidden state, h_t (Chen et al., 2020b; Alla and Adari, 2019);

$$f_t = \sigma(\boldsymbol{W}_f[h_{t-1}, x_t] + b_f), \tag{6.1}$$

$$I_t = \sigma(\boldsymbol{W}_i[h_{t-1}, x_t] + b_I), \tag{6.2}$$

$$\tilde{C}_t = \tanh(\boldsymbol{W}_C[h_{t-1}, x_t] + b_C), \tag{6.3}$$

$$C_t = f_t \cdot C_{t-1} + I_t \cdot \tilde{C}_t, \tag{6.4}$$

$$o_t = \sigma(\boldsymbol{W}_i[h_{t-1}, x_t] + b_o), \tag{6.5}$$

$$h_t = o_t \cdot \tanh(C_t). \tag{6.6}$$

$C_0 = 0$ and $h_0 = 0$, indicate initial values, and t represents the time steps. The activation function is represented by, σ. It takes values between 0 to 1, thereby, ensuring that the data is removed completely, partially removed, or preserved. $\tilde{C}_t$ is a "candidate" hidden state. Its values are updated using the current input value and the previous hidden state's value. I_t is an input gate that controls the amount of information from the newly computed current state that is allowed to pass through, h_{t-1} connects the previously hidden layer and the current hidden layer recurrently, $\boldsymbol{W}$ represents the weight matrix that connects the inputs to the current hidden layer, the internal memory of a cell unit is represented by C_t, and the output of a hidden state is given by h_t.

The LSTM neural network uses the activation functions, tanh and sigmoid. Neural networks use these activation functions to learn complex data patterns. They work by converting the output signal from a previous cell into a form that serves as the input to the next cell. Also, they add non-linearity in data to make it similar to real world data or problems (Schilling, 2016; Alla and Adari, 2019). Ideally, tanh is used in situations where signals from historical data points are required because it can sustain information for a longer period before going to zero (Alla and Adari, 2019). Also, Figure 6.1 shows that we need another activation function called the sigmoid function to either forget or recall some of the information.

We use LSTM networks as they are capable of learning long-term dependencies through recurrently connected subnets known as memory blocks (Graves and Schmidhuber, 2005). LSTM networks can learn complex structures within the sequential ordering of sequences. In addition, they utilise internal memory to remember information across long input sequences. Long short-term memory (LSTM) networks are designed to solve the vanishing gradient problem associated with RNNs.

6.2.2.3 Bi-directional long-term memory recurrent neural network (Bi-LSTM)

The LSTM addresses the problem of long-time lags found in RNNs. There are situations where predictions have to be made by looking at both the prior and subsequent inputs. The bidirectional LSTM (Bi-LSTM) proposed by Hochreiter and Schmidhuber (1997) addresses the problem of making predictions based on previous and subsequent inputs. Figure 6.2 shows that the Bi-LSTM has a forward layer that first calculates the network from time $T = 1$ to time $T = t$. The hidden layers' output at each time-step from $T = 1$ to time-step $T = t$ is saved. Then a reverse calculation of the network using a backward layer occurs and the outcome of the hidden layer at each time from time-step t to time-step 1 is calculated and saved (Hu et al., 2019). Chawla et al. (2019) mentions that the outputs of the forward and backward layer are then combined at each time step using one of the following means: (i) Concat: Where the outputs are concatenated together. (ii) Mul: Where the outputs are multiplied together, (iii) Sum: Where the outputs are added and (iv) Ave: Where the average of the two outputs is taken.

We implement concat in our proposed model to merge the outputs from the forward and backyard layers as it is the default method often used in bidirectional LSTMs (Kiperwasser and Goldberg, 2016; Ding et al., 2018; Li et al., 2018b; Sahu and Anand, 2018; Rhanoui et al., 2019; Chawla et al., 2019; Mu and Xu, 2019). In addition, concat doubles the output vector size that serves as input to the next layer (Chawla et al., 2019), and this will result in better performance or a lower log loss. We train our proposed model using the Backpropagation Through Time (BPTT) algorithm (Pascanu et al., 2013) to resolve the problem of the vanishing/exploding gradient.

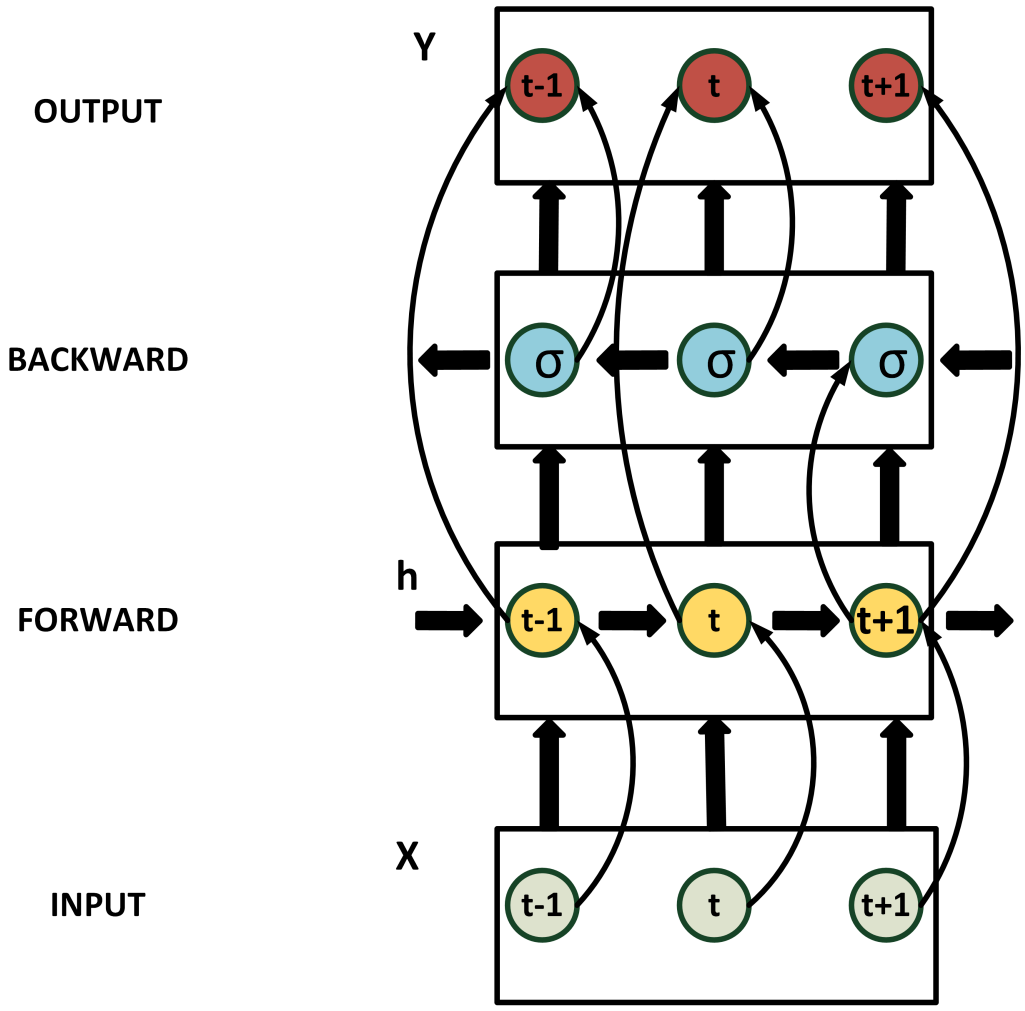


Figure 6.2: Schematic representation of a Bi-LSTM.

6.2.3 Proposed Architecture

Figure 6.3 shows the architecture of the CNN-Bi-LSTM that uses CNN layers as well as max-pooling layers for extracting features from input data, combined with a bi-directional LSTM network for interpreting the features across time steps and also perform sequence prediction. The proposed CNN-Bi-LSTM will consist of three CNN layers, then a Bi-LSTM layer and a dense layer as the output. Also, the architecture includes dropout layers that are deployed to address the problem of over-fitting that is common in deep neural networks (Zhang et al., 2020). Our proposed architecture follows the suggestions made by Nguyen et al. (2016); Johnson and Zhang (2015) in that, we replace the

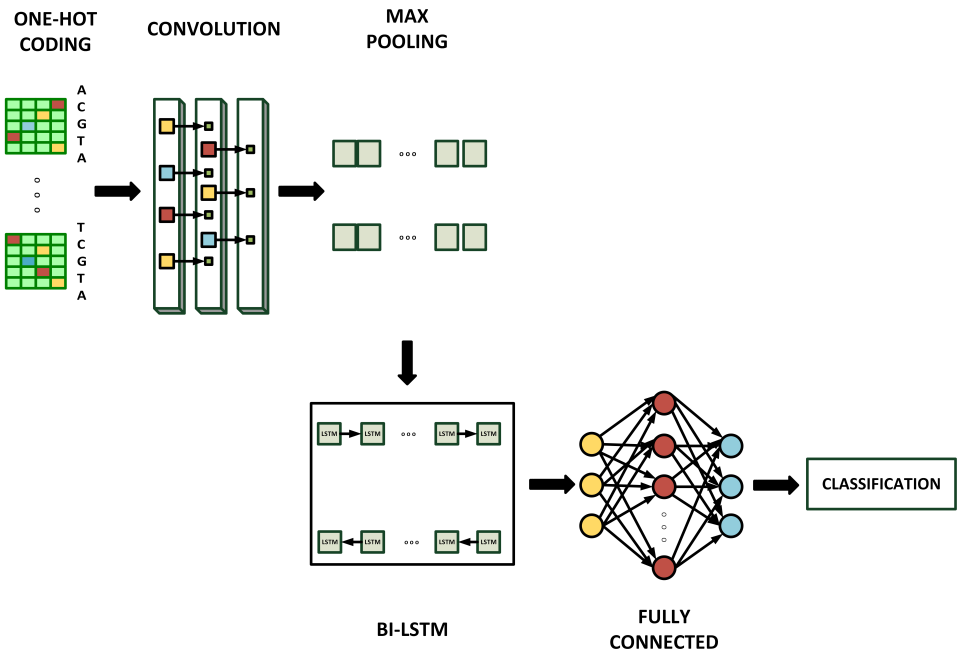


Figure 6.3: Schematic representation of the CNN-Bi-LSTM.

coding/encoding layer and embedding layers by directly applying the CNN to high-dimensional one-hot vectors; i.e., embeddings of text regions are directly learned without going through the word embedding learning process. Also, we utilise one Bi-LSTM layer.

6.2.4 Experiments

We carried out experiments to determine the classification performance of the CNN-Bi-LSTM algorithm on the SARS CoV-2 data set described in Section III. For deep learning methods, pre-processing of data is very important. We created class labels to indicate whether a genome sequence was that of SARS-CoV-2 (positive samples) or not (negative samples). From the NCBI genes database, we obtained 34 positive samples all of which were marked as SARS-CoV-2 gene sequences (Table 6.1). Also, we obtained 295 negative samples, none of which was marked as SARS CoV-2 gene sequences.

We used **Keras** (Chollet and others, 2015) to define the CNN-Bi-LSTM model by first creating the CNN layers, then the Bi-LSTM layers and output layers. The CNN-Bi-LSTM model was trained to classify SARS-CoV-2 virus sequences amongst coronaviruses'; as well as classify whether a virus gene sequence contains SARS CoV-2 regulatory motifs or not. The deep learning models were

trained independently using batch sizes of 64 as recommended by Alipanahi et al. (2015) and Zhang et al. (2020). We used Kera's default weights and biases. The models are trained for 100 epochs using the recommended default learning rate, lr = 0.001 (Kingma and Adam, 2014b; Zhang et al., 2020). We used dropout ratios equal to 0.1, 0.3, and 0.5. Following Zhang et al. (2020), we changed the number of cells in the Bi-LSTM layer from 32 to 256 and set the default number of cells to 32. The number of filters in the CNN layers is changed from 32 to 256 and we used a default value of 32 filters. Additionally, we used the binary log-loss (binary cross-entropy) and the efficient **Adam** (Kingma and Adam, 2014b) optimisation algorithm. The output layer was a fully connected layer with sigmoid as the activation function to perform binary classification (Zhang et al., 2020). Finally, we evaluated the skill of deep learning models. Deep learning algorithms are stochastic and have some additional sources of variation. The additional randomness allows model flexibility during the learning phase. However, this flexibility can make the model be unstable i.e., producing different results when the model is trained on the same data. To mitigate this problem, we carried out 100 iterations of each experiment and then took the average of the evaluation metrics for 100 iterations. Each model was trained for 100 epochs.

6.3 Results

The most commonly used model evaluation metric for binary classification is accuracy which can be misleading when used as the only performance metric in the case where the data is unbalanced. The data for classifying SARS CoV-2 genes was unbalanced with 10.3% positive and 89.7% negative samples. The data set for classifying virus genes with regulatory motifs for the SARS CoV-2 genes was unbalanced with 3.69% positive samples (with regulatory motifs) and 96.31% negative samples. This means that classification may not work well as the classifiers may be biased towards the majority class. Therefore, the deep learning models are evaluated and compared by making use of a confusion matrix and then deriving the following metrics:

(i) Sensitivity (Sens) =

$$\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

(ii) Specificity (Spec) =

$$\frac{\text{TN}}{\text{TN} + \text{FP}}$$

(iii) Precision (Prec) =

$$\frac{\text{TP}}{\text{TP} + \text{FP}}$$

(iv) Accuracy (Acc) =

$$\frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{FP}}$$

(v) Mathew's Correlation Coefficient (MCC)

=

$$\frac{\text{TP} \cdot \text{TN} - \text{FN} \cdot \text{FP}}{\sqrt{(\text{TP} + \text{FN})(\text{TP} + \text{FP})(\text{TN} + \text{FN})(\text{TN} + \text{FP})}}$$

,

Where TP represents the true positives, TN represents the true negatives, FP and FN represent the false positives and false-negatives, respectively. Boughorbel et al. (2017) states that MCC in the interval [-1, 1], with 1 indicating that there is perfect classification, -1 indicating a perfect misclassification.

(vi) Cohen's Kappa (κ): is a robust statistic that can be used to assess the performance of classifiers. Also, Kappa considers a model's accuracy obtained by chance. κ can be calculated using; $\kappa = \frac{O-E}{1-E}$ (Kuhn et al., 2013), where O is the accuracy that is observed and E is the expected accuracy. In this paper, we will use Cohen's Kappa to assess the performances of our algorithms when performing classification tasks. κ is similar to correlation coefficients and takes values from -1 to +1 inclusive; where a value of 0 means that the predicted class and observed class do not agree, while a value of 1 indicates that the observed class and the predicted class agree perfectly (Kuhn et al., 2013). Also, Landis and Koch (1977b) states that κ values less than 0.20 indicate poor agreement, values between 0.20 - 0.40 indicate fair agreement, values between 0.40 - 0.60 indicate moderate agreement whilst substantial agreement starts at a value of 0.61. Excellent examples and explanations on the use of Cohen's Kappa for classification can be found in Kuhn et al.

(2013). In addition, Kuhn (2012) provides a caret R package for computing Cohen's Kappa.

The most commonly used model evaluation metric for binary classification is accuracy which can be misleading when used as the only performance metric in the case where the data is unbalanced. The data for classifying SARS CoV-2 genes was unbalanced with 10.3% positive and 89.7% negative samples. The data set for classifying virus genes with regulatory motifs for the SARS CoV-2 genes was unbalanced with 3.69% positive samples (with regulatory motifs) and 96.31% negative samples. This means that classification may not work well as the classifiers may be biased towards the majority class. Therefore, we will use Cohen's Kappa to evaluate how the actual classes and the classes predicted by the CNN-Bi-LSTM, CNN-LSTM, and CNN models agree.

(vii) No information Rate (NIR) and P-Value [Acc > NIR]. A good model is one where the accuracy is significantly greater than the no information rate. This means that a model with an accuracy that is less than the NIR is poor at classifying imbalanced data as it is just predicting the majority class most of the time. Such a model is said to be unreliable (Amruthnath and Gupta, 2018). In addition, the model is also said to be poor if the rate of the majority class equals the classification accuracy. Therefore, a hypothesis test is carried out to assess if the overall accuracy rate is greater than the rate of the majority class (NIR), i.e., P-Value [Acc > NIR]. A significant P-value [Acc > NIR] indicates that our model is better than just classifying all into the majority class.

