IMPACT OF ANOMALY DETECTION ON MACHINE LEARNING MODELS IN THE PREDICTION OF COVID-19 USING CATEGORICAL DATA

ARVIND R NAIR

A thesis submitted in partial fulfillment of the requirements of Liverpool John Moores University for the degree of Masters in Data Science

DECEMBER 2021

DEDICATION

I dedicate this work to my wife Sandhya for her constant encouragement, her consistent belief in my abilities and for the many compromises she made for my benefit so that I could complete this journey of learning. I also dedicate this to my father Raju and my mother Chitra, for their wisdom and advice and for being my constant pillars of support that got me through the hardships of life.

ACKNOWLEDGEMENTS

I would like to thank my Thesis Supervisor Amulya Dash for his guidance with all the technical queries I have put forward and for being easily accessible. The support he has provided me has helped smoothen a research process that has in nearly all other aspects been a challenging endeavour. I also would like to thank IIIT-B and LJMU for the many learnings I have had during the course of my Master's degree.

ABSTRACT

Our world is in the middle of a pandemic due to Covid-19, a highly contagious disease with limited options for treatment. Early detection and isolation are considered the primary strategy in containing its spread. This will require a capacity to make early and informed decisions with the highest accuracy possible from available data on patient characteristics and other data points related to diagnostics and early symptoms. As part of this endeavour, our study tested three Anomaly Detection (AD) techniques separately on the Covid-19 dataset that is categorical in nature and evaluated its effects on the performance of several Machine Learning (ML) models, which included Logistic Regression, Decision Tree, Random Forest, XGBoost and Support Vector Machines. The evaluation of ML models was based on their performance prior to anomaly removal (baseline model) and post-removal using the AD techniques (AD-based models). The expectation was that AD-based ML model training is a more efficient process and could result in significant increases in prediction performance. While similar and related methodologies may have been previously conducted on quantitative data, our study tested the effects of AD on a Covid-19 categorical dataset. Our evaluation showed that AD indeed can have a positive impact on ML models as it was observed that gains could be made in the predictive power of the models.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ML……..	Machine Learning
AI……..	Artificial Intelligence
AD..……	Anomaly Detection
DL..……	Deep Learning
LSTM..……..	Long Short-Term Memory
CNN..……..…	Convolutional Neural Network
RF…….	Random Forest
SVM..……	Support Vector Machine
LR..……	Logistic Regression
GBM..……	Gradient Boosting Machine
NB..……	Naïve Bayes
DT..……	Decision Tree
KNN..……	K-Nearest Neighbours
ROAD..……	Ranking-based Outlier Analysis and Detection
Covid-19..…….	Coronavirus Disease 2019
RT-PCR..…….	Real-Time Polymerase Chain Reaction
CT..……	Computed Tomography
MRI..……	Magnetic Resonance Imaging
EDA..……	Exploratory Data Analysis
SMOTE..……..	Synthetic Minority Class Oversampling Technique
XGBoost……	Extreme Gradient Boosting
TP..…….	True Positive
TN..…….	True Negative
FP..…….	False Positive
FN..…….	False Negative
HBOS..……	Histogram-based Outlier Score
XGBOD..……	Extreme Gradient Boosting Outlier Detection
RBDA..……	Rank-Based Detection Algorithm
IF..……..	Isolation Forest
EIF..…….	Extended Isolation Forest
TOS..…….	Transformed outlier scores

CARE..…….	Cumulative Agreement Rates Ensemble
SARS-Cov-2..…...	Severe Acute Respiratory Syndrome Coronavirus 2
ROC..…….	Receiver Operating Characteristic
RBF..…….	Radial Basis Function

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

The advent of the Covid-19 pandemic resulted in a worldwide disruption of life as we know it. Wide-ranging impacts have been observed in healthcare systems, economic health, gender equality, public health, and many others. Research work in the year 2020 revolved significantly around Covid-19. One estimate suggested that around 4% of the world's research was devoted to coronavirus with the bulk of it initially focused on disease spread, hospitalization outcomes and diagnostics, and mitigatory strategies such as testing (Else, 2020).

Artificial Intelligence (AI)/ML has been applied significantly in the epidemiological research of Covid-19. ML has been used to identify variables, explain interconnections that help understand the key reasons behind Covid-19 cases and related deaths. Deep Learning (DL) techniques, such as Long short-term memory (LSTM), were found effective in detecting Covid-19 using time series data. Chest X-rays and CT scans are popular in screening for Covid-19 using Convolutional Neural Network (CNN) or hybrid DL techniques. AI/ML has also been applied in other interesting ways such as finding new molecules that help confirm Covid-19 cases, embedded AI in cameras and smartphones to detect infected people, and even use of drones for transportation of food and medicines in areas with infection (Dogan et al., 2021).