In addition to the metrics above, the predictive performance of each deep learning model is assessed using the AUC ROC.

6.3.1 Parameter Analysis

6.3.1.1 Performance comparison using different learning rates

To obtain optimal performance for classifying SARS CoV-2, the hyper-parameters of our deep learning algorithms need to be tuned. The learning rate (lr) is an important hyper-parameter that has to be tuned for the deep learning algorithms to obtain optimal results. Zhang et al. (2020) state that with a lower

lr, the training phase of the deep learning algorithm becomes more reliable. However, a lower lr may come at the expense of taking much time during the optimisation phase as the updated values of the loss function may be small (Zhang et al., 2020). A higher lr may cause the training stage not to converge and it even diverges (Zhang et al., 2020). Also, Zhang et al. (2020) mentions that with a higher learning rate, the optimisation phase may skip the optimal value, and the optimisation phase of the loss function may become even worse. Thus, there is a risk of skipping the optimal value when using a larger learning rate and this may adversely affect the accuracy of the algorithm (Wilson and Martinez, 2001). This is because a larger learning rate requires more training time as it is continually skipping the optimal value and "unlearning" what has already been learned, resulting in unproductive oscillations of the accuracy. These oscillations will cause poor generalisation of the accuracy because the training weights never settle down to give an optimal value (minimum). As recommended by Zhang et al. (2020), we used the (default) learning rate, lr = 0.001 for the Adam algorithm for stochastic optimisation to update the parameters. Moreover, Kingma and Adam (2014b) states that a default lr = 0.001 for the Adam optimiser is a good learning rate for stochastic optimisers.

6.3.1.2 Performance comparison using different dropout ratios

Deep neural networks with many parameters may suffer from the problem of over-fitting. To address this problem, we use the dropout technique described in detail in (Srivastava et al., 2014b). The dropout technique temporarily removes a hidden and or a visible unit together with all its incoming and outgoing connections. The units that are selected to be dropped out are selected at random. In this paper, we investigate the effect of the dropout technique in preventing over-fitting and improving accuracy. We applied dropouts after the convolutional and max-pooling layers as well as in the LSTM cell implementation. Tables 6.2 and 6.3 show that the performance of our proposed model (CNN-Bi-LSTM) is similar and stable for dropout ratios 0.1 and 0.3. However, the performance drops slightly when the dropout ratio is set to 0.5. Probably, this shows that a higher dropout of 0.5 may be resulting in a higher variance to some of the layers, and this has the effect of degrading training and, reducing performance. Thus, at a 0.5 dropout ratio, the capacity of our model is marginally diminished causing the performance of the model to marginally de-

teriorate. Therefore, for the sake of comparison, we specify a dropout ratio of 0.1 for implementation in the CNN, CNN-LSTM, and CNN-Bi-LSTM models.

Table 6.2: A comparison of CNN-BiLSTM's performance with changing dropout ratios.

Dropout ratio	Precision(%)	Specificity(%)	Sensitivity(%)
0.1	99.81	99.97	99.97
0.3	99.81	99.97	99.97
0.5	99.26	98.33	98.33

Table 6.3: A comparison of CNN-BiLSTM's performance with changing dropout ratios.

Dropout ratio	AUC ROC(%)	Acc(%)	MCC	Kappa
0.1	99.81	99.95	0.9782	0.9975
0.3	99.00	99.94	0.9596	0.9775
0.5	99.91	99.9	0.9782	0.9667

6.3.1.3 Performance comparison using different numbers of convolutional filters in CNN

We gradually varied the number of filters or kernels in CNN from 32, 64, 128 to 256. By varying the number of kernels or filters in CNN, we were able to evaluate Sens, Spec, Acc, Prec, MCC, AUC ROC, and Cohen' Kappa values on the training data set. Table 6.4 shows how the evaluation metrics vary under different numbers of convolutional filters. We see that the values of Sens, Spec, Acc, Prec for the CNN-Bi-LSTM model are slightly higher than those of the CNN-LSTM and CNN models. Also, we observe that the AUC ROC values for the CNN-Bi-LSTM model are superior to those of the other models as the number of convolutional filters increases. This indicates that our proposed model outperforms the CNN-LSTM and the CNN models. Specifically, the AUC ROC for the CNN-Bi-LSTM model improves considerably as the number of filters increases from 32 to 128. Table 6.4 shows that when the number of filters is equal to 32, the CNN-Bi-LSTM model performs marginally better than the CNN-LSTM and CNN models in all metrics. For example, when the number of convolutional filters is 32, the values of Sens, Spec, Prec, Acc, AUC

ROC, MCC, and Kappa for our proposed model are 99.97%, 99.97%, 99.91%, 99.95%, 99.81%, 0.9978, and 0.9978, respectively. These results show that the performance of the CNN-Bi-LSTM is comparable to that of the CNN-LSTM model and performs marginally better by gaps of 1.01%, 1.01%, 0.65%, 0.3%, 6.27%, 0.0159%, and 0.0164% respectively. Similarly, our proposed model's performance is comparable to that of the CNN model and performs marginally better by gaps of 1.43%, 1.43%, 0.09%, 0.30%, 8.64%, 0.00%, and 0.024% respectively. Therefore, for the sake of comparison, we use the default 32 cells in the convolutional layers of all three models.

Table 6.4: Performance comparison using different numbers of filters in CNN.

	Cell numbers	CNN-Bi-LSTM	CNN-LSTM	CNN
Sens (%)	32	99.97	98.96	98.54
	64	99.16	96.52	99.38
	128	99.83	97.71	97.91
	256	99.97	97.71	99.33
Spec (%)	32	99.97	98.96	98.54
	64	99.16	96.52	99.38
	128	99.97	97.71	97.91
	256	99.97	97.71	99.33
Prec(%)	32	99.91	99.26	99.82
	64	99.92	99.26	99.92
	128	99.83	99.26	100.0
	256	99.82	99.26	99.95
Acc (%)	32	99.95	99.65	99.65
	64	99.85	99.44	99.84
	128	99.95	98.19	99.39
	256	99.95	99.74	99.89
AUC ROC (%)	32	99.81	93.54	91.17
	64	100.0	91.80	96.46
	128	100.0	93.91	94.67
	256	99.52	92.21	94.75
MCC	32	0.9978	0.9819	0.9778
	64	0.9782	0.9819	0.9964
	128	0.9782	0.9819	1.000
	256	0.9978	0.9819	0.9921
Cohen's Kappa	32	0.9978	0.9814	0.9734
	64	0.9882	0.9380	0.9915
	128	0.988	0.9582	0.9614
	256	0.9978	0.9582	0.9912

6.3.1.4 Performance comparison using different numbers of cells in LSTM

We carried out experiments with different numbers of cells in the LSTM part of the model to choose the optimal number of cells that improves the performances of the deep learning algorithms. By varying the numbers of cells from 32, 64, 128 to 256, we were able to evaluate Sens, Spec, Prec, Acc , MCC, AUC ROC, NIR and Cohen' Kappa values on the training data set. Table 6.5 shows the performances of the CNN-Bi-LSTM and CNN-LSTM with a different number of cells in the LSTM. The results show that Sens, Spec, Prec and Acc for our proposed model are generally higher than those of the CNN-LSTM model. The AUC ROC of our proposed model significantly increases when the number of cells changes from 32 to 128 and then stabilises when the number of cells is 256. Furthermore, Table 6.5 shows that the best performing number of cells in the LSTM is 32. The values of Sens, Spec, Prec, Acc, AUC ROC, MCC, and Kappa for the CNN-Bi-LSTM model when the number of cells is 32 are: 99.97%, 99.97%, 99.81%, 99.95%, 99.81%, 0.9978, and 0.9978, respectively. These values show that our proposed model outperforms the CNN-LSTM model by gaps of 1.01%, 1.01%, 0.55%, 0.3%, 6.27%, 0.0159, and 0.0164 respectively. Therefore, for the sake of comparison, we use the default 32 cells in the LSTM layers.

6.3.1.5 Model training time

We also consider the cost in terms of the time each model takes to train for 100 epochs, i.e., the time it takes to complete 100 training epochs as shown in Table 6.6.

Table 6.6 shows that adding a Bi-LSTM layer after the CNN layers results in the proposed model taking much more time to train for 100 epochs than the CNN-LSTM and CNN models. Moreover, the results show that the additional time taken by CNN-Bi-LSTM offers marginally better performance than the CNN-LSTM and CNN models because the Bi-LSTM layer has additional training capabilities (Siami-Namini et al., 2019).

Table 6.5: Performance comparison using different numbers of cells in LSTM.

	Cell numbers	CNN-Bi-LSTM	CNN-LSTM
Sens (%)	32	99.97	98.96
	64	99.94	99.56
	128	97.96	99.28
	256	99.92	99.20
Spec (%)	32	99.97	98.96
	64	99.94	99.56
	128	97.96	99.26
	256	99.92	99.2
Prec(%)	32	99.81	99.26
	64	99.63	99.74
	128	99.58	99.61
	256	99.44	99.07
Acc (%)	32	99.95	99.65
	64	99.90	99.90
	128	99.49	99.80
	256	99.85	99.70
AUC ROC (%)	32	99.81	93.54
	64	99.91	93.59
	128	1.000	93.62
	256	99.99	91.84
MCC	32	0.9978	0.9819
	64	0.9956	0.9926
	128	0.9691	0.9880
	256	0.9932	0.9811
Cohen' Kappa	32	0.9978	0.9814
	64	0.9955	0.9921
	128	0.9639	0.9869
	256	0.9925	0.9791

6.3.2 Performance comparison

6.3.2.1 Performance comparison of CNN-Bi-LSTM, CNN-LSTM and CNN models

Using the results from Table 6.4, we evaluated the peak performances of the three models. Table 6.7 displays the *peak* performance comparisons of the three models when they are used to classify SARS CoV-2 virus amongst coronaviruses. Our proposed model is comparable and achieves similar performances to those of the other models in almost all the evaluation metrics. The results show that the CNN-Bi-LSTM achieves 99.97%, 99.97%, 99.92%,

Table 6.6: Model total training time for 100 epochs.

Models	TrainableParameters	Training epochs	Training time(s)
CNN-Bi-LSTM	27 892	100	166.33
CNN-LSTM	17 268	100	98.1442
CNN	31 394	100	71.4451

99.95%, 100.0%, and 0.9978 for Sens, Spec, Prec, Acc, AUC ROC, and Cohen's Kappa, respectively. These values show that at the peak, our proposed model's performance is marginally higher than that of the CNN-LSTM model by gaps of 1.01%, 1.01%, 0.66%, 0.21%, 6.09%, and 0.0063 respectively for Sens, Spec, Prec, Acc, AUC ROC, and Cohen's Kappa. Similarly, our proposed model's performance is marginally higher than that of the CNN model by gaps of 0.59%, 0.59%, 0.06%, 0.54%, and 0.0063 respectively for Sens, Spec, Acc, AUC ROC, and Cohen's Kappa. These results show that the CNN-Bi-LSTM that combines the CNN and Bi-LSTM layers marginally improves performance compared to the other models. Furthermore, these results demonstrate the added advantage of using the Bi-LSTM layer which incorporates both previous input values and future input values.

Table 6.7: Peak performance comparisons in the classification of SARS CoV-2 amongst coronaviruses.

	CNN-Bi-LSTM	CNN-LSTM	CNN
Sens (%)	99.97	99.86	99.38
Spec (%)	99.97	99.86	99.38
Prec(%)	99.92	99.26	100
Acc (%)	99.95	99.74	99.89
AUC ROC (%)	100.00	93.91	99.46
MCC	0.9978	0.9819	1.000
Cohen's Kappa	0.9978	0.9814	0.9915

6.3.2.2 Approximate Statistical Tests for Comparing the CNN-Bi-LSTM, CNN-LSTM, and CNN models

Table 6.7 shows that the peak performances of our proposed model are comparable and in some cases marginally higher than those of the CNN-LSTM and CNN models. However, there is a need to perform hypothesis tests that can spot any differences better than the human eye to examine if the dif-

ferences in the performance of the models are statistically significant. Thus, we applied the post-hoc 5×2-fold cv paired t-test as opposed to the k-fold cross-validated paired t-test (Dietterich, 1998) to test for the differences in performance relative to the AUC ROC. The k-fold cross-validation is widely used to evaluate the performance of different models by computing and directly comparing different performance metrics (Berrar, 2019). However, in the k-fold cross-validated paired t-test, the training data sets may overlap. For example, in 10-fold cross-validation, each pair of the training data sets shares 80% of the data examples. This presents a problem as the overlap may prevent the paired t-test from obtaining good estimates of the amount of the variation that would have been accounted for had the training data sets been entirely independent of the other previous training data sets (Dietterich, 1998). Also, Dietterich (1998), mentions that the 10-fold cross-validation technique shows higher probabilities of type 1 errors. To solve the problem where the training data sets may overlap, Dietterich (1998) recommended using a 5×2-fold cv paired t-test which is based on repeating two-fold cross-validations five times. The two-fold cross-validation is used because it yields larger test data sets as well as training data sets that are disjoint. The 5×2-fold cv paired t-test is a more powerful test than the k-fold cross-validated paired t-test as it directly measures variation that is due to the choice of the training data set. Thus, we use the 5×2-fold cv paired t-test to perform a post-hoc analysis to determine the statistical significance of the differences in the means of the performance metric scores. Following Zhang et al. (2020); Livieris et al. (287), we chose the AUC ROC as a specific measure to choose the model that would be more accurate on new test data. The test statistic $\tilde{t}$, for the 5×2-fold cv paired t-test is calculated using the following equation (Dietterich, 1998)

$$\tilde{t} = \frac{p_1^{(1)}}{\sqrt{\frac{1}{5}\sum_{i=1}^{5} s_i^2}}, \tag{6.7}$$

where $p_1^{(1)}$ is the difference in the AUC ROC scores of the CNN-Bi-LSTM vs CNN or CNN-LSTM models for the first fold of the first iteration, s_i^2 is the variance of the AUC ROC score differences of the ith iteration. The variance is computed using; $\left(p_i^{(1)} - \bar{p_i}\right)^2 + \left(p_i^{(2)} - \bar{p_i}\right)^2$. In addition, $p_i^{(j)}$ is the difference in the AUC ROC scores of the CNN-Bi-LSTM vs CNN or CNN-LSTM models for the ith iteration and fold j and $\bar{p_i} = \left(p_i^{(1)} + p_i^{(2)}\right)/2$.

Under H_0, $\tilde{t}$ approximately follows a t distribution with 5 degrees of freedom. We let H_0, be such that there is no statistically significant difference between the AUC ROC of the CNN-Bi-LSTM vs CNN or CNN-LSTM models and H_1, the alternative hypothesis, such that there is a statistically significant difference between the AUC ROC of the CNN-Bi-LSTM vs CNN or CNN-LSTM models. Accepting the null hypothesis, H_0, for a given level of significance would mean that the differences in the estimated performance metrics are due to chance. However, if H_0 is rejected, we conclude that any differences in the performance metrics are due to the differences in the models.

Table 6.8 shows the post-hoc statistical analysis, using the *5 × 2*-fold cv paired t-test relative to the AUC ROC performance metric for the CNN-Bi-LSTM versus the CNN models. The 5 × 2 cv Paired t-test from Table 6.8 produced a t-value = 3.877. This t-value is assumed to follow a t-distribution with 5 degrees of freedom. Thus, the critical value, $t_{5,0.975} = 2.571$. Since t value = $3.877 > t_{5,0.975} = 2.571$, we conclude that the differences in the AUC ROC scores are due to the differences in the performance of the CNN-Bi-LSTM and CNN models. Thus, the CNN-Bi-LSTM outperforms the CNN model relative to the AUC ROC.

Table 6.8: 5 × 2 cv Paired t-test for the CNN-Bi-LSTM and the CNN Models Relative to the AUC ROC.

Folds	**CNN-Bi-LSTM Scores**	**CNN Scores**	**Scores differences**
Fold 1	98.89	93.06	5.83
Fold 2	100	98.89	1.11
Fold 1	100	99.42	0.58
Fold 2	100	99.83	0.17
Fold 1	100	100	0
Fold 2	100	100	0
Fold 1	100	100	0
Fold 2	100	100	0
Fold 1	100	100	0
Fold 2	100	100	0
Mean	**99.89**	**99.01**	**0.869**
stdev	**0.333**	**2.038**	**1.745**

Table 6.9 shows the post-hoc statistical analysis, using the *5 × 2*-fold cv paired t-test relative to the AUC ROC performance metric for the CNN-Bi-LSTM versus the CNN-LSTM models. The 5 × 2 cv Paired t-test from Table 6.9

produced a t-value = 3.654. The critical value, $t_{5,0.975} = 2.571$. Since t value = $3.654 > t_{5,0.975} = 2.571$, we conclude that the differences in the AUC ROC scores are statistically significant and are due to the differences in performance of the CNN-Bi-LSTM and CNN-LSTM models. The results show that relative to the AUC ROC, the CNN-Bi-LSTM performs better than the CNN-LSTM.

Table 6.9: 5 × 2 cv Paired t-test for the CNN-Bi-LSTM and the CNN-LSTM Models Relative to the AUC ROC.

Folds	**CNN-Bi-LSTM Scores**	**CNN-LSTM Scores**	**Scores differences**
Fold 1	75.83	74.44	1.39
Fold 2	100	98.89	1.11
Fold 1	99.42	98.26	1.16
Fold 2	100	100	0
Fold 1	100	100	0
Fold 2	100	100	0
Fold 1	100	100	0
Fold 2	100	100	0
Fold 1	100	100	0
Fold 2	100	100	0
Mean	**97.52**	**97.16**	**0.367**
stdev	**7.233**	**7.594**	**0.564**

6.3.2.3 Performance comparison of the CNN-Bi-LSTM with different data sets

To evaluate the performance of the proposed CNN-Bi-LSTM model on new data, we conducted experiments using different data sets with 25%, 50%, 75%, and 100% of the data set with regulatory motifs for the SARS CoV-2 gene sequences obtained from the NCBI database. Table 6.10 shows the genes with regulatory motifs for the SARS CoV-2 discovered by (Dinka and Milkesa, 2020).

Dinka and Milkesa (2020) analysed whether the following eleven genes had regulatory motifs for SARS-CoV-2 virus: orf1ab/43740578, orf8/43740577, orf10/43740576, N/43740575, orf7b/43740574, orf7a/43740573, orf6/43740572,

Table 6.10: Genes with Regulatory motifs for the SARS CoV-2.

Name /Gene ID	**Description**
orf8/43740577	orf8 protein
orf10/43740576	orf10 protein
N/43740575	Nucleocapsid phosphoprotein
orf7b/43740574	orf7b protein
orf7a/43740573	orf7a protein
orf6/43740572	orf6 protein
M/43740571	Membrane glycoprotein
E/43740570	Envelope protein
orf3a/43740569	orf3a protein
S/43740568	Surface glycoprotein

M/43740571, E/43740570, orf3a/43740569 and S/43740568, using MEME (Bailey et al., 1994). The searches were done to identify common candidate regulatory motifs that serve as positions where transcription factors (TFs) can bind to. In turn, TFs control the expression of the SARS CoV-2 genes (Dinka and Milkesa, 2020). The authors found out that ten of these genes except the orf1ab/43740578 gene had DNA sequences that were responsible for turning on/off the SARS CoV-2 genes. All the genes that contained the regulatory motifs for the SARS CoV-2 were attached to label 1. Also, the gene orf1ab/43740578 is present in SARS CoV-2 genes but it was attached to the label 0 as it does not have regulatory motifs for the SARS CoV-2 genes (Dinka and Milkesa, 2020). Also, all other genes from the *Coronaviridae* family that do not contain regulatory motifs for the SARS CoV-2 genes were attached to the label 0.

The data for classifying whether a virus gene contains regulatory motifs for the SARS CoV-2 genes was organised and summarised as shown in Table 6.11.

Table 6.11 shows that the data set is unbalanced with 3.69% positive samples (with regulatory motifs) and 96.31% negative samples. We used 80% of the data set for training and 20% for testing. Based on the experimental results in Section 5.3, we extracted the parameters shown in Table 6.12. With these parameter settings, we performed experiments using the different fractions of the data set to evaluate the performance of the CNN-Bi-LSTM.

Table 6.11: Data for classifying whether a virus gene contains regulatory motifs for the SARS CoV-2 genes.

Virus gene	**Class Label**	**Number of Samples**
With regulatory motifs	1	76
Without regulatory motifs	0	1982

Table 6.12: Optimum parameter settings for the CNN-Bi-LSTM, CNN-LSTM and CNN models.

Parameter	CNN-Bi-LSTM	CNN-LSTM	CNN
Learning rate	0.001	0.001	0.001
Dropout ratio	0.1	0.1	0.1
Number of Kernels	32	32	32
Number of Cells	32	32	-
Epochs	100	100	100
Batch size	64	64	64
Number of Iterations	100	100	100

Table 6.13 shows that the performance of the CNN-Bi-LSTM remains excellent when applied to a new data set. The new data set is used to classify whether a virus gene contains regulatory motifs for the SARS CoV-2 genes or not. Additionally, we find out that as the cardinality of the data increases, the AUC ROC increases. This shows that our model's performance improves with more data. At 100% the size of our data set, there is more training data that the CNN-Bi-LSTM effectively uses to improve its performance.

6.3.3 Identifying Nucleotides in Regulatory Motifs for the SARS CoV-2 genes using Saliency Maps

In this paper, we use the saliency map to show which bases in a virus gene sequence are important for predicting whether the sequence contains regulatory motifs for the SARS CoV-2 virus gene or not. Moreover, the map shows the gradient of the model's prediction for each nucleotide. This means that the saliency map shows the changes in the output response value (i.e., whether a sequence contains regulatory motifs or not) concerning small changes in the input nucleotide sequence (Zou et al., 2019). The gradients can be positive or negative and all the positive values in the gradients tell us that a small change

Table 6.13: Performance of the CNN-Bi-LSTM for classifying whether a virus gene contains regulatory motifs for the SARS CoV-2 genes or not.

	Sample size (%)	CNN-Bi-LSTM
Sens(%)	25	99.76
	50	99.79
	75	99.99
	100	99.76
Spec(%)	25	99.76
	50	99.79
	75	99.99
	100	99.76
Prec(%)	25	99.99
	50	99.79
	75	99.99
	100	99.76
Acc(%)	25	99.71
	50	99.98
	75	99.99
	100	99.98
AUC ROC(%)	25	98.9
	50	99.85
	75	100.00
	100	100.00
MCC	25	0.998
	50	0.998
	75	0.999
	100	0.998
Cohen's Kappa	25	0.997
	50	0.998
	75	0.999
	100	0.997

to that nucleotide will change the output value.

Using our best performing model (CNN-Bi-LSTM model), the saliency map shown in Figure 6.4 shows the bases that have high magnitudes of saliency values. Bases with high saliency values are important for predicting the sequence contains regulatory motifs for the SARS CoV-2 virus or not. The saliency map has therefore revealed nucleotides that are responsible for predicting whether a virus gene has regulatory motifs for the SARS CoV-2 virus gene.

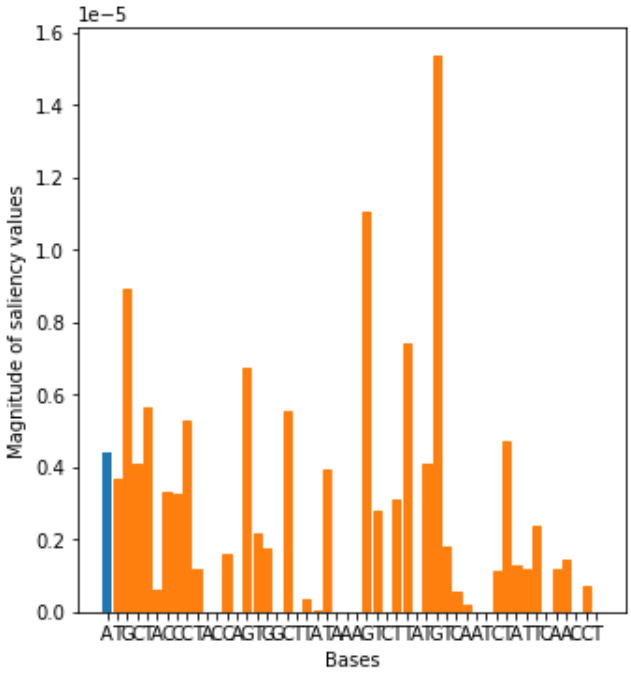


Figure 6.4: Saliency map for bases in one of the positive samples (orange indicates the actual bases in motif.)

6.4 Discussion

The main findings from the performance evaluations of the deep learning models are: 1) at peak, the CNN-Bi-LSTM achieves performance scores for Sens, Spec, Prec, Acc, AUC ROC that are comparable to those of the CNN and CNN-LSTM models; 2) the CNN-Bi-LSTM, CNN-LSTM and CNN models produced high scores on the more reliable statistical measures, the MCC and Cohen's Kappa, which are used to measure the quality of binary (two-class) classifications. The high MCC and Cohen's Kappa values show that all these models are useful for binary classification, an indication that the models obtained excellent results in all of the four confusion matrix categories (true positives, false-negatives, true negatives, and false positives); 3) our proposed model, the CNN-Bi-LSTM can classify the SARS CoV-2 virus, which is very similar to other viruses in the Coronaviridae family; 4) the 5×2-fold cv paired t-tests shows that at peak, the CNN-Bi-LSTM achieves an AUC ROC of 100% which is significantly higher than that of the CNN and CNN-LSTM models. Consequently, the proposed CNN-Bi-LSTM model achieves good binary clas-

sification results; 5) the P-value [Acc > NIR] for CNN-Bi-LSTM (2.2e-16 < 0.05), CNN-LSTM (2.2e-16 < 0.05) and CNN (2.2e-16 < 0.05) were all significant at a 5% level of significance. These results show that the classification accuracy is significantly greater (at 5% level of significance) than the NIR. This means that the deep learning models are useful for predicting 1s (positive samples) and 0s (negative samples) even when using unbalanced data. We used the P-value [Acc > NIR] because the accuracy may not be sufficient as a measure of performance especially in our case where the data sets are imbalanced.

The primary goal of this paper was to develop a classifier (CNN-Bi-LSTM) that could efficiently distinguish between SARS-CoV-2 gene sequences from non-SARS CoV-2 gene sequences and then compare its classification performance to that of the CNN and CNN-LSTM classifiers. Based on experimental results and the 5×2-fold cv paired t-test, the CNN-Bi-LSTM outperformed the CNN-LSTM and CNN models in classifying SARS CoV-2 gene sequences relative to the AUC ROC. The AUC ROC is a better measure for differentiating between classes. For example, if AUC ROC = 1, then a classifier is able to perfectly distinguish between all the SARS CoV-2 gene sequences and non-SARS CoV-2 gene sequences. The differences in performance between the CNN-Bi-LSTM and the other models is statistically significant at 5% level of significance as shown by the 5×2-fold cv paired t-tests in Tables 6.8 and 6.9. This shows that the CNN-Bi-LSTM model can be used as an alternative model to the CNN and CNN-LSTM. The CNN-Bi-LSTM model takes advantage of the ability of the CNN layers to extract as many features as possible from the DNA sequences. In addition, the model uses the Bi-LSTM layers to learn past and future states in making predictions as well as using the temporal features present in DNA sequences. The Bi-LSTM can keep the chronological order between data, which is very important when analysing long DNA sequences. Thus, by combining these two models into a CNN-Bi-LSTM, we have created a model that takes advantage of the power of the CNN in capturing features that are then used as the input for the Bi-LSTM layers. Therefore, we have developed a hybrid model that meets the objective of efficiently classifying SARS-CoV-2 among coronaviruses. The CNN-Bi-LTSM model consists of three convolutional layers followed by max-pooling layers and a single Bi-LSTM layer as well as a fully connected dense layer fully con-

nected neural network layer which contains 100 neurons for classification. The convolutional layers had 32 kernels and the Bi-LSTM had 32 cells. The results of Tables 6.4 and 6.5 show that increasing further the number of kernels in the CNN and the number of cells in the Bi-LSTM was not beneficial as there were no significant improvements in the performance of the proposed model. Based on the findings by Zeng et al. (2016), we used three convolutional layers because using additional layers of convolution and max-pooling may make the neural network harder to train because it is now "deeper". Nguyen et al. (2016) used two convolutional layers followed by max-pooling when classifying DNA sequences using the CNN model. Table 6.6 shows that the training time for 100 epochs also increases with model complexity, the CNN-Bi-LSTM has an additional bi-directional layer that uses information from past and future states simultaneously, thus, it can understand the context better. Also, 6.6 shows that the overall number of parameters for the CNN model is greater than that of the CNN-Bi-LSTM and CNN-LSTM models. The CNN model contains 31 394 trainable parameters, and the CNN-Bi-LSTM contains 27 892 trainable parameters. The CNN has 12.56% more parameters. This difference in the number of trainable parameters is a result of differences in the size of the dense layer of the two models. The dense layer of CNN models is connected to all the values of the preceding layer and will require a larger weight matrix to parametrise the connection. Conversely, the feature map is processed sample by sample by the CNN-Bi-LSTM model using the recurrent Bi-LSTM part of the model. Therefore, the CNN-Bi-LSTM will require a much-reduced number of parameter values. We note that even though the CNN-Bi-LSTM is a complex model compared to the CNN model, it has fewer parameters. This has implications on the computational resources required when using the CNN-Bi-LSTM model.

We included in the CNN part of the model 1D max-pooling layers but in practice, this is not always the case as reported by Sainath et al. (2015). We used the max-pooling layers to reduce the number of parameters that the models need to learn and thus reduce the training time required. Therefore, the max-pooling layer performs a down-sampling of sequential data via the 1D max-pooling operation. In this paper, we focused more on optimising hyperparameters that influence the network architectures such as the number of kernels in CNN layers as well as the number of cells in the LSTM layers,

that have an impact on performance. We observed that those parameters such as the learning rate and the dropout technique had less effect on performance. For example, we used drop-out rates equal to 0.1, 0.3, and 0.5 yielding little difference in terms of performance. Also, this finding is supported by Zeng et al. (2014); Ordóñez and Roggen (2016).

Additionally, we demonstrated that our proposed model was robust enough when applied to new data (data sets for classifying whether a gene sequence contains regulatory motifs for the SARS CoV-2). Table 6.13 shows the performance of the CNN-BILSTM model when applied to data sets of increasing cardinality. As the cardinality of the data sets increased, there were no significant improvements in performance. This shows the robustness of our proposed model as it is capable of obtaining a very good performance even with relatively small data sets. This finding seems to indicate that although deep learning techniques are often employed with large amounts of data, they may be applied in situations where obtaining large and labelled data sets may be costly.

6.5 Limitations of the study and future work

Deep learning models require more time to train. This is because they have a large number of parameters that need to be trained. Well-trained models are often computationally demanding and they also require large memory. Thus, the deployment of deep learning models can be hampered by computational and memory requirements in cases where there is limited computational power. Thus, in this paper, we could not develop "deeper" architectures as they require more computational resources. Another limitation of our deep learning approach is that the models do not offer easily available explanations on how SARS CoV-2 gene sequences are classified in a particular way, compared to the alignment-based methods. Thus, we used deep learning models more as "black boxes" without providing an explainable justification for their classification results. Additionally, our deep learning models require a large set of training data, as opposed to alignment-based methods that can work even with one genome sequence per class. Thus, deep learning models require several examples per training class. Despite these limitations, the deep learning methods were able to correctly classify SARS Cov-2 amongst coronaviruses and also classify whether a sequence contains regulatory motifs for the SARS CoV-2 or

not.

For future work, we may evaluate the effect of increasing the number of both convolutional and Bi-LSTM layers subject to the availability of computational resources to find a trade-off between how a model performs versus training time. Still, for future work, we will also recommend investigating the causal effect of changes in the composition of the regulatory motifs. In addition, we recommend the use of our proposed model to classify other viral genes as well as explore RNA-protein binding predictions.