The research into AI/ML approaches to solve issues related to Covid-19 grows by the day but the use of AD to assist in the process remains relatively low. Anomalies or outliers are objects that are not consistent with the pattern of the majority of the instances in the dataset. It can be quite useful to detect anomalies in data because they are capable of distorting analysis and predictive power of trained models thereby affecting decision making. Significant research has been done in the field of anomaly/outlier detection since its study is of importance to multiple disciplines which include statistics, data mining, and machine learning among others.

AD has innumerable applications in various domains such as cybersecurity, fraud detection, healthcare, medical diagnosis, and disease outbreak detection to name a few (Ruff et al., 2021). AD therefore can be considered as a suitable technique to be employed in improving ML models for the prediction of Covid-19.

In our study, we propose to develop a methodology that applies AD as an integral step in the training of ML models to predict Covid-19 early on using a symptomatic dataset. As part of the research, we will study the impact of AD on a categorical Covid-19 dataset and how it affects the predictive power of the trained models. Given that detection and containment of Covid-19 remains an important strategy, ML models with high accuracy could provide the necessary advantage for frontline healthcare workers to act early on.

1.2 Problem Statement

As part of our research, we looked at recent works related to the role of ML in Covid-19 detection and mitigation. We observed that shallow ML techniques such as Support Vector Machines (SVM) and Random Forest (RF) are popular and employed for scenarios such as early detection of Covid-19 and prioritized usage of RT-PCR tests. We also saw that DL techniques along with medical imaging are a common approach in disease detection and understanding its spread. However, as these approaches lacked the use of Anomaly Detection (AD), we looked at different categories of possible AD techniques that could be applied such as Isolation Forest and its extensions (Hariri et al., 2021). We found that the use of AD in categorical data was limited due to varying definitions of what qualifies as an anomaly. We further noted that the use of AD in a Covid-19 dataset that is categorical in nature is yet to be explored. The impact of AD on predictive models that detect Covid-19 at an early stage is something that is worth looking into.

1.3 Aim and Objectives

This research is aimed at understanding the impact of Anomaly Detection on the prediction of Covid-19 using symptomatic data. The goal is to investigate whether Anomaly Detection can improve the quality of data that is categorical in nature and thereby increase the effectiveness of machine learning models in predicting Covid-19 at an early stage.

The objectives of the research are stated as follows:

- To suggest an Anomaly Detection technique that is suitable for application on the Covid-19 dataset that is categorical in nature.
- To evaluate the impact of the Anomaly Detection technique on the predictive power of five different machine learning models, specifically Logistic Regression, Decision Tree, Random Forest, XGBoost, and Support Vector Machines.
- To improve the prediction of Covid-19 at an early stage using the symptomatic data.

1.4 Scope of the Study

The dataset includes symptomatic data from a few provinces in China. The data, as a result, is not representative of the Covid-19 symptoms in other parts of the world where different virus mutations, social circumstances and differences in healthcare systems could result in variations in symptomatic data. The dataset is predominantly categorical in nature and therefore the Anomaly Detection method chosen may not work for quantitative data. Only shallow Machine Learning models, specifically Logistic Regression, Decision Trees, Random Forest, XGBoost, and Support Vector Machine are being employed in this study. Deep Learning techniques are not being used since the quantity of data is insufficient to train them. The study is intended to assist and complement the diagnostic capabilities of the healthcare system and its frontline workers only for Covid-19. The process can inspire or be replicated but cannot be deployed as-is for the detection and mitigation of other diseases.

1.4.1 Deliverables

This section will describe the deliverables that are necessary to achieve the stated objectives of this study.

1. Explore various AD techniques for categorical data.
2. Feasibility and implementation test of the AD technique on our dataset.
3. Exploratory Data Analysis to profile the Covid-19 dataset.
4. Compare differences in prediction power of models prior to and post-removal of anomalies.
5. Train predictive models that have higher predictive scores on symptomatic Covid-19 data

1.5 Significance of the Study

Treatment of Covid-19 is at best experimental and vaccine rollouts are still at a very early stage. As a result, early detection and containment of the Covid-19 spread remain one of the most crucial strategies available to stave off its worst impacts on healthcare at an individual as well as infrastructure level. Machine Learning-based detection of Covid-19 at an early stage plays an important role in effective prognosis and decision-making for healthcare workers. By using Anomaly Detection to improve the effectiveness of ML models, the strategy of early detection and mitigation of Covid -19 could be strengthened and made more efficient.

1.6 Structure of the Study

Our thesis is structured in the following manner. Chapter 1 includes the background of our study and discusses our problem statement (Sections 1.1 and 1.2 respectively). Section 1.3 details the aim and objectives of our research. Section 1.4 provides information on the scope of our study and subsection 1.4.1 details the deliverables. The study's significance is provided in Section 1.5.

Chapter 2 reviews the literature available on Covid-19 and Anomaly Detection (AD) that highlights the gaps described in the problem statement from Chapter 1. Sections 2.2 and 2.3 look at the role of Machine Learning and Deep Learning in the detection and mitigation of Covid-19. Sections 2.4 and 2.5 and their subsections dive deep into the world of AD and look at various Machine Learning & statistical techniques designed for various use cases. We particularly pay attention to ensemble-based learning techniques such as Isolation Forest for their versatility as well as unsupervised learning techniques for their usefulness in data with unlabeled anomalies. Section 2.6 looks specifically at the progress of AD in categorical data. The role of AD in the detection of Covid-19 is looked at in Section 2.7.

Chapter 3 looks at the research methodology and the proposed AD framework. The focus of this chapter is Section 3.2 which looks at our research approach. This section looks at different modules of our proposed framework including Exploratory Data Analysis, Data Preparation, Anomaly Detection, model predictions, and performance evaluation, and discusses the different techniques we employ in each of them in detail.

Chapter 4 looks at the specifics of our model development process as we execute the proposed methodology. Section 4.2 shares our findings from the EDA, which primarily includes univariate and bivariate analyses. Section 4.3 focuses on data preparation steps prior to model building. It looks at the test train split ratio, how the AD techniques were deployed, and details the application of class imbalance treatment. Section 4.4 is centered on the model building and tuning process.

Chapter 5 discusses the results and evaluations post the model development stage. Section 5.2 compares and evaluates the results for each predictive model along with the impact of the AD techniques. Section 5.3 interprets our evaluations and how they fulfill our research objectives.

Chapter 6 summarizes and provides conclusions in Section 6.2 that are relevant to our objectives. Section 6.3 briefs on the knowledge contributions of this research. Section 6.4 makes recommendations for future research.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Our literature review has been conducted to understand the progress in ML-based research related to Covid-19. We also endeavour to get a grasp of Anomaly Detection (AD) at a conceptual level, review various approaches to AD that include statistical and ML-based techniques and take a look at how AD is already playing a role in Covid-19 research.

2.2 Machine Learning in Covid-19 detection

Much research has been published that involves ML approaches in the detection and mitigation of Covid-19. A review of the AI/ML methods being used in the detection of Covid-19 found that over 50% of studies chose Random Forest (RF) as at least one of their ML classifiers because of its capacity to pick good features for classification. Support Vector Machines (SVM) was another ML method that was found popular in Covid-19 studies for similar reasons (Dogan et al., 2021). In an endeavour to speed up decision making on treatments and isolation strategies, machine learning classification algorithms, specifically RF, Gradient Boosting Machine (GBM), XGBoost, SVM, and Decision Tree (DT) were deployed to identify which symptoms of a patient were significant predictors of Covid-19 (such as fever, cough, runny nose) and also to understand which among the models were suitable to consistent predictions across age brackets (Ahamad et al., 2020).

Another study proposed the use of ML algorithms to help with prioritization and targeted use of RT-PCR tests to identify Covid-19. The study suggested the use of ML to predict the risks of a positive diagnosis using data from routine tests in situations where the RT-PCR test results took too long. In this case too, RF and SVM are featured along with others like Logistic Regression (LR), GBM, and Neural Networks (Afm et al., 2020).

2.3 Deep Learning & Medical Imaging in Covid-19 detection

Because of the prevalence of medical image data such as X-rays, CT scans, and MRI, DL techniques such as CNN models have been a popular choice either by themselves or together with other methods like AlexNet and LSTM. One of them was a combined DL system that was tested on an X-ray dataset which included samples of Covid-19 patients among others. The system consisted of a CNN network and LSTM, the former of which was used to extract features from the X-ray images and the latter to make predictions based on said features and was able to achieve a significant accuracy of 99.4% (Islam et al., 2020).

Another proposed solution was COVIDetectioNet, a novel system that used a three-stage process. A pretrained AlexNet architecture was employed as a feature extractor followed by the use of the Relief algorithm in identifying the most efficient of these derived features. Finally, the SVM method was used to classify these features (Turkoglu, 2021). A detailed study of several traditional ML models (such as SVM, KNN, RF, DT), as well as DL models (such as GoogleNet, ResNet, Xception), was conducted to understand the automatic identification of Covid-19 using X-ray images. The novelty here was a unique process that was used to select the ideal transfer learning model which was then employed to extract features and train both ML and DL models (Mohammed et al., 2021). But none of the research discussed so far has considered using Anomaly Detection (AD) methods to improve model predictions.

2.4 Anomaly Detection & Machine Learning

In the area of machine learning, at a high level, one way to look at AD techniques is to categorize them based on the traits of the data - into supervised, semi-supervised, and unsupervised.

2.4.1 Supervised Anomaly Detection

In supervised AD, all instances of the dataset are initially labelled to indicate whether they are outliers or not. The dataset is then divided into train and test followed by the classification models being trained to predict the labels on the test set. SVM and Bayesian Networks are a couple of models used in such cases.

SVM was employed as part of an AD method in wireless sensor networks where the model was able to classify node data as a local outlier, network outlier, or cluster outlier (Mohamed and Kavitha, 2011). Maritime security was another interesting domain where supervised AD was found useful. With the help of data available via the Automated Identification System, Bayesian Networks could detect deviations from regular movement patterns of vessels and thereby assist in the identification of security risks or illegal trafficking (Mascaro et al., 2014). The disadvantage in the supervised approach is they require labelled instances which is a manual process, is often scarcely available, and is expensive to make. Furthermore, identifying new patterns of outliers becomes difficult when the existing models are trained only on the known ones.