6.6 Conclusions

When there is a viral disease outbreak such as that of COVID-19, there is a need for an understanding of the virus's genomic sequence to swiftly act towards containing the virus, treating those that are affected by the virus, and developing vaccines that help to disrupt the spread of the virus. Current tools that are used to detect the virus such as the molecular technique and RT-PCR require support from newer and faster deep learning methods. Thus, it is vital to develop diagnostic tools capable of reliably identifying the SARS CoV-2 virus and then distinguishing it from other coronaviruses or pathogens. These newer methods help in improving the detection rate. Since the SARS CoV-2 is very similar to other coronaviruses, the other coronaviruses can exhibit respiratory infections that are the same as those of SARS CoV-2. Consequently, the identification of the SARS CoV-2 becomes a challenge. It is, therefore, essential to carry out similarity comparisons that can timeously differentiate a novel virus such as SARS CoV-2 from other viruses that are comparable. The similarity comparisons of the SARS CoV-2 virus with other similar and known viruses are crucial in distinguishing whether a DNA sequence is that of SARS-CoV-2 or not. Traditionally, alignment-based methods such as BLAST can be time-consuming. These methods can face challenges when comparing large numbers of sequences that have significant differences in their composition. The advantages of using alignment-free approaches are that they have a quick turn-around in producing desired results and they can simultaneously handle a substantial number of sequences at the same time.

In this paper, we were able to easily compare short sequences of genes with different compositions that were coming from different regions of a complete

genome sequence. For example, the **orf1ab** virus gene from SARS CoV-2 was labelled as a negative sample even though it came from the same sequence (SARS CoV-2 virus complete genome sequence) as other positive sequences that came from the same SARS CoV-2 gene sequence.

We combined a CNN and Bi-LSTM to classify SARS CoV-2 genes from other coronaviruses as well as classify whether a genome sequence contains regulatory motifs that serve as binding sites of transcription factors that regulate the expression of SARS CoV-2 genes. In addition, correct classification is important in discovering different species of coronaviruses, which may affect people in the future. In addition, the SARS CoV-2 virus gene is highly transmissible, hence the proper identification of the SARS CoV-2 is very important in the management of the spread of the virus. Our experimental results using the SARS CoV-2 data sets have shown that the CNN-Bi-LSTM has outperformed the CNN and CNN-LSTM and it can be applied to identify accurately SARS CoV-2 gene virus amongst coronaviruses. The CNN-Bi-LSTM can effectively and efficiently classify DNA sequences data sets of varying cardinalities that it had not seen before. Our proposed model, the CNN-Bi-LSTM outperformed the CNN and CNN-LSTM in detecting whether a virus gene contains regulatory motifs for the SARS CoV-2 virus. Using saliency maps we were able to identify the nucleotides or bases that are important in predicting whether a given gene sequence contains regulatory motifs for the SARS CoV-2 or not. By identifying candidate regulatory motifs together with the bases that predict whether a given sequence is that of SARS CoV-2 or not, it enables scientists to understand the virus's regulation mechanism(s) of gene expression.

CHAPTER 7

Discussion of the Research Papers' Contributions

7.1 Introduction

This chapter outlines the fundamental contributions that were made by this research. Also, the chapter summarises the contributions of each research objective (chapter) to the evaluation of causal inference. The research was conducted to explore how machine learning and statistics can collaborate to tackle the problem of causal inference. Machine learning algorithms have performed better than humans when faced with complex tasks such as voice generation and recognition (Padmanabhan and Johnson Premkumar, 2015), video games (Galway et al., 2008), image, and object recognition (Leonard, 2019). However, evaluating causal inference remains a challenge. Deep learning algorithms such as deep neural networks are especially good at uncovering some hidden patterns in large data sets, and they accurately examine x-ray and MRI scans to identify cancerous cells (Gulum et al., 2021) or label a large number of video frames and images per second (Yan et al., 2020). However, they struggle when it comes to making simple causal inferences.

Causal inference is a statistical tool that can be used by machine learning and artificial intelligence (AI) to measure the causal effects of multiple variables. This research was carried out to show researchers that it is very crucial to start incorporating causal inference into machine learning systems and not to just focus on predicting outcomes. We note that randomized controlled tri-

als (RCTs) have been the "gold" standard in inferring causal effect. RCTs, divide a population of objects/individuals into treatment and control groups, and then administer the treatment to one group and a placebo to the other. The outcomes of the two groups are then measured and compared to give the effect of treatment, assuming that the treatment and control groups are not too different. Thus, inferences of the effectiveness of the treatment are based on the differences in outcomes between the two groups. However, such experiments may not be feasible, especially when ethical issues must be addressed when dealing with, for example, observational data. A characteristic of observational data is that the cause and effect are very hard to recognise. Another characteristic is that such data may be confidential. In this thesis, we have adopted a popular approach to inferring causes from observational data called the potential outcomes framework (Rubin, 2005).

The potential outcomes framework assumes that there are no additional causes In addition the ones we are considering. We have used the potential outcomes framework to provide a way of quantifying the causal effects, i.e., a causal estimands. By using the potential outcomes framework, we have formally articulated the assumptions under which the causal estimands can be estimated on average for a given population. Thus, the potential outcomes framework provides the mathematical link between the data and causal effect estimands. This study has also highlighted the issue of confounding bias, which poses one of the primary challenges when estimating the effect of treatment using data from, for example, an observational study. As a result, we have used propensity scores to reduce confounding bias, and also make a valuable contribution by exploring the applicability of a propensity scores-potential outcomes framework to deep learning algorithms. Thus, by using propensity scores in deep learning models, we have contributed in overcoming some of the challenges that are encountered when evaluating causal inference using machine learning methods. Specifically, the research has shown that the propensity scores and the potential outcomes frameworks can be used to address one of the problems that machine learning methods face, i.e., most real-world data are not generated in the same way as the data that we normally use to train these models. This means that machine learning algorithms are often not robust enough to handle changes in the input data type, and cannot always generalize well. By contrast, causal inference explicitly overcomes this problem by using the po-

tential outcomes/counterfactual framework by considering what might have happened when faced with a lack of information. Consequently, causal inference can then make machine learning models more robust and generalisable.

The contributions of each specific research paper/chapter are outlined in the following sections

7.2 Evaluating uses of Deep Learning Methods for Causal Inference

Objective 1 (Paper 1): To investigate whether or not deep learning methods can be used to estimate propensity scores, which are then used to statistically assess covariate balance and evaluate causal effects. In addition, the paper evaluates the performance of logistic regression and deep learning algorithms in reducing bias and standard errors of the causal effects.

Chapter 3 assessed the performance of logistic regression and deep neural algorithms in reducing bias and standard errors of the treatment effects. Furthermore, statistical methods were employed to assess covariate balance. The research conducted has shown that statistics and machine learning can complement each other in evaluating causal inference. The chapter contributed to addressing a deficiency of adequate literature available on the collaborations of statistics and "cutting edge" deep learning methods in evaluating causal inference. Chapter 3 contributes to addressing this deficiency by providing some ideas on how one can combine statistical propensity score methods with deep learning algorithms to evaluate causal inference. Furthermore, research has shown that deep learning algorithms that require less functional form assumptions and computational time can be used to estimate propensity scores. However, when using propensity scores, it is important to achieve covariate balance. For example, in an observational study, the treated and untreated groups are not directly comparable because they may systematically differ at baseline. The propensity score thus plays an important role in balancing the study groups to make them comparable. This "balancing property" means that, if we control for the propensity score when comparing the groups, we have effectively turned the observational study into a randomized block experiment, where "blocks" are groups of subjects with the same propensities.

Traditionally, logistic regression has been used to estimate propensity scores, and a majority of published propensity score analyses use logistic regression to estimate the scores (Westreich et al., 2010). This is because logistic regression is preferable for probability predictions because it produces probabilities that are in the [0, 1] range. Furthermore, logistic regression is easy to implement in most statistical packages such as STATA, SAS, and R. However, the main focus of the research was to explore how a deep learning/ statistical learning algorithms can be used to estimate propensity scores. The results in Chapter 3 have successfully demonstrated that a deep learning algorithm such as the deep neural network can be adapted and used for the classification tasks with promising results. Thus, we have treated the estimation of propensity scores using the deep neural network as a classification task. Deep neural networks have the advantage that they can handle high-dimensional data better than logistic regression. In addition, a deep neural network with the optimum degree of complexity can approximate any smooth polynomial function, regardless of the order of the polynomial or the number of interaction terms. This means that a researcher does not have to worry *a priori* about having the correct functional form.

The propensity scores were estimated by feeding the inputs, which are the covariates $\boldsymbol{X}$ and the outcome $\boldsymbol{Y}$, across all units into the deep neural network. Instead of predicting the class output values, 0 or 1, the last layer of the deep neural network was modified and trained to yield probabilities between 0 and 1 for each new/test unit. This modification of the deep neural network gave probabilities that are the estimated propensity scores. As discussed in Chapter 3, the procedure then became a generalisation of the *logistic regression* function. Thus, we have made a crucial contribution to the body of statistical knowledge by demonstrating that deep learning algorithms such as the deep neural network can be used for estimating propensity scores through classification. The propensity scores are then used to assess covariate balance. Covariate balance is done to manage the bias-variance trade-off by ensuring balance on as many covariates and their transformations as possible while retaining a high effective sample size. Our results show that logistic regression provided adequate covariate balance compared to the deep neural network models using the average standardised absolute mean difference (ASAMD). Although logistic regression achieved good covariate balance, its absolute biases were consistently higher

than those of the deep neural network model. This important finding may suggest that achieving covariate balance may not be enough to lower bias, or it may be that ASAMD does not adequately measure covariate balance.

The study has shown that deep learning methods can be used in cases where the objective is to reduce absolute bias in treatment effects obtained by using propensity scores. Also, deep learning techniques make fewer assumptions than logistic regression, and often deal implicitly with interactions and nonlinearities. We strongly believe that deep learning algorithms such as the deep neural network can be employed as a mechanism for estimation of propensity scores and reducing bias. Thus, this research has provided a platform for further exploration by researchers.

7.3 A Deep Learning Method for Evaluating Causal Inference Using Change-Points

Objective 2 (Paper 2): To develop a hybrid model that incorporates a deep learning algorithm, the long short-term memory (LSTM) model, and a statistical nonparametric estimator, the kernel quantile estimator (KQE). This hybrid model will be used to detect change points in time series data and apply the model to evaluate the effects of the COVID-19 interventions imposed by by the South African Government.

In this paper (Chapter 4) we used deep learning methods to detect change points in time series data, and hence evaluate the causal effect of a change point. The main objective of the article was to determine whether the change points detected by our algorithm represented the interventions of the South African government or the measures that were taken to control the spread of COVID-19. A hybrid model that incorporated the long short-term memory (LSTM) algorithm and a statistical nonparametric estimator, the kernel quantile estimator (KQE) (Whata and Chimedza, 2021b) was developed to detect change points in time series data. Also, the hybrid model is a nonparametric model, and it does not require advance knowledge of the true number of change-points. The model was then applied to estimate the effects of the COVID-19 interventions imposed by the South African Government. We used the COVID-19 Community Mobility Reports by Google (Google LLC., 2020)

to detect change points and consequently quantify the causal effect of the South African government interventions such as a lockdown (a change-point) on population mobility. The developed model was able to detect abrupt changes such as lockdowns that are sufficiently "large" regardless of the noise levels in the data and the size of the data. This is crucial to avoid having several false positives. The hybrid model used a data point reconstruction error, which is the error between the original value of the data point and its low-dimensional reconstruction, to detect a change point as an anomaly.

Current change point detection methods that detect changes in parameters such as the mean or variance have a deficiency in that they do not detect isolated abnormal points such as anomalies, and they are usually supplemented with tools such as the Shewhart control (Taylor, 2000). The main contribution of our study is the development of an algorithm that addresses this shortcoming as the change-points are detected as anomalies in the time series. In addition, the algorithm does not depend on the statistical properties such as the mean before or after a change-point. A key component of the performance of reconstruction-based methods is the threshold, which represents the value of the reconstruction error where a data point is labelled as an anomaly or change-point. We have developed a method that does not estimate changes in the mean process or changes in the mean and/or variance. However, authors such as Eckley et al. (2011); Niekum et al. (2014); Aminikhanghahi and Cook (2017); Gao et al. (2020) have developed parametric models that have detected change-points in model parameters. Parametric methods have worked efficiently when the underlying functional form is specified correctly. We have shown in this study that our proposed model, the LSMAE and KQE, does not make strong assumptions about a specific functional form. Thus, the model can freely learn any functional form from the training data.

One of the specific objectives of Chapter 4 was to develop an approach to inferring causal effects in time series data using South Africa's COVID-19 interventions as an example. This is, to our knowledge, the first application available in South Africa that incorporates change point analysis on population mobility trends and causal inference to *quantify* the effects of an intervention such as a full lockdown on population movements. The approach used a hybrid model LSMAE and KQE to detect change-points that were subsequently used to infer the causal effects of the COVID-19 interventions. The LSMAE and

KQE algorithm was developed to detect change points using time series data on population mobility trends before and after an intervention. Subsequently, the BSTSM were implemented in the CausalImpact R package (Brodersen et al., 2015) to predict the counterfactuals. The causal effect was estimated as the difference between the observed population mobility (before the intervention) and the population mobility that would have been observed had the intervention not occured (counterfactual).

As the counterfactual cannot be observed, the BSTSM were used to estimate them. The contribution of this study to the available literature is that counterfactuals or potential outcomes can also be applied to time series data when evaluating causal inference. It has also shown that when evaluating the impact of an intervention, it is crucial to include a counterfactual. Thus, the counterfactual framework has successfully been applied to assess the causal effect of an intervention within an interrupted time series model (Morgan and Winship, 2015). The interrupted time series (ITS) model is the simplest design that is applied to time series data. The primary weakness of an ITS model is that the evolution of a response variable prior to an intervention may not be a sufficiently good predictor of how the response variable would evolve in the absence of an intervention. Thus, we have recognised that the counterfactual trajectory is poorly predicted by simply taking a straightforward linear extrapolation from the observed data before the intervention. Also, assuming that the counterfactual trajectory is a linear trajectory would result in substantial overestimation of the treatment effect. Thus, this study makes a valuable contribution by showing that the counterfactual can be estimated at the point where there is a change-point using the BSTSM framework. The BSTSM framework uses available prior knowledge about the model parameters to explicitly model the counterfactual of a time series observed both before and after the intervention. Additionally, the framework also uses state-space time series models which include linear regressions of the contemporaneous predictors (Brodersen et al., 2015).

This study has demonstrated that the LSTMAE and KQE coupled with the BSTSM can be successfully applied to real-world data to accurately detect change points that are the result of a policy intervention. Also, as mentioned by Brodersen et al. (2015), using BSTSM improves existing methods by providing a fully Bayesian time series estimate for the effect. The advantage of

using BSTSM is that they use model averaging to construct the most appropriate synthetic control for modelling the counterfactual. This is an improvement from just using a linear extrapolation of the observed data before an intervention.

7.4 Credibility of Causal Estimates from Regression Discontinuity Designs (RDD)

Objective 3 (Paper 3): To evaluate the credibility of causal estimates from regression discontinuity designs with multivariate assignment variables.

In this paper (Chapter 5), we focused on a method that can be used to evaluate statistical causal inference. Specifically, we provided an easier way of using multivariate regression discontinuity designs as a method of causal inference, using the potential-outcomes framework. We have shown that whenever a researcher is faced with the complexity caused by having more than one assignment variable, it is crucial to choose an appropriate regression discontinuity design. This chapter demonstrated that when working with an observational study, where one is comparing the outcomes for exposed and unexposed units, the problem of causal inference can be tackled in a regression discontinuity design.