2.4.2 Semi-supervised Anomaly Detection

Semi-supervised AD typically involves the use of training datasets where (i) both regular samples as well as outliers are made available with labels along with unlabelled data or (ii) normal data is exclusively made available. The model learns from the labelled data and is then expected to find deviating instances from an unlabelled training set. An example of this is a hybrid model, consisting of a Deep AutoEncoder and an ensemble of KNN based outlier detectors, that was proposed in which the former would transform a high dimensional dataset into a compact form post which the latter handled the task of detecting outliers (Song et al., 2017). The hybrid setup provided the advantage of using only a portion of the training set due to its compact nature of distribution post-transformation as well as helped reduce computational costs.

2.4.3 Unsupervised Anomaly Detection

However, Unsupervised learning has been the leading approach to AD. This is because in typical circumstances the availability of outlier data with indicative labels is scarce. In the event that labelled data is available, they are hardly sufficient to cover different characteristics of outliers thus limiting the effectiveness of the supervised approach (Ruff et al., 2021). An example would be the use of Isolation Forest employed as an AD solution in the Information Security domain.

In one such scenario, an anomalous user detection system was devised by developing a baseline model of an enterprise's user activity by using Isolation Forest to calculate a threshold anomaly score. When a new user enters the system, a score is calculated and compared with the threshold score, which if crossed results in the user being flagged (Sun et al., 2016). K-Means clustering was used to improve heart disease prediction by employing it as an AD solution. This was proven to be effective by comparing accuracies of various classification models - namely SVM, Naïve Bayes (NB), LR, RF, and KNN – prior to and post-removal of outliers tagged by K-Means (Ripan et al., 2021).

Another example is the Histogram-based Outlier Detection (HBOS) which starts by constructing a histogram for each of the features within the dataset. For numerical attributes, two approaches are available – Static bin-width histograms and dynamic-bin width histograms. The first is the typical approach where the histograms constructed are of *k* bins of equal width for a given range of values. Sample frequency is used to determine the density or the height of the bins – a higher density would mean a lesser chance that the samples within the bin are anomalies. The second approach involves sorting the values followed by $\frac{N}{k}$ values being allocated to each bin where *k* is the total bins and *N* is the count of instances. This formula keeps the width the same for all the bins while the height can be a measure of the frequency of the data points which in the case of outliers will be lower. HBOS is observed to work well in identifying global outliers but performed poorly for local outlier identification (Goldstein and Dengel, 2012).

2.5 Anomaly Detection for High Dimensional Data

Another way to look at AD is by grouping them into three categories based on the method of application of AD: Neighbour-Based, Subspace-Based, and Ensemble-Based. These categories consist of a mix of both ML as well as statistical approaches and are more suitable for data that is high dimensional in nature where standard AD techniques may be less effective. Table 2.1 below shows a comparison.

***Table 2.1**: A Comparison of AD Types (Xu et al., 2019)*

AD Type	Description	Advantages	Disadvantages
Neighbour-Based	Anomalies are identified using neighbourhood information	1. Interpretation is easy. 2. Independent of distribution of data	1. Performance is relatively poor. 2. Sensitive to input parameters
Subspace-Based	Anomalies are detected by looking at different subsets of features	1. Highly effective in specific cases 2. Highly efficient	1. Finding relevant subspaces is not an easy task
Ensemble-Based	Several AD techniques are employed to arrive at a consensus.	1. Highly accurate 2. Less sensitive by design as it is resistant to overfitting	1. Not very efficient

2.5.1 Neighbour-Based Anomaly Detection

The Neighbour-Based AD techniques use what can be described as 'neighbourhood information' of a datapoint to check whether it's far or close in terms of proximity to other data objects. Several methods that fall under this umbrella are dependent on distance-based measures such as weighted distance or average distance between a data object and its nearest neighbours. These techniques are however not effective in the case of data that is high-dimensional in nature. Rank-Based Detection Algorithm (RBDA) is one statistical AD solution proposed for such a scenario. RBDA ranks each data object to use it as a measure of proximity to its neighbours. The ranking technique focuses on the question of whether the datapoint under consideration is 'central' with respect to its nearest neighbours. A relatively low rank would imply that the datapoint is among its neighbours while a higher rank would indicate that it is placed towards the periphery (Huang et al., 2013). RBDA however does not include distance-based information to improve its findings which could be helpful in certain cases.

2.5.2 Subspace-Based Anomaly Detection

In high-dimensional datasets, data becomes sparse and it becomes difficult to identify the real outliers. These outliers that are often low-dimensional or are abnormalities that are local in nature end up being masked when the outlier analysis is carried out in full dimensionality. This happens because of noise effects from irrelevant dimensions. Therefore, identification of anomalies from subspaces is an approach that is now considered in such cases to be effective. Most approaches can be categorized into sparse subspace and relevant subspace techniques. The former projects high dimension datapoints onto a lower dimension and subspaces that are sparse. The latter set of techniques uses local information that is represented as relevant features to detect outliers (Aggarwal, 2017).

2.5.3 Ensemble-Based Anomaly Detection

Ensemble-Based techniques are popular in ML and are also regularly employed in the field of AD. These methods make use of results from dissimilar models and combine them to produce models that are robust. This approach reduces dependency on any one model as well as the risk of skewing results due to any data locality. As a result, Ensemble-Based AD has become a popular space of research since no single algorithm can detect different anomaly types. This method is useful in situations where the nature of the outlier is in question such as whether it is cluster-based, distance-based or any other type of model-based (Wang et al., 2019).

Extreme Gradient Boosting Outlier Detection (XGBOD) is one recently proposed ensemble technique that makes use of a hybrid approach wherein both supervised and unsupervised ML techniques are employed (Zhao and Hryniewicki, 2018). The combination of both types of techniques allows their strengths to be combined. XGBOD is comprised of three phases. In the first phase, the original dataset is augmented by using different unsupervised AD techniques on itself. The augmentation is referred to as Transformed outlier scores (TOS) which are considered as a finer representation of the data. The AD techniques used here can be any and even identical but a heterogenous set is encouraged as the ensemble is likely to improve its ability to generalize if the outputs are diverse. The second phase is a selection process to keep only the useful outlier scores generated from the first phase. This can be compared to a feature selection process where the dataset is pruned to control for complexity and improve accuracy.

The final phase employs XGBoost as a classifier on the now refined feature and predicts the outliers. The XGBOD framework was tested on seven datasets to conclude that the improvements were significant in comparison to available alternatives. Figure 2.1 explains the framework of the ensemble.

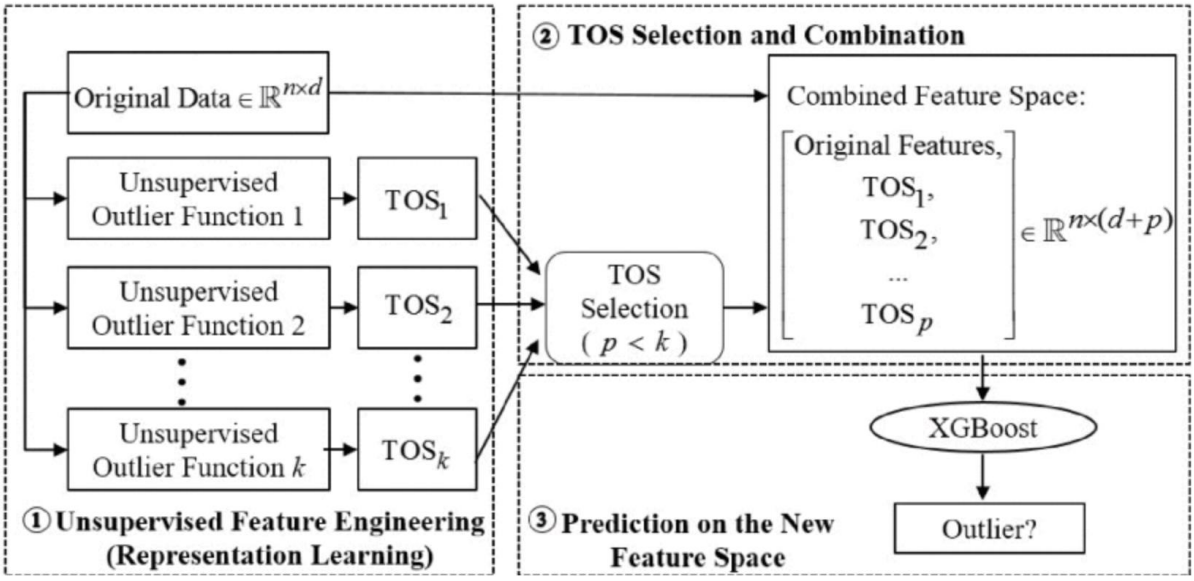

Figure 2.1: *Framework of the XGBOD ensemble (Zhao and Hryniewicki, 2018)*

Another approach is the Cumulative Agreement Rates Ensemble (CARE), an unsupervised technique that treats outliers as a binary classification problem (Rayana et al., 2016). This approach involves looking at ordinary data as a majority class and outliers as a minority with a focus on reducing both bias and variance for AD. CARE is a sequential ensemble which means that it consists of base learners that are developed over several iterations, their learning is sequential and there is dependency within them. Each iteration consists of two phases with the aim of reducing variance. The first phase combines the outputs of the base learners with the help of an unsupervised technique that estimates weights for the learners. The second phase aggregates the results of the current iteration with that of the previous ones in a cumulative manner. Also, the aggregated result from previous iterations is used to remove obvious outliers before feeding the updated data into the latest iteration. This helps bring down the bias as well.