Several papers have used the MRDD for causal inference, but their methods for estimating the causal effect estimands of interest vary substantially. However, there are few authors, such Papay et al. (2011); Reardon and Robinson (2012); Wong et al. (2013); Cheng (2016), who have extended the conventional RDD to the MRDD. For example, Wong et al. (2013) compared four different methods for MRDD. Papay et al. (2011, 2014) proposed a specific method, the multidimensional response surface RDD, for evaluating causal inference with multiple assignment variables. Reardon and Robinson (2012) compared the four methods proposed by Wong et al. (2013) and the multidimensional response surface method used by Papay et al. (2011). Despite the attempt of these authors to extend conventional RDD to MRDD, the literature on estimation methods for MRDD is still very scant or limited. Chapter 5 extends the nascent literature on MRDD by applying the frontier MRDD proposed by Reardon and Robinson (2012) in several ways:

(i) the frontier MRDD is applied to simulated data with different levels of variability induced in the simulated data sets as well as using different sample sizes,

(ii) the frontier MRDD is applied to a real world data set using the graduate admission data set (Acharya et al., 2019),

(iii) the credibility of the causal estimands obtained using the frontier MRDD are assessed by using supplementary analyses

Supplementary analyses have not always been applied systematically and consistently in the empirical literature. However, this study has highlighted the importance of carrying them out and highlighting their growing importance. One of the most useful and widely used supplementary analyses is the McCrary (2008) test in regression discontinuity designs. The McCrary test assesses whether there is a discontinuity in the density of the forcing variable at the threshold. The identification strategy underlying regression discontinuity designs relies on the assumption that units just to the left and just to the right of the threshold are comparable. That argument is difficult to reconcile if, say, for some reason there is a discontinuity that is not attributable to the treatment effect and when there is evidence of manipulation of the assignment variables. Instead of using the McCrary test, we successfully applied a newer intuitive and easy-to-implement nonparametric density estimator that is based on local polynomial techniques (Cattaneo et al., 2020), to test for evidence of manipulation of the assignment variables. Also, the manipulation test results obtained from using the nonparametric density estimator are easy to interpret. Cattaneo et al. (2020) indicate that the density estimator is fully automatic and does not require any other transformation or pre-binning of the data. Moreover, the authors indicate that nonparametric manipulation tests using local-polynomial density estimators are becoming increasingly important for the falsification of regression discontinuity designs.

Based on the outcomes discussed in Chapter 5, we recommend using the frontier MRDD as it is easy to implement. It reduces the assumptions checking process to a series of well defined single assignment variables RDD, whose methods are well-defined in the literature (Reardon and Robinson, 2012). It allows for a data-driven bandwidth selection process, thereby, eliminating the possibility of manipulating the results by choosing an arbitrary bandwidth.

Furthermore, through the use of supplementary analyses, we found credible evidence of a relationship between the outcome variable and the assignment variable(s) in the simulated data using graphical analysis. This gave credibility to the causal estimates as it supported the existence of treatments effects at the cut-offs. In addition, by applying the local polynomial density estimators to the simulated data, we found no evidence of manipulation of the assignment variable, thus making the causal estimates more credible. Another important contribution of Chapter 5 is that we have demonstrated that the frontier approach can be implemented successfully to a real-world data set with significant causal estimates. Also, we have demonstrated the importance of accompanying every conventional or multivariate regression discontinuity design with supplementary analyses to give more credibility to the causal estimates.

7.5 Deep Learning for SARS CoV-2

Objective 4 (Paper 4): To develop a hybrid deep learning model for the classification of SARS CoV-2 virus genes and also use approximate statistical tests to compare the predictive performance of the CNN-BiLSTM, CNN-LSTM and, CNN models in classifying SARS-CoV-2 genes.

Chapter 6 presented paper 4. In this chapter, we evaluated statistically the predictive performance of different deep learning algorithms in modelling rare binary outcomes. Specifically, we used statistical hypothesis tests to compare the predictive performance of the CNN-BiLSTM, CNN-LSTM and, CNN models in classifying SARS-CoV-2 genes. Following recommendations by Dietterich (1998), we used the 5 $\times$ 2-fold cv paired t-test to perform post-hoc analyses to determine the statistical significance of the differences in the means of the performance metric scores obtained by the CNN-BiLSTM, CNN-LSTM and, CNN models in classifying SARS-CoV-2 genes. Also, the chapter makes a contribution to the global effort to combat the SARS COV-2 virus by proposing a deep learning algorithm that uses a convolutional neural network (CNN) as well as a bi-directional long short-term memory (Bi-LSTM) neural network, for the classification of SARS CoV-2 among coronaviruses. In this chapter, we also classified whether a genome sequence contains candidate regulatory motifs, i.e., we were interested in identifying sequences that contain regulatory

motifs for the SARS COV-2 viruses amongst other SARS sequences that belong to the Coronavirus family. The algorithm proposed in this article is a useful diagnostic tool that is capable of reliably identifying the SARS CoV-2 virus and distinguishing it from other coronaviruses or other pathogens. Correct classification is also important in discovering different species of coronaviruses that may affect people in the future. In addition, the SARS CoV-2 virus gene is highly transmissible, so proper identification of SARS CoV-2 is very important in the management of the spread of the virus. Our algorithm can be used in practice to augment polymerise chain reaction (PCR) and antibody testing techniques that are currently the dominant ways global healthcare systems are using to test citizens for Covid-19. The algorithm developed has the potential to contribute meaningfully to the fight against Covid-19 as it is quicker and reliable in the detection of SARS COV-2 genome sequences and differentiating them from other virus strains of the same Coronavirus family.

This chapter contributes in similar ways to Chapter 3 by developing statistical learning methods, such as deep learning algorithms, that can perform the classification task accurately. Instead of predicting the class probabilities that give the propensity score, this chapter focused more on predicting the class labels. To obtain accurate propensity scores using the classification task, one must have an accurate classifier whose classification accuracy is significantly greater than the no information rate (NIR). Therefore, the main contribution of this chapter was the development of a classifier (CNN-Bi-LSTM) that could efficiently distinguish between SARS-CoV-2 gene sequences from non-SARS CoV-2 gene sequences and then compare its classification performance with that of CNN and CNN-LSTM classifiers. Also, these classifiers can be used to predict propensity scores as seen in Chapter 3. The chapter extends the current literature on (deep learning) classification by recommending a hypothesis testing procedure that can be used to compare different classifiers and evaluate whether or not their difference in performance is statistically significant. These hypothesis tests are important in that they can spot any differences better than the human eye and also test if differences in the performance of the models are statistically significant. Following Dietterich (1998), we recommend the use of the post-hoc 5×2-fold cv paired t-test as opposed to the k-fold cross-validated paired t-test to test for differences in performance relative to a performance metric such as the AUC-ROC. This is because in the

k-fold cross-validation paired t-test, the training data sets may overlap. According to Dietterich (1998), an overlap of the training data sets may prevent the paired t-test from obtaining good estimates thereby, leading to higher probabilities of type 1 errors. Therefore, the problem of overlapping of the training data sets was tackled by using a 5×2-fold cv paired t-test which is based on repeating two-fold cross-validations five times. The outcomes of the application of the 5×2-fold cv paired t-test are discussed in Chapter 6.

In Chapter 6, we have demonstrated that a fruitful collaboration between statistics and deep learning is possible. We have shown that it may not be enough to just report on the performance metrics such as AUC-ROC or classification accuracy, but one has to perform some statistical post-hoc analysis to test the hypotheses that the performances of two or more competing deep learning classifiers are statistically significant. Such statistical hypothesis tests to assess statistically the difference in performance will assist researchers to select the appropriate classification model(s) when comparing two or more deep learning algorithms.
All python and R codes developed and used in this thesis are available upon request by email the author at albert.whata@spu.ac.za.

CHAPTER 8

Conclusions and Recommendations for Future Work

8.1 Conclusions

Machine learning (ML) algorithms are generally good at predicting outcomes rather than explaining causality. Several research papers have shown that most ML algorithms are very good at finding correlations in data, but not so good at understanding causality. This limits the ability to use ML for decision making. For example, business needs tools that can understand causal relationships between data and solutions that are easily generalisable. This means that it has become increasingly important to improve the generalisation of ML methods. This is because in their current state, ML methods are often biased, have a general lack of explainability, and have a limited ability to generalise the patterns that they discover in training data sets. In this study, we used the RCM to evaluate causal inference. The RCM was based on "potential outcomes" to define causal effects, that is, we derived inferences for causal effects from the observed data by conceptualizing the problem as one of imputing the missing potential outcomes (counterfactuals). Using available data on the actual treatments that were received by units, we then modelled the outcomes that would have been observed given a set of covariates. This was done to derive the predictions of the potential outcomes that would have been observed if the treatment assignments had been different. Thus, this way we generated stochastic predictions of the counterfactuals in this study. A comparison of the

counterfactual and the actually observed potential outcomes was made which made it possible to calculate any causal-effect estimand that we were interested in. Because of the stochastic nature of the missing counterfactuals, repeated experiments were performed to yield different values for the causal-effect estimate. The variations in the stochastic predictions then allowed us to estimate the average treatment effects. The RCM was employed in this study because of its flexibility in articulating causal estimates as well as its ability to handle difficult cases such as observational studies. In observational studies, we noted that potential outcomes and estimated propensity scores play a vital role in estimating causal estimates because the analysis is done *as if* the assignment mechanisms were unconfounded. Thus, we used a propensity score-potential outcome framework to evaluate statistical causal inference. We also used ML as part of causal inference. The main focus of this research was to find ways in which statistics and ML can work together to evaluate causal inference. The research's aims and objectives were achieved through the following outcomes;

1. *Development of a deep neural network (DNN) to estimate propensity scores.* A DNN was developed that can be used to estimate propensity scores, which were in turn used to estimate causal effect estimates using both simulated data and a real-world data set. A performance comparison of DNN and logistic regression (LR) was performed and it was shown that DNN produced causal estimates that were less biased on average than logistic regression (Chapter 3). However, the propensity scores obtained from LR achieved covariate balance compared to those of DNN. This means that it is not enough to achieve covariate balance if the objective is to reduce bias.

2. *Development of an algorithm that combines a deep learning algorithm (LSTMAE) and a kernel quintile estimator (KQE) to detect change points.* This was a critical contribution as we developed an algorithm, the LSTMAE and KQE, that does not make strong assumptions about a specific functional form. Therefore, the algorithm can freely learn any functional form from the training data. This algorithm can be used to evaluate causal inference in time series data. After automatically detecting change points, methods such as the Bayesian Structural Time Series (BSTSM) model can be employed to estimate the causal effect of a change point. To the best of our knowledge, this is the first algorithm that has successfully detected the change points or interventions (lockdowns) that were imposed by the South

African government in its fight to contain the spread of COVID-19. Also, it is the first time that a deep learning algorithm and statistical techniques have been integrated to detect change points that were used to evaluate the causal effects of COVID-19 interventions (lockdowns) imposed by the South African government.

3. *Extension of the conventional RDD to the multiple RDD*
   The work carried out in this study was the first detailed simulation study on the effects of matric points and household income on the chance that a student gets NSFAS funding in South Africa using regression discontinuity designs. The work was an extension of conventional RDD to MRDD. As detailed in the simulation study and the case study in Chapter 5, we have highlighted the importance of supplementary analyses. Therefore, we have added to the body of research on regression discontinuity designs by extending the conventional RDD as well as by conducting supplementary analyses that seek to add more *credibility* to the causal estimates obtained through primary analyses. We strongly recommend the use of the frontier multivariate regression discontinuity design, as it is easier to implement. Additionally, it incorporates discontinuities in multivariate assignment variables into single regression discontinuity designs along a number of frontiers of the treatment variables.

4. *Development of deep learning methods for classification*
   The work outlined in this study is an attempt to develop a deep learning algorithm that can accurately classify SARS CoV-2 among Coronaviruses. Thus, the developed models performed a classification task. In Chapter 3 it was shown that if one is to use deep learning algorithms for estimating propensity scores, one has to ensure that the model is accurate and sensible when performing binary classification. It was shown that it is not enough to report on metrics such as classification accuracy or area under receiver operating characteristic curve when comparing the performance of competing ML models. However, further statistical analysis is required to perform statistical tests of hypothesis to ascertain whether or not the difference in performance between the competing models is statistically significant. Several papers report on a number of performance metrics, but few have taken the extra step to perform post-hoc statistical hypothesis tests to test for the difference in performance between two ML models. Thus, this work has

demonstrated the importance of such post-hoc statistical hypothesis analysis as outlined in Chapter 6.

8.2 Recommendations for Future Work

Although the methods developed in this thesis have strengthened the applicability of both statistical and deep learning methods for evaluating causal inference, there remain some critical areas that need to be further addressed through research. These are;

- More simulation studies using different data generating mechanisms as well as applications to real-world data sets are required for estimating propensity scores using deep learning algorithms,
- Using CNN-BiLSTM developed in Chapter 6 to estimate propensity scores. Also, further work is required on assessing covariate balance by using propensity scores that are estimated by more deep learning algorithms,
- Evaluation of the performance of the developed deep learning classification algorithm (CNN-BiLSTM) on the new variants of SARS-CoV-2,
- Apply the methods developed in Chapter 5 to estimate the causal effect of household income and matric points when the consolidated data of these variables becomes available from NSFAS.
- Apply the LSTMAE + KQE algorithm in other work such as in anomaly-based intrusion detection systems (IDS) to validate the methods.

References

Department of Co-operative Governance and Traditional Affairs (2020). *Regulations Issued in Terms of Section 27(2) of the Disaster Management Act, 2002*). Government Gazzette, 43107:657, 18 march. https://www.gov.za/speeches/disaster-management-act-2002-regulations-issued-terms-disaster-management-act-2002-minister.

South Africa. Department. Public Service and Administration (2020). State of disaster covid-19: Public service return to work guidelines after easing of national lockdown, the dpsa: Circular no. 18 of 2020, 01 may. http://www.dpsa.gov.za/covid19.php.

South Africa. Dept. of Co-operative Governance and Traditional Affairs. (2020). Disaster management act, 2002: Declaration of a national state of disaster. government gazette no. 43096:313, 15 mar. https://www.gov.za/documents/disaster-management-act-declaration-national-state-disaster-covid-19-coronavirus-16-mar.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.

Abadie, A., Diamond, A., and Hainmueller, J. (2015). Comparative politics and the synthetic control method. *American Journal of Political Science*, 59(2):495–510.

ACAPS. (2020). Covid-19 government measures dataset. https://www.acaps.org/covid-19-government-measures-dataset.

Acharya, M. S., Armaan, A., and Antony, A. S. (2019). A comparison of regression models for prediction of graduate admissions. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pages 1–5. IEEE.

Adams, G. (2020). A beginner's guide to rt-pcr, qpcr and rt-qpcr. *The Biochemist*, 42(3):48–53. doi: doi.org/10.1042/BIO20200034.

Ajam, T. (2020). The economic costs of the pandemic–and its response: more eyes on covid-19: perspectives from economics. *South African Journal of Science*, 116(7-8).

Akosa, J. (2017). Predictive accuracy: A misleading performance measure for highly imbalanced data. In *Proceedings of the SAS Global Forum*, pages 2–5.

Albawi, S., Mohammed, T. A., and Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. IEEE. doi:10.1109/ICEngTechnol.2017.8308186.

Ali, M. S., Prieto-Alhambra, D., Lopes, L., Ramos, D., Bispo, N., Ichihara, M. Y., Pescarini, J. M., Williamson, E., Fiaccone, R., Barreto, M. L., et al. (2019). Propensity score methods in health technology assessment: principles, extended applications, and recent advances. *Frontiers in Pharmacology*, 10:973.

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature Biotechnology*, 33(8):831–838. doi: 10.1038/nbt.3300.

Alla, S. and Adari, S. K. (2019). *Beginning Anomaly Detection Using Python-Based Deep Learning*. Springer.

Allen, M. (2017). *The SAGE encyclopedia of communication research methods*. Sage Publications.

Aloi, A., Alonso, B., Benavente, J., Cordera, R., Echániz, E., González, F., Ladisa, C., Lezama-Romanelli, R., López-Parra, Á., Mazzei, V., et al. (2020). Effects of the covid-19 lockdown on urban mobility: Empirical evidence from the city of santander (spain). *Sustainability*, 12(9):3870.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410. doi:10.1016/S0022-2836(05)80360-2.

Amin, M. Z. and Nadeem, N. (2018). Convolutional neural network: text classification model for open domain question answering system. *arXiv preprint arXiv:1809.02479*.

Aminikhanghahi, S. and Cook, D. J. (2017). A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367.

Amruthnath, N. and Gupta, T. (2018). A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 355–361. IEEE. doi:10.1109/IEA.2018.8387124.

Angrist, J. D. and Pischke, J.-S. (2008). *Mostly harmless econometrics: An empiricist's companion*. Princeton university press.

Antonakis, J., Bendahan, S., Jacquart, P., and Lalive, R. (2014). And solutions. *The Oxford handbook of leadership and organizations*, page 93.