Recent research on Ensemble-Based AD looked at emulating the success of the boosting technique in supervised training settings in unsupervised AD scenarios. This could be done by using a novel ensemble selecting method called BoostSelect. The proposed method is capable of focusing on both the accuracy and diversity of the members selected for the ensemble, unlike existing techniques that focus only on either of them while performing quite well on benchmark datasets (Campos et al., 2018).

Isolation Forest (IF) is another Ensemble-Based AD technique that is quite popular these days. Instead of profiling regular instances as many other AD models do IF is designed and optimized to specifically isolate anomalous instances. This helps avoid false positives where regular instances are identified as anomalies or fewer anomalies are caught. IF takes advantage of the fact that anomalies are a minority and have properties that are quite different from regular instances. This is done by making use of tree structures called iTrees or Isolation Trees as part of the ensemble (Liu et al., 2008). Figure 2.2 shows the basic workings of an Isolation Tree.

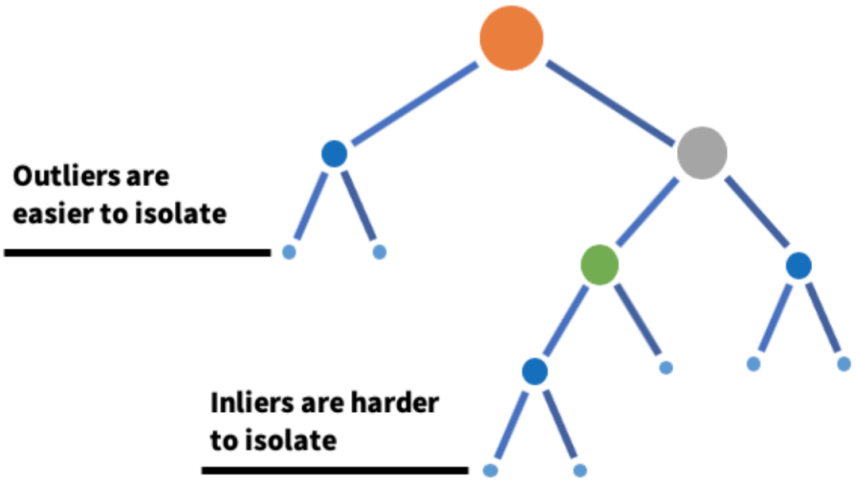

Figure 2.2: *A basic representation of the Isolation Tree* (Verbus, 2019)

The characteristics of the tree building result in the regular instances isolating at a deeper level while the anomalies isolate closer to the root of the tree. IF requires only two variables, the sampling size and the number of iTrees. IF sorts and ranks the data points based on their path lengths to identify anomalies.

The path length is a measure that indicates the number of edges travelled by data points from the root node to the external node and anomalies are those with very short lengths. The measure is called anomaly scores and falls within a range of 0 to 1 and indicates the following:

- A data instance with a score very close to 1 is considered an anomaly.
- An instance that has a score below 0.5 can be regarded as a regular instance.
- If all instances within the data sample approximately have a score of 0.5, that is indicative of the sample containing no anomalies at all.

Figure 2.3 is a representation of an IF ensemble making use of anomaly scores.

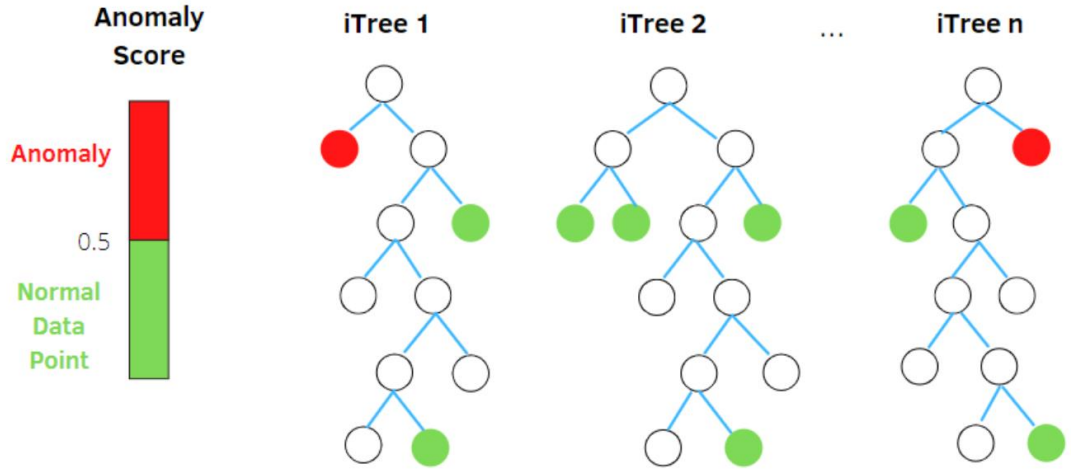

Figure 2.3: *Application of anomaly scores in Isolation Forest (Anello, 2021)*

IF is a popular AD technique as it is highly efficient, has a linear time-complexity with low resource requirements, performs much better than comparable models, and is a great choice for datasets that are high in volume as well as datasets with high dimensions. However, recent research has established that the standard form of IF while computationally efficient suffers from bias due to the manner in which tree branching is carried out (Hariri et al., 2021). Generally, in each of the iTrees the algorithm picks a sub-sample of the data from which a random value from a random attribute is selected to split datapoints and send them to the left or right branch. This is repeatedly carried out at each node until the datapoint is isolated or a fixed tree depth is reached. The splitting process described is represented as random cuts in Fidepicting the branching for a normally distributed dataset.