Arai, Y., Otsu, T., Seo, M. H., et al. (2019). Causal inference on regression discontinuity designs by high-dimensional methods. Technical report, Suntory and Toyota International Centres for Economics and Related ....

Arthurs, N., Stenhaug, B., Karayev, S., and Piech, C. (2019). Grades are not normal: Improving exam score models using the logit-normal distribution. *International Educational Data Mining Society*.

Asi, H. and Duchi, J. C. (2019). The importance of better models in stochastic optimization. *Proceedings of the National Academy of Sciences*, 116(46):22924–22930.

Atalan, A. (2020). Is the lockdown important to prevent the covid-19 pandemic? effects on psychology, environment and economy-perspective. *Annals of medicine and surgery*, 56:38–42.

Athey, S., Imbens, G., Pham, T., and Wager, S. (2017). Estimating average treatment effects: Supplementary analyses and remaining challenges. *American Economic Review*, 107(5):278–81.

Athey, S. and Imbens, G. W. (2015). Machine learning methods for estimating heterogeneous causal effects. *stat*, 1050(5).

Athey, S. and Imbens, G. W. (2017). The state of applied econometrics: Causality and policy evaluation. *Journal of Economic Perspectives*, 31(2):3–32.

Athey, S., Imbens, G. W., and Wager, S. (2016a). Approximate residual balancing: De-biased inference of average treatment effects in high dimensions. *arXiv preprint arXiv:1604.07125*. doi: 10.1111/rssb.12268.

Athey, S., Imbens, G. W., Wager, S., et al. (2016b). Efficient inference of average treatment effects in high dimensions via approximate residual balancing. Technical report.

Austin, P. C. (2011). An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate behavioral research*, 46(3):399–424.

Austin, P. C. and Stuart, E. A. (2015). 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*, 34(28):3661–3679.

Bailey, T. L., Elkan, C., and others (1994). Fitting a mixture model by expectation maximization to discover motifs in bipolymers.

Barry, D. and Hartigan, J. A. (1993). A bayesian analysis for change point problems. *Journal of the American Statistical Association*, 88(421):309–319.

Baser, O. et al. (2007). Propensity score matching with limited overlap. *Economics Bulletin*, 9(8):1–8.

Beal, S. J. and Kupzyk, K. A. (2014). An introduction to propensity scores: what, when, and how. *The Journal of Early Adolescence*, 34(1):66–92.

Bellora, N., Farré, D., and Mar Alba, M. (2007). PEAKS: identification of regulatory motifs by their position in DNA sequences. *Bioinformatics*, 23(2):243–244. doi: 10.1093/bioinformatics/btl568.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Berrar, D. (2019). Cross-validation. *Encyclopedia of bioinformatics and computational biology*, 1:542–545. doi: 10.1016/B978-0-12-809633-8.20349-X.

Blei, D. M. (2015). Regularized regression. *Technometrics*.

Bonaccorsi, G., Pierri, F., Cinelli, M., Porcelli, F., Galeazzi, A., Flori, A., Schmidth, A. L., Valensise, C. M., Scala, A., Quattrociocchi, W., et al. (2020). Evidence of economic segregation from mobility lockdown during covid-19 epidemic. *Available at SSRN 3573609*.

Bontemps, L., McDermott, J., Le-Khac, N.-A., et al. (2016). Collective anomaly detection based on long short-term memory recurrent neural networks. In *International Conference on Future Data and Security Engineering*, pages 141–152. Springer.

Boughorbel, S., Jarray, F., and El-Anbari, M. (2017). Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PloS one*, 12(6). doi:10.1371/journal.pone.0177678.

Bouttell, J., Craig, P., Lewsey, J., Robinson, M., and Popham, F. (2018). Synthetic control methodology as a tool for evaluating population-level health interventions. *J Epidemiol Community Health*, 72(8):673–678.

Brodersen, K. H., Gallusser, F., Koehler, J., Remy, N., Scott, S. L., et al. (2015). Inferring causal impact using bayesian structural time-series models. *The Annals of Applied Statistics*, 9(1):247–274.

Brookhart, M. A., Stürmer, T., Glynn, R. J., Rassen, J., and Schneeweiss, S. (2010). Confounding control in healthcare database research: challenges and potential approaches. *Medical care*, 48(6 0):S114.

Brown, K., Merrigan, P., and Royer, J. (2018). Estimating average treatment effects with propensity scores estimated with four machine learning procedures: Simulation results in high dimensional settings and with time to event outcomes. *Available at SSRN 3272396*.

Calonico, S., Cattaneo, M. D., Farrell, M. H., and Titiunik, R. (2017). rdrobust: Software for regression-discontinuity designs. *The Stata Journal*, 17(2):372–404.

Cannas, M. and Arpino, B. (2019). A comparison of machine learning algorithms and covariate balance measures for propensity score matching and weighting. *Biometrical Journal*, 61(4):1049–1072.

Cattaneo, M. D., Frandsen, B. R., and Titiunik, R. (2015). Randomization inference in the regression discontinuity design: An application to party advantages in the us senate. *Journal of Causal Inference*, 3(1):1–24.

Cattaneo, M. D., Idrobo, N., and Titiunik, R. (2019). *A practical introduction to regression discontinuity designs: Foundations*. Cambridge University Press.

Cattaneo, M. D., Jansson, M., and Ma, X. (2018a). Manipulation testing based on density discontinuity. *The Stata Journal*, 18(1):234–261.

Cattaneo, M. D., Jansson, M., and Ma, X. (2020). Simple local polynomial density estimators. *Journal of the American Statistical Association*, 115(531):1449–1455.

Cattaneo, M. D., Titiunik, R., and Vazquez-Bare, G. (2018b). rdlocrand: Local randomization methods for rd designs. *R package version 0.3. URL: https://CRAN. R-project. org/package= rdlocrand*.

Cattaneo, M. D., Titiunik, R., Vazquez-Bare, G., and Keele, L. (2016). Interpreting regression discontinuity designs with multiple cutoffs. *The Journal of Politics*, 78(4):1229–1248.

Cepeda, M. S., Boston, R., Farrar, J. T., and Strom, B. L. (2003). Comparison of logistic regression versus propensity score when the number of events is low and there are multiple confounders. *American journal of epidemiology*, 158(3):280–287.

Chattopadhyay, I. (2014). Causality networks. *arXiv preprint arXiv:1406.6651*.

Chawla, A., Jacob, P., Lee, B., and Fallon, S. (2019). Bidirectional lstm autoencoder for sequence based anomaly detection in cyber security. *International Journal of Simulation–Systems, Science & Technology*. doi:10.5013/IJSSST.a.20.05.07.

Chen, C.-W., Tseng, S.-P., Kuan, T.-W., and Wang, J.-F. (2020a). Outpatient text classification using attention-based bidirectional lstm for robot-assisted servicing in hospital. *Information*, 11(2):106.

Chen, C.-W., Tseng, S.-P., Kuan, T.-W., and Wang, J.-F. (2020b). Outpatient Text Classification Using Attention-Based Bidirectional LSTM for Robot-Assisted Servicing in Hospital. *Information*, 11(2). doi:10.3390/info11020106.

Chen, S. (2020). *Causality Inference between Time Series Data and Its Applications*. PhD thesis, Columbia University.

Cheng, Y. A. (2016). Regression discontinuity designs with multiple assignment variables.

Chikahara, Y. and Fujino, A. (2018). Causal inference in time series via supervised learning. In *IJCAI*, pages 2042–2048.

Chollet, F., Allaire, J., et al. (2017). R interface to keras. `https://github.com/rstudio/keras`.

Chollet, F. and others (2015). Keras. Publisher: GitHub.

Cochran, W. (2015). Introduction to observational studies and the reprint of cochran's paper "observational studies" and comments. *Observational Studies 1*, pages 124–125.

Dasgupta, I., Wang, J., Chiappa, S., Mitrovic, J., Ortega, P., Raposo, D., Hughes, E., Battaglia, P., Botvinick, M., and Kurth-Nelson, Z. (2019). Causal reasoning from meta-reinforcement learning. *arXiv preprint arXiv:1901.08162*.

de Vries, B. B. P., van Smeden, M., and Groenwold, R. H. (2018). Propensity score estimation using classification and regression trees in the presence of missing covariate data. *Epidemiologic Methods*, 7(1).

Deaton, A. and Cartwright, N. (2018). Understanding and misunderstanding randomized controlled trials. *Social Science & Medicine*, 210:2–21.

Dehejia, R. H. and Wahba, S. (2002). Propensity score-matching methods for nonexperimental causal studies. *Review of Economics and statistics*, 84(1):151–161.

Dehning, J., Zierenberg, J., Spitzner, F. P., Wibral, M., Neto, J. P., Wilczek, M., and Priesemann, V. (2020). Inferring change points in the spread of covid-19 reveals the effectiveness of interventions. *Science*. doi: 10.1126/science.abb9789.

Deke, J. (2014). Using the linear probability model to estimate impacts on binary outcomes in randomized controlled trials. Technical report, Mathematica Policy Research.

Deza, M. M. and Deza, E. (2009). Encyclopedia of distances. In *Encyclopedia of distances*, pages 1–583. Springer.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Ding, Z., Xia, R., Yu, J., Li, X., and Yang, J. (2018). Densely connected bidirectional lstm with applications to sentence classification. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 278–287. Springer.

Dinka, H. and Milkesa, A. (2020). Unfolding SARS-CoV-2 viral genome to understand its gene expression regulation. *Infection, Genetics and Evolution*. doi:10.1016/j.meegid.2020.104386.

Dobson, A. J. and Barnett, A. G. (2018). *An introduction to generalized linear models*. CRC press.

Doudchenko, N. and Imbens, G. W. (2016). Balancing, regression, difference-in-differences and synthetic control methods: A synthesis. Technical report, National Bureau of Economic Research.

Dutta, V., Choraś, M., Pawlicki, M., and Kozik, R. (2020). A deep learning ensemble for network anomaly and cyber-attack detection. *Sensors*, 20(16):4583.

Eckley, I. A., Fearnhead, P., and Killick, R. (2011). Analysis of changepoint models. *Bayesian Time Series Models*, pages 205–224.

Edgington, E. S. (1985). Random assignment and experimental research. *Educational Administration Quarterly*, 21(3):235–246.

Egleston, B. L., Scharfstein, D. O., Freeman, E. E., and West, S. K. (2007). Causal inference for non-mortality outcomes in the presence of death. *Biostatistics*, 8(3):526–545.

Eichler, M. (2013). Causal inference with multiple time series: principles and problems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1997):20110613.

Elsayed, M. S., Le-Khac, N.-A., Dev, S., and Jurcut, A. D. (2020). Network anomaly detection using lstm based autoencoder. In *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pages 37–45.

Erdman, C. and Emerson, J. W. (2007). bcp: An R package for performing a bayesian analysis of change point problems. *Journal of Statistical Software*, 23(3):1–13.

Farahnakian, F. and Heikkonen, J. (2018). A deep auto-encoder based approach for intrusion detection system. In *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pages 178–183. IEEE.

Farahnakian, F. and Heikkonen, J. (2019). Anomaly-based intrusion detection using deep neural networks.

Farrell, M. H., Liang, T., and Misra, S. (2018). Deep neural networks for estimation and inference. *arXiv preprint arXiv:1809.09953*.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Gallagher, C., Lund, R., and Robbins, M. (2013). Changepoint detection in climate time series with long-term trends. *Journal of Climate*, 26(14):4994–5006. doi: 10.1175/JCLI-D-12-00704.1.

Galway, L., Charles, D., and Black, M. (2008). Machine learning in digital games: a survey. *Artificial Intelligence Review*, 29(2):123–161.

Gao, W., Yang, H., and Yang, L. (2020). Change points detection and parameter estimation for multivariate time series. *Soft Computing*, 24(9):6395–6407.

Garrod, N. and Wildschut, A. (2021). How large is the missing middle and what would it cost to fund? *Development Southern Africa*, 38(3):484–491.

Gharibzadeh, S., Mansournia, M. A., Rahimiforoushani, A., Alizadeh, A., Amouzegar, A., Mehrabani-Zeinabad, K., and Mohammad, K. (2018). Comparing different propensity score estimation methods for estimating the marginal causal effect through standardization to propensity scores. *Communications in Statistics-Simulation and Computation*, 47(4):964–976.

Girman, C. J., Gokhale, M., Kou, T. D., Brodovicz, K. G., Wyss, R., and Stürmer, T. (2014). Assessing the impact of propensity score estimation and implementation on covariate balance and confounding control within and across important subgroups in comparative effectiveness research. *Medical care*, 52(3):280.

Goldstein, N. D., LeVasseur, M., and McClure, L. A. (2020). On the convergence of epidemiology, biostatistics, and data science. *Harvard Data Science Review*, 2(2).

Goller, D., Lechner, M., Moczall, A., and Wolff, J. (2020). Does the estimation of the propensity score by machine learning improve matching estimation? the case of germany's programmes for long term unemployed. *Labour Economics*, page 101855.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Goodman-Bacon, A. and Marcus, J. (2020). Using difference-in-differences to identify causal effects of covid-19 policies.

Google LLC. (2020). Google covid-19 community mobility reports. https://www.google.com/covid19/mobility/.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, 18(5-6):602–610. doi:10.1016/j.neunet.2005.06.042.

Greifer, N. (2017). cobalt: Covariate balance tables and plots. *R package version*, 2(0).

Guidoum, A. C. (2015). Kernel estimator and bandwidth selection for density and its derivatives. *The kedd package, version*, 1.

Gulum, M. A., Trombley, C. M., and Kantardzic, M. (2021). A review of explainable deep learning cancer detection models in medical imaging. *Applied Sciences*, 11(10):4573.

Hahn, J. (1998). On the role of the propensity score in efficient semiparametric estimation of average treatment effects. *Econometrica*, pages 315–331.

Hainmueller, J. (2012). Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies. *Political analysis*, pages 25–46.

Hannenhalli, S. (2008). Eukaryotic transcription factor binding sites—modeling and integrative search methods. *Bioinformatics*, 24(11):1325–1331. doi: 10.1093/bioinformatics/btn198.

Harder, V. S., Stuart, E. A., and Anthony, J. C. (2010). Propensity score techniques and the assessment of measured covariate balance to test causal associations in psychological research. *Psychological methods*, 15(3):234.

Hassaballah, M. and Awad, A. I. (2020). *Deep learning in computer vision: principles and applications*. CRC Press.

Heckman, J. J. (2008). Econometric causality. *International statistical review*, 76(1):1–27.

Heinrich, C., Maffioli, A., Vazquez, G., et al. (2010). A primer for applying propensity-score matching. *Inter-American Development Bank*.

Hernán, M. A., Hernández-Díaz, S., and Robins, J. M. (2004). A structural approach to selection bias. *Epidemiology*, pages 615–625.

Hernán, M. A., Hsu, J., and Healy, B. (2019). A second chance to get causal inference right: a classification of data science tasks. *Chance*, 32(1):42–49.

Hernán, M. A. and Robins, J. M. (2006). Instruments for causal inference: an epidemiologist's dream? *Epidemiology*, pages 360–372.

Hettich, S. (1999). Kdd cup 1999 data. *The UCI KDD Archive*.

Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240.

Hill, R. C., Fomby, T. B., Escanciano, J. C., Hillebrand, E., and Jeliazkov, I. (2017). *Regression discontinuity designs: Theory and applications*. Emerald Group Publishing.

Hirano, K., Imbens, G. W., and Ridder, G. (2003). Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4):1161–1189.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. doi: 10.1162/neco.1997.9.8.1735.

Hoffmann, J., Maestrati, L., Sawada, Y., Tang, J., Sellier, J. M., and Bengio, Y. (2019). Data-driven approach to encoding and decoding 3-d crystal structures. *arXiv preprint arXiv:1909.00949*.

Holland, P. W. (1986). Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960.

Holland, P. W. and Rubin, D. B. (1987). Causal inference in retrospective studies. *ETS Research Report Series*, 1987(1):203–231.

Hu, S., Ma, R., and Wang, H. (2019). An improved deep learning method for predicting DNA-binding proteins based on contextual features in amino acid sequences. *PloS one*, 14(11):12–18. doi:10.1371/journal.pone.0225317.

Hu, Y., Luo, S., Han, L., Pan, L., and Zhang, T. (2020). Deep supervised learning with mixture of neural networks. *Artificial Intelligence in Medicine*, 102:101764.

Imai, K. and Ratkovic, M. (2014). Covariate balancing propensity score. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):243–263.

Imbens, G. (2019). Potential outcome and directed acyclic graph approaches to causality: Relevance for empirical practice in economics. Technical report, National Bureau of Economic Research.

Imbens, G. and Zajonc, T. (2011). Regression discontinuity design with multiple forcing variables. *Report, Harvard University*.

Imbens, G. W. and Lemieux, T. (2008). Regression discontinuity designs: A guide to practice. *Journal of econometrics*, 142(2):615–635.

Imbens, G. W. and Wooldridge, J. M. (2009). Recent developments in the econometrics of program evaluation. *Journal of economic literature*, 47(1):5–86.

Inada, M. (2012). Unbiased estimation of factorial effect by using analysis of covariance or propensity score method for observational studies in laboratory medicine. *Rinsho byori. The Japanese journal of clinical pathology*, 60(7):689–697.

Jagtap, A. D., Kawaguchi, K., and Karniadakis, G. E. (2020). Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136.

Jansson, A. (2017). Predicting trajectories of golf balls using recurrent neural networks.

Jianqiang, Z., Xiaolin, G., and Xuejun, Z. (2018). Deep convolution neural networks for twitter sentiment analysis. *IEEE Access*, 6:23253–23260. DOI: 10.1109/ACCESS.2017.2776930.

Johnson, R. and Zhang, T. (2015). Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112. doi:10.3115/v1/N15-1011.

Kamangar, F. (2012). Confounding variables in epidemiologic studies: basics and beyond. *Archives of Iranian medicine*, 15(8):0–0.

Kane, T. J. (2003). A quasi-experimental estimate of the impact of financial aid on college-going. Technical report, National Bureau of Economic Research.

Kang, J., Chan, W., Kim, M.-O., and Steiner, P. M. (2016). Practice of causal inference with the propensity of being zero or one: assessing the effect of arbitrary cutoffs of propensity scores. *Communications for statistical applications and methods*, 23(1):1.

Karsoliya, S. (2012). Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. *International Journal of Engineering Trends and Technology*, 3(6):714–717.

Karwa, V., Slavković, A. B., Donnell, E. T., et al. (2011). Causal inference in transportation safety studies: Comparison of potential outcomes and causal diagrams. *The Annals of Applied Statistics*, 5(2B):1428–1455.

Keele, L. J. and Titiunik, R. (2015). Geographic boundaries as regression discontinuities. *Political Analysis*, 23(1):127–155.

Kherlenchimeg, Z. and Nakaya, N. (2018). Network intrusion classifier using autoencoder with recurrent neural network. In *Proceedings of the Fourth International Conference on Electronics and Software Science (ICESS2018), Takamatsu, Japan*, pages 5–7.

Killick, R. and Eckley, I. (2014). changepoint: An r package for changepoint analysis. *Journal of statistical software*, 58(3):1–19.

Kingma, D. P. and Adam, J. B. (2014a). a method for stochastic optimization. arxiv preprint. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Adam, J. B. (2014b). a method for stochastic optimization. arXiv preprint. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kiperwasser, E. and Goldberg, Y. (2016). Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327. doi:10.1101/2020.02.26.967026v2.

Kozik, R. and Choraś, M. (2014). Machine learning techniques for cyber attacks detection. In *Image Processing and Communications Challenges 5*, pages 391–398. Springer.

Kuhn, M. (2012). The caret package. *R Foundation for Statistical Computing, Vienna, Austria. URL https://cran. r-project. org/package= caret.* Publisher: Citeseer.

Kuhn, M., Johnson, K., and others (2013). *Applied predictive modeling*, volume 26. Springer.

Kursa, M. B., Rudnicki, W. R., et al. (2010). Feature selection with the boruta package. *J Stat Softw*, 36(11):1–13.

Landis, J. R. and Koch, G. G. (1977a). The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Landis, J. R. and Koch, G. G. (1977b). The measurement of observer agreement for categorical data. *biometrics*, pages 159–174. doi:10.2307/2529310.

Lanza, S. T., Moore, J. E., and Butera, N. M. (2013). Drawing causal inferences using propensity scores: A practical guide for community psychologists. *American journal of community psychology*, 52(3-4):380–392.

Le, P. and Zuidema, W. (2016). Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs. *arXiv preprint arXiv:1603.00423*. doi:10.18653/v1/W16-1610.

Lechner, M. et al. (2011). *The estimation of causal effects by difference-in-difference methods*. Now Hanover, MA.

Lee, B. K., Lessler, J., and Stuart, E. A. (2010). Improving propensity score weighting using machine learning. *Statistics in medicine*, 29(3):337–346.

Lee, D. S. and Lemieux, T. (2010). Regression discontinuity designs in economics. *Journal of economic literature*, 48(2):281–355.

Leonard, J. K. (2019). Image classification and object detection algorithm based on convolutional neural network. *Science Insights*, 31(1):85–100.

Lewis, D. (1974). Causation. *The journal of philosophy*, 70(17):556–567.

Li, F., Morgan, K. L., and Zaslavsky, A. M. (2018a). Balancing covariates via propensity score weighting. *Journal of the American Statistical Association*, 113(521):390–400.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. *bioinformatics*, 25(14):1754–1760. doi:10.1093/bioinformatics/btp324.

Li, Y., Han, R., Bi, C., Li, M., Wang, S., and Gao, X. (2018b). DeepSimulator: a deep simulator for Nanopore sequencing. *Bioinformatics*, 34(17):2899–2908. doi:10.1093/bioinformatics/bty223.

Li, Z., Li, J., Wang, Y., and Wang, K. (2019). A deep learning approach for anomaly detection based on sae and lstm in mechanical equipment. *The International Journal of Advanced Manufacturing Technology*, 103(1-4):499–510.

Linden, A. and Adams, J. L. (2012). Combining the regression discontinuity design and propensity score-based weighting to improve causal inference in program evaluation. *Journal of evaluation in clinical practice*, 18(2):317–325.

Linden, A., Uysal, S. D., Ryan, A., and Adams, J. L. (2016). Estimating causal effects for multivalued treatments: a comparison of approaches. *Statistics in Medicine*, 35(4):534–552.

Linden, A. and Yarnold, P. R. (2017). Using classification tree analysis to generate propensity score weights. *Journal of evaluation in clinical practice*, 23(4):703–712.

Livieris, I. E., Kiriakidou, N., Stavroyiannis, S., and Pintelas, P. (2021, Art No 287). An advanced cnn-lstm model for cryptocurrency forecasting. *Electronics*, 10(3).

Long, C., Xu, H., Shen, Q., Zhang, X., Fan, B., Wang, C., Zeng, B., Li, Z., Li, X., and Li, H. (2020, Art. No 108961). Diagnosis of the coronavirus

disease (covid-19): rrt-pcr or ct? *European journal of radiology*, 126,. doi: 10.1016/j.ejrad.2020.108961.

Lopez-Rincon, A., Tonda, A., Mendoza-Maldonado, L., Mulders, D. G., Molenkamp, R., Perez-Romero, C. A., Claassen, E., Garssen, J., and Kraneveld, A. D. (2021). Classification and specific primer design for accurate detection of sars-cov-2 using deep learning. *Scientific reports*, 11(1):1–11.

Lu, R., Zhao, X., Li, J., Niu, P., Yang, B., Wu, H., Wang, W., Song, H., Huang, B., Zhu, N., et al. (2020). Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding. *The lancet*, 395(10224):565–574. doi: 10.1016/S0140-6736(20)30251-8.

Lunt, M. (2014). Selecting an appropriate caliper can be essential for achieving good balance with propensity score matching. *American journal of epidemiology*, 179(2):226–235.

Luo, Z., Gardiner, J. C., and Bradley, C. J. (2010). Applying propensity score methods in medical research: pitfalls and prospects. *Medical Care Research and Review*, 67(5):528–554.

Mackenzie, J. S. and Smith, D. W. (2020). COVID-19: a novel zoonotic disease caused by a coronavirus from China: what we know and what we don't. *Microbiology Australia*, 41(1):45–50. doi: 10.1071/MA20013.

Macukow, B. (2016). Neural networks–state of art, brief history, basic models and architecture. In *IFIP international conference on computer information systems and industrial management*, pages 3–14. Springer.

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9):1315–1316.

Mastakouri, A. A. and Schölkopf, B. (2020). Causal analysis of covid-19 spread in germany. *arXiv preprint arXiv:2007.11896*.

Mathew, A., Amudha, P., and Sivakumari, S. (2020). Deep learning techniques: An overview. In *International Conference on Advanced Machine Learning Technologies and Applications*, pages 599–608. Springer.

Matsudaira, J. D. (2008). Mandatory summer school and student achievement. *Journal of Econometrics*, 142(2):829–850.

McCaffrey, D. F., Ridgeway, G., and Morral, A. R. (2004). Propensity score estimation with boosted regression for evaluating causal effects in observational studies. *Psychological methods*, 9(4):403.

McConnell, K. J. and Lindner, S. (2019). Estimating treatment effects with machine learning. *Health services research*, 54(6):1273–1282.

McCrary, J. (2008). Manipulation of the running variable in the regression discontinuity design: A density test. *Journal of econometrics*, 142(2):698–714.

McDonald, R. J., McDonald, J. S., Kallmes, D. F., and Carter, R. E. (2013). Behind the numbers: propensity score analysis—a primer for the diagnostic radiologist. *Radiology*, 269(3):640–645.

Meng, Q., Catchpoole, D., Skillicom, D., and Kennedy, P. J. (2017). Relational autoencoder for feature extraction. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 364–371. IEEE.

Messeri, P. (2016). Counterfactual and causal inference: Methods and principles for social science research. *Canadian Studies in Population*, 43(1-2):169–171.

Metsky, H. C., Freije, C. A., Kosoko-Thoroddsen, T.-S. F., Sabeti, P. C., and Myhrvold, C. (2020). Crispr-based surveillance for covid-19 using genomically-comprehensive machine learning design. *BioRxiv*. doi: 10.1101/2020.02.26.967026.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J. (2020). Deep learning based text classification: A comprehensive review. *arXiv preprint arXiv:2004.03705*.

Mithas, S. and Krishnan, M. S. (2009). From association to causation via a potential outcomes approach. *Information Systems Research*, 20(2):295–313.

Morgan, S. L. (2013). *Handbook of causal analysis for social research*. Springer.

Morgan, S. L. and Winship, C. (2015). *Counterfactuals and causal inference*. Cambridge University Press.

Mu, X. and Xu, A. (2019). A Character-Level BiLSTM-CRF Model With Multi-Representations for Chinese Event Detection. *IEEE Access*, 7:146524–146532. doi:10.1109/ACCESS.2019.2943721.

Muggeo, V. M. et al. (2008). Segmented: an r package to fit regression models with broken-line relationships. *R news*, 8(1):20–25.

Muller, C. J. and MacLehose, R. F. (2014). Estimating predicted probabilities from logistic regression: different methods correspond to different target populations. *International journal of epidemiology*, 43(3):962–970.

Murad, A. and Pyun, J.-Y. (2017). Deep recurrent neural networks for human activity recognition. *Sensors*, 17(11):2556.

Naimi, A. I. and Kaufman, J. S. (2015). Counterfactual theory in social epidemiology: reconciling analysis and action for the social determinants of health. *Current Epidemiology Reports*, 2(1):52–60.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., and Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of big data*, 2(1):1–21.

Nguimbous, Y. N., Ksantini, R., and Bouhoula, A. (2019). Anomaly-based intrusion detection using auto-encoder. In *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–5. IEEE.

Nguyen, N. G., Tran, V. A., Ngo, D. L., Phan, D., Lumbanraja, F. R., Faisal, M. R., Abapihi, B., Kubo, M., Satou, K., and others (2016). DNA sequence classification by convolutional neural network. *Journal of Biomedical Science and Engineering*, 9(05). doi:10.4236/jbise.2016.95021.

Nian, G., Peng, B., Sun, D. J., Ma, W., Peng, B., and Huang, T. (2020). Impact of covid-19 on urban mobility during post-epidemic period in megacities: From the perspectives of taxi travel and social vitality. *Sustainability*, 12(19):7954.

Nichols, A., McBride, L., et al. (2019). Propensity scores and causal inference using machine learning methods. In *Presentation in the Track session*

"Machine Learning in Applied Economics" at the annual meeting of the Agricultural and Applied Economics Association (AAEA), Atlanta, July, pages 21–23.

Niekum, S., Osentoski, S., Atkeson, C. G., and Barto, A. G. (2014). Champ: Changepoint detection using approximate model parameters. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST.

Nika, V., Babyn, P., and Zhu, H. (2014). Change detection of medical images using dictionary learning techniques and principal component analysis. *Journal of Medical Imaging*, 1(2):024502. doi: 10.1117/1.JMI.1.2.024502.

Nilsson, M. (2013). Causal inference in a 22 factorial design using generalized propensity score.

Nørgaard, M., Ehrenstein, V., and Vandenbroucke, J. P. (2017). Confounding in observational studies based on large health care databases: problems and potential solutions–a primer for the clinician. *Clinical epidemiology*, 9:185.

Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.

Olmos, A. and Govindasamy, P. (2015). Propensity scores: a practical introduction using r. *Journal of MultiDisciplinary Evaluation*, 11(25):68–88.

Ordóñez, F. J. and Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1). doi: 10.3390/s16010115.

Osman, S. M. I. and Sakib, N. (2020). Stay home save lives: A machine learning approach to causal inference to evaluate impact of social distancing in the us.

Otter, D. W., Medina, J. R., and Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624.

Padmanabhan, J. and Johnson Premkumar, M. J. (2015). Machine learning in automatic speech recognition: A survey. *IETE Technical Review*, 32(4):240–251.

Pan, W. and Bai, H. (2018). Propensity score methods for causal inference: an overview. *Behaviormetrika*, 45(2):317–334.

Papay, J. P., Murnane, R. J., and Willett, J. B. (2014). High-school exit examinations and the schooling decisions of teenagers: Evidence from regression-discontinuity approaches. *Journal of research on educational effectiveness*, 7(1):1–27.

Papay, J. P., Willett, J. B., and Murnane, R. J. (2011). Extending the regression-discontinuity approach to multiple assignment variables. *Journal of Econometrics*, 161(2):203–207.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318.

Pearl, J. et al. (2009). Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146.

Pepelyshev, A. and Polunchenko, A. S. (2015). Real-time financial surveillance via quickest change-point detection methods. *arXiv preprint arXiv:1509.01570*. doi: 10.4310/SII.2017.v10.n1.a9.

Petersen, M. L., Porter, K. E., Gruber, S., Wang, Y., and Van Der Laan, M. J. (2012). Diagnosing and responding to violations in the positivity assumption. *Statistical methods in medical research*, 21(1):31–54.

Porter, K. E., Reardon, S. F., Unlu, F., Bloom, H. S., and Cimpian, J. R. (2017). Estimating causal effects of education interventions using a two-rating regression discontinuity design: Lessons from a simulation study and an application. *Journal of Research on Educational Effectiveness*, 10(1):138–167.

Poulos, J. (2017). Rnn-based counterfactual timeseries prediction. *arXiv preprint arXiv:1712.03553*.

Prosperi, M., Guo, Y., Sperrin, M., Koopman, J. S., Min, J. S., He, X., Rich, S., Wang, M., Buchan, I. E., and Bian, J. (2020). Causal inference and counterfactual prediction in machine learning for actionable healthcare. *Nature Machine Intelligence*, 2(7):369–375.

Quang, D. and Xie, X. (2016). DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic acids research*, 44(11). doi: 10.1093/nar/gkw226.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Radke, R. J., Andra, S., Al-Kofahi, O., and Roysam, B. (2005). Image change detection algorithms: a systematic survey. *IEEE transactions on image processing*, 14(3):294–307. doi: 10.1109/TIP.2004.838698.

Raghunathan, K. (2010). Demystifying the american graduate admissions process. *StudyMode. com*.

Ramachandra, V. (2018). Deep learning for causal inference. *arXiv preprint arXiv:1803.00149*.