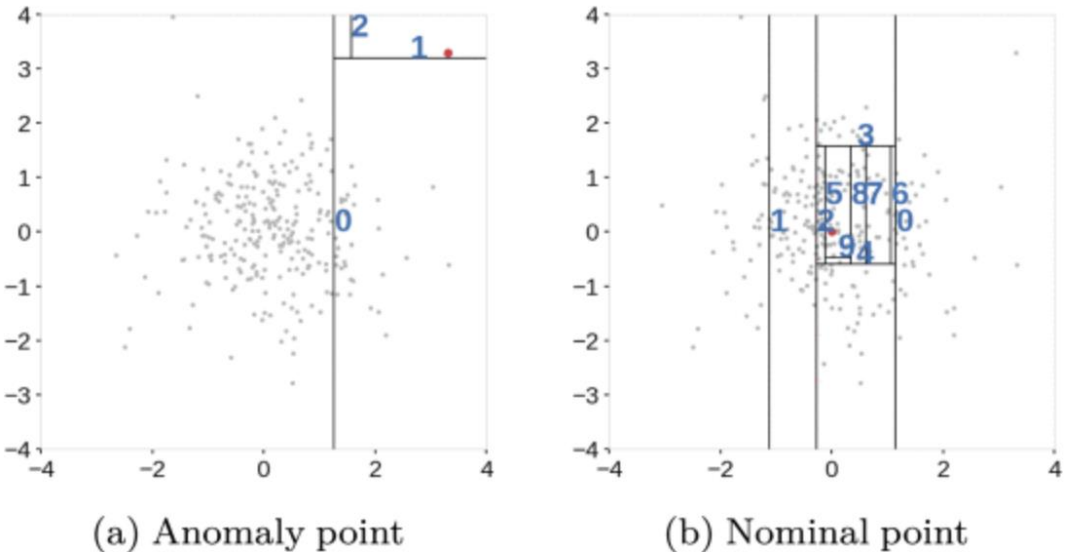

Figure 2.4: *A comparison of the branching process of the standard IF between (a) An anomaly point and (b) a nominal point. Branching is only horizontal or vertical here* (Hariri et al., 2021).

As seen in Figure 2.4, horizontal and vertical lines are the random cuts used to isolate a specific datapoint (highlighted as a red point in each image). Because the cuts are constrained to being horizontal or vertical, they tend to run through regions that are relatively sparse and don't contain many data points but end up cutting them as well. As a result, some of the datapoints that are anomalous in nature end up being subjected to more branching compared to similar anomalies in other areas and end up with different anomaly scores. The problem becomes more pronounced in cases where, for instance, the dataset has two normally distributed clusters or if the data distribution is sinusoidal in nature.

An extension to IF which is named Extended Isolation Forest (EIF) was proposed as a solution to resolve this problem. In the standard IF the cuts are horizontal or vertical but the algorithm does not have any fundamental requirement for this constraint. EIF therefore applies a random 'slope' to the branch cuts at each split. Despite this modification, the required inputs for the algorithm remain at two - specifically, a random slope for branching and a random value for the branch cut. Images in Figure 2.5 depict the sloped branches being utilized for the normally distributed dataset as before.