Ramachandra, Vikas and Sun, Haoqiao (2020). Causal inference for covid-19 interventions. Last accessed 15 October 2020.

Randhawa, G. S., Hill, K. A., and Kari, L. (2020). MLDSP-GUI: an alignment-free standalone tool with an interactive graphical user interface for DNA sequence comparison and analysis. *Bioinformatics*, 36(7):2258–2259. doi:10.1093/bioinformatics/btz918.

Randhawa, G. S., Soltysiak, M. P., El Roz, H., de Souza, C. P., Hill, K. A., and Kari, L. (2020, Art. No e0232391). Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: COVID-19 case study. *Plos one*, 15(4). doi:10.1371/journal.pone.0232391.

Reardon, S. F. and Robinson, J. P. (2012). Regression discontinuity designs with multiple rating-score variables. *Journal of research on Educational Effectiveness*, 5(1):83–104.

Reiter, J. (2000). Using statistics to determine causal relationships. *The American Mathematical Monthly*, 107(1):24–32.

Rhanoui, M., Mikram, M., Yousfi, S., and Barzali, S. (2019). A CNN-BiLSTM Model for Document-Level Sentiment Analysis. *Machine Learning and Knowledge Extraction*, 1(3):832–847. doi:10.3390/make1030048.

Robins, J. M., Hernan, M. A., and Brumback, B. (2000). Marginal structural models and causal inference in epidemiology.

Rohrbeck, C. (2013). Detection of changes in variance using binary segmentation and optimal partitioning.

Rosenbaum, P. R. (2002). Observational studies. In *Observational studies*, pages 1–17. Springer, New York, NY,.

Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Roux, S., Adriaenssens, E. M., Dutilh, B. E., Koonin, E. V., Kropinski, A. M., Krupovic, M., Kuhn, J. H., Lavigne, R., Brister, J. R., Varsani, A., and others (2019). Minimum information about an uncultivated virus genome (MIUViG). *Nature Biotechnology*, 37(1):29–37. doi:10.1038/nbt.4306.

Roy, P. K., Singh, J. P., and Banerjee, S. (2020). Deep learning to filter sms spam. *Future Generation Computer Systems*, 102:524–533. doi: 10.1016/j.future.2019.09.001.

Roy, R. and George, K. T. (2017). Detecting insurance claims fraud using machine learning techniques. In *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pages 1–6. IEEE.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688.

Rubin, D. B. (2003). Basic concepts of statistical inference for causal effects in experiments and observational studies. *Cambridge, MA: Harvard University, Department of Statistics*.

Rubin, D. B. (2004). Teaching statistical inference for causal effects in experiments and observational studies. *Journal of Educational and Behavioral Statistics*, 29(3):343–367.

Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331.

Rubin, D. B. et al. (2006). Causal inference through potential outcomes and principal stratification: application to studies with "censoring" due to death. *Statistical Science*, 21(3):299–309.

Sahu, S. K. and Anand, A. (2018). Drug-drug interaction extraction from biomedical texts using long short-term memory network. *Journal of biomedical informatics*, 86:15–24. doi:10.1016/j.jbi.2018.08.005.

Sainath, T. N., Vinyals, O., Senior, A., and Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12–18. IEEE. doi:10.1109/ICASSP.2015.7178838.

Schilling, F. (2016). *The effect of batch normalization on deep convolutional neural networks*.

Schreiber, T. (2000). Measuring information transfer. *Physical review letters*, 85(2):461.

Schuler, M. S. and Rose, S. (2017). Targeted maximum likelihood estimation for causal inference in observational studies. *American journal of epidemiology*, 185(1):65–73.

Seo, S., Oh, M., Park, Y., and Kim, S. (2018). DeepFam: deep learning based alignment-free method for protein family modeling and prediction. *Bioinformatics*, 34(13):i254–i262. doi:10.1093/bioinformatics/bty275.

Setodji, C. M., McCaffrey, D. F., Burgette, L. F., Almirall, D., and Griffin, B. A. (2017). The right tool for the job: choosing between covariate balancing and generalized boosted model propensity scores. *Epidemiology (Cambridge, Mass.)*, 28(6):802.

Setoguchi, S., Schneeweiss, S., Brookhart, M. A., Glynn, R. J., and Cook, E. F. (2008). Evaluating uses of data mining techniques in propensity score estimation: a simulation study. *Pharmacoepidemiology and drug safety*, 17(6):546–555.

Sheather, S. J. and Marron, J. S. (1990). Kernel quantile estimators. *Journal of the American Statistical Association*, 85(410):410–416.

Shpyrko, V. and Koval, B. (2019). Fraud detection models and payment transactions analysis using machine learning. In *SHS Web of Conferences*, volume 65, page 02002. EDP Sciences.

Siami-Namini, S., Tavakoli, N., and Namin, A. S. (2019). The performance of lstm and bilstm in forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3285–3292. IEEE. doi:10.1109/BigData47090.2019.9005997.

Siloko, I., Ikpotokin, O., and Siloko, E. (2019). A key note on performance of smoothing parameterizations in kernel density estimation. *Tanzania Journal of Science*, 45(1):1–8.

Singaravel, S., Suykens, J., and Geyer, P. (2018). Deep-learning neural-network architectures and methods: Using component-based models in building-design energy prediction. *Advanced Engineering Informatics*, 38:81–90.

Singh, A. (2017). Anomaly detection for temporal data using long short-term memory (lstm).

Sivaram, A., Das, L., and Venkatasubramanian, V. (2020). Hidden representations in deep neural networks: Part 1. classification problems. *Computers & Chemical Engineering*, 134:106669.

Śmieja, M., Struski, Ł., Tabor, J., Zieliński, B., and Spurek, P. (2018). Processing of missing data by neural networks. In *Advances in Neural Information Processing Systems*, pages 2719–2729.

South African Government (2020). President cyril ramaphosa: South africa's response to coronavirus covid-19 pandemic. the presidency, 23 apr. https://www.gov.za/speeches/president-cyril-ramaphosa-south-africas-response-coronavirus-covid-19-pandemic-23-apr-2020.

Sovilj, D., Budnarain, P., Sanner, S., Salmon, G., and Rao, M. (2020). A comparative evaluation of unsupervised deep architectures for intrusion detection in sequential data streams. *Expert Systems with Applications*, page 113577.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014a). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014b). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958. Publisher: JMLR. org.

Stuart, E. A. (2010). Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1.

Stuart, E. A., Lee, B. K., and Leacy, F. P. (2013). Prognostic score–based balance measures can be a useful diagnostic for propensity score methods in comparative effectiveness research. *Journal of clinical epidemiology*, 66(8):S84–S90.

Tahmasbi, R. and Rezaei, S. (2008). Change point detection in garch models for voice activity detection. *IEEE transactions on audio, speech, and language processing*, 16(5):1038–1046. doi: 10.1109/TASL.2008.922468.

Tan, Y., Jin, B., Nettekoven, A., Chen, Y., Yue, Y., Topcu, U., and Sangiovanni-Vincentelli, A. (2019). An encoder-decoder based approach for anomaly detection with application in additive manufacturing. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1008–1015. IEEE.

Taylor, W. A. (2000). Change-point analysis: a powerful new tool for detecting changes.

Thistlethwaite, D. L. and Campbell, D. T. (1960). Regression-discontinuity analysis: An alternative to the ex post facto experiment. *Journal of Educational psychology*, 51(6):309.

Titiunik, R. (2015). Can big data solve the fundamental problem of causal inference? *PS: Political Science & Politics*, 48(1):75–79.

Toulis, P., Volfovsky, A., and Airoldi, E. M. (2018). Propensity score methodology in the presence of network entanglement between treatments. *arXiv preprint arXiv:1801.07310*.

Tran, K. P., Du Nguyen, H., and Thomassey, S. (2019). Anomaly detection using long short term memory networks and its applications in supply chain management. *IFAC-PapersOnLine*, 52(13):2408–2412.

Tsapeli, F., Musolesi, M., and Tino, P. (2017). Non-parametric causality detection: An application to social media and financial data. *Physica A: Statistical Mechanics and its Applications*, 483:139–155.

Urban, C. J. and Gates, K. M. (2021). Deep learning: A primer for psychologists. *Psychological Methods*.

van den Oord, A. G. A., Dieleman, S. E. L., Kalchbrenner, N. E., Simonyan, K., Vinyals, O., and Espeholt, L. (2020). Speech recognition using convolutional neural networks. US Patent 10,586,531.

Van der Berg, S. (2011). Current poverty and income distribution in the context of south african history. *Economic History of Developing Regions*, 26(1):120–140.

Vansteelandt, S. and Daniel, R. M. (2014). On regression adjustment for the propensity score. *Statistics in medicine*, 33(23):4053–4072.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives*, 28(2):3–28.

Varian, H. R. (2016). Causal inference in economics and marketing. *Proceedings of the National Academy of Sciences*, 113(27):7310–7315.

Verma, J. and Abdel-Salam, A.-S. G. (2019). *Testing statistical assumptions in research*. John Wiley & Sons.

Von Hippel, P. (2015). Linear vs. logistic probability models: Which is better, and when. *Statistical Horizons*.

Wang, J. and Li, S. (2018). Comparing the influence of depth and width of deep neural network based on fixed number of parameters for audio event detection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2681–2685. IEEE.

Wang, Y., Yao, H., and Zhao, S. (2016). Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242.

Wehle, H.-D. (2017). Machine learning, deep learning, and ai: What's the difference? In *Internationan Conference on Data scientist innovation day, Bruxelles, Belgium*.

Westreich, D., Lessler, J., and Funk, M. J. (2010). Propensity score estimation: machine learning and classification methods as alternatives to logistic regression. *Journal of clinical epidemiology*, 63(8):826.

Whata, A. and Chimedza, C. (2021a). Deep learning for sars cov-2 genome sequences. *IEEE Access*, 9:59597–59611.

Whata, A. and Chimedza, C. (2021b). A machine learning evaluation of the effects of south africa's covid-19 lockdown measures on population mobility. *Machine Learning and Knowledge Extraction*, 3(2):481–506.

Wijeyakulasuriya, D. A., Eisenhauer, E. W., Shaby, B. A., and Hanks, E. M. (2020). Machine learning for modeling animal movement. *Plos one*, 15(7):e0235750.

Williams, T. C., Bach, C. C., Matthiesen, N. B., Henriksen, T. B., and Gagliardi, L. (2018). Directed acyclic graphs: a tool for causal studies in paediatrics. *Pediatric research*, 84(4):487–493.

Wilson, D. R. and Martinez, T. R. (2001). The need for small learning rates on large problems. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, volume 1, pages 115–119. IEEE. doi:10.1109/IJCNN.2001.939002.

Winship, C. and Morgan, S. L. (1999). The estimation of causal effects from observational data. *Annual review of sociology*, 25(1):659–706.

Wolpher, M. (2018). Anomaly detection in unstructured time series datausing an lstm autoencoder.

Wong, V. C., Steiner, P. M., and Cook, T. D. (2013). Analyzing regression-discontinuity designs with multiple assignment variables: A comparative study of four estimation methods. *Journal of Educational and Behavioral Statistics*, 38(2):107–141.

Woo, M.-J., Reiter, J. P., and Karr, A. F. (2008). Estimation of propensity scores using generalized additive models. *Statistics in medicine*, 27(19):3805–3816.

Wooldridge, J. (2009). Estimating average treatment effects: Unconfoundedness. *Lecture notes. BGSE/IZA Course in Microeconometrics*.

World Health Organization (2020). Who director-general's opening remarks at the media briefing on covid-19 - 11 march 2020 for websites. Last accessed 22 November 2020.

Wright, R. E. (1995). Logistic regression.

Wyss, R., Ellis, A. R., Brookhart, M. A., Girman, C. J., Jonsson Funk, M., LoCasale, R., and Stürmer, T. (2014). The role of prediction modeling in propensity score estimation: an evaluation of logistic regression, bcart, and the covariate-balancing propensity score. *American journal of epidemiology*, 180(6):645–655.

Wyss, R., Schneeweiss, S., Van Der Laan, M., Lendle, S. D., Ju, C., and Franklin, J. M. (2018). Using super learner prediction modeling to improve high-dimensional propensity score estimation. *Epidemiology*, 29(1):96–106.

Yan, X., Gilani, S. Z., Feng, M., Zhang, L., Qin, H., and Mian, A. (2020). Self-supervised learning to detect key frames in videos. *Sensors*, 20(23):6941.

Yang, Y., Yang, M., Yuan, J., Wang, F., Wang, Z., Li, J., Zhang, M., Xing, L., Wei, J., Peng, L., et al. (2020). Laboratory diagnosis and monitoring the viral shedding of sars-cov-2 infection. *The Innovation*, 1(3).

Yao, L., Chu, Z., Li, S., Li, Y., Gao, J., and Zhang, A. (2020). A survey on causal inference. *arXiv preprint arXiv:2002.02770*.

Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018). Recent trends in deep learning based natural language processing. *ieee Computational Intelligence Magazine*, 13(3):55–75. doi:10.1109/MCI.2018.2840738.

Yuan, Y., Ding, X., and Bar-Joseph, Z. (2020). Causal inference using deep neural networks. *arXiv preprint arXiv:2011.12508*.

Zeileis, A., Leisch, F., Hornik, K., and Kleiber, C. (2001). strucchange. an r package for testing for structural change in linear regression models.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zeng, H., Edwards, M. D., Liu, G., and Gifford, D. K. (2016). Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics*, 32(12):i121–i127. doi:10.1093/bioinformatics/btw255.

Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P., and Zhang, J. (2014). Convolutional neural networks for human activity recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services*, pages 1–5. IEEE. doi: 10.4108/icst.mobicase.2014.257786.

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., and Chawla, N. V. (2019a). A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1409–1416.

Zhang, H., Hung, C.-L., Liu, M., Hu, X., and Lin, Y.-Y. (2019b). NCNet: Deep Learning Network Models for Predicting Function of Non-coding DNA. *Frontiers in genetics*, 10. doi: 10.3389/fgene.2019.00432.

Zhang, S., Zheng, D., Hu, X., and Yang, M. (2015). Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia conference on language, information and computation*, pages 73–78.

Zhang, Y., Qiao, S., Ji, S., and Li, Y. (2020). DeepSite: bidirectional LSTM and CNN models for predicting DNA–protein binding. *International Journal of Machine Learning and Cybernetics*, 11(4):841–851. doi:10.1007/s13042-019-00990-x.

Zhang, Z. and Sabuncu, M. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pages 8778–8788.

Zhao, J., Yuan, Q., Wang, H., Liu, W., Liao, X., Su, Y., Wang, X., Yuan, J., Li, T., Li, J., et al. (2020a). Antibody responses to sars-cov-2 in patients with novel coronavirus disease 2019. *Clinical infectious diseases*, 71(16):2027–2034. doi: 10.1093/cid/ciaa344.

Zhao, P., Su, X., Ge, T., and Fan, J. (2016). Propensity score and proximity matching using random forest. *Contemporary clinical trials*, 47:85–92.

Zhao, S., van Dyk, D. A., and Imai, K. (2020b). Propensity score-based methods for causal inference in observational studies with non-binary treatments. *Statistical Methods in Medical Research*, 29(3):709–727.

Zhong, Y. and Zhao, M. (2020). Research on deep learning in apple leaf disease recognition. *Computers and Electronics in Agriculture*, 168:105146.

Zhou, J. and Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods*, 12(10):931–934. doi: 10.1038/nmeth.3547.

Zhu, Y., Coffman, D. L., and Ghosh, D. (2015). A boosting algorithm for estimating generalized propensity scores with continuous treatments. *Journal of causal inference*, 3(1):25–40.

Zielezinski, A., Vinga, S., Almeida, J., and Karlowski, W. M. (2017, Art. No 186). Alignment-free sequence comparison: benefits, applications, and tools. *Genome biology*, 18(1). doi:10.1186/s13059-017-1319-7.

Zou, J., Huss, M., Abid, A., Mohammadi, P., Torkamani, A., and Telenti, A. (2019). A primer on deep learning in genomics. *Nature genetics*, 51(1):12–18. doi:10.1038/s41588-018-0295-5.

Zubizarreta, J. R. (2015). Stable weights that balance covariates for estimation with incomplete outcome data. *Journal of the American Statistical Association*, 110(511):910–922.