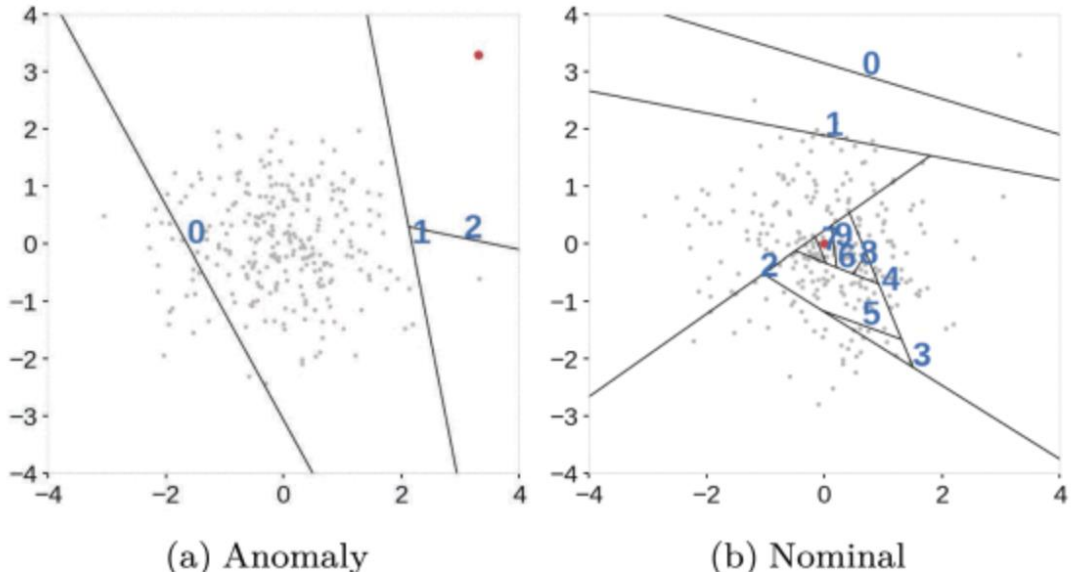

Figure 2.5: *A comparison of the branching process of the Extended IF between (a) An anomaly point and (b) a nominal point. The branching is improved with slopes* (Hariri et al., 2021).

The differences between the standard IF and EIF can also be seen in the anomaly score maps seen in Figure 2.6. Ideally, the anomaly score should be lowest at the centre and progressively increase as we move radially outward resulting in a circularly shaped score map. This however isn't the case with IF as we can see observe rectangular regions that result in lower anomaly scores for datapoints that are not anomalies. The problem appears to be fixed in the EIF as observed in the second image. Evaluations based on benchmark datasets also show that EIF consistently performs better than standard IF.

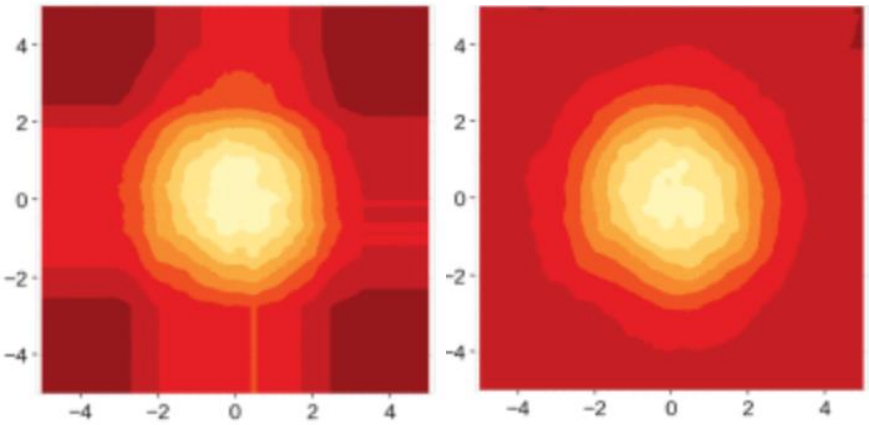

Figure 2.6: *Anomaly scores of a normalized dataset - Standard IF vs Extended IF* (Hariri et al., 2021).

2.6 Anomaly Detection in Categorical Data

What is notable about all the AD solutions discussed so far is that they are applied in cases where data is often quantitative. Categorical data has generally received relatively less consideration. This is because AD in categorical data is a challenge due to a number of reasons. Many AD techniques depend on distance functions to identify anomalous observations that deviate from a representative pattern of normal data. But neither is identifying such patterns nor measuring distance in categorical data a straightforward task. Furthermore, varying definitions of outliers in categorical data exist in the literature. Depending on the definition adopted, different AD methods may end up choosing different instances as outliers (Taha and Hadi, 2019).

Fortunately, despite these challenges, there has been relatively recent progress in research on viable options to execute AD in categorical data. Frequency-Based, Density-Based, Distance-Based, and Clustering-Based AD methods are some of them. An example would be the Ranking-based Outlier Analysis and Detection method (ROAD) which is a two-phased clustering algorithm. The first step involves computing the density of datapoints/objects as well as exploring the clustering of the dataset. The second step calculates a frequency-based rank as well as a clustering-based rank of each datapoint. The two separate rankings are then used to then determine the outliers (Suri et al., 2013).

Isolation Forest (IF) and HBOS, both of which we have already discussed in earlier parts of our review can also be employed for categorical data. In the enterprise case study we previously discussed, apart from using IF with continuous data, it was also employed for use with categorical data by mapping the data with numeric values. This resulted in the algorithm treating the categorical dimensions in the same way as it did numeric dimensions (Sun et al., 2016). The HBOS technique takes the count of values of each category of each feature which aids in the calculation of the frequency or the histogram's height (Goldstein and Dengel, 2012).

Another example is an ensemble learning method that was proposed for datasets that are categorical in nature. The ensemble consisted of several AD techniques which include KNN, SVM, IF, Local Outlier Factor (LOF), and Elliptic Envelope (EE).

The ensemble also employed two encoding techniques, One Hot Encoder and Binary Encoder, which are used to convert the categorical data into numerical. The AD performance of the ensemble was tested on three different datasets and found to be consistently better across all metrics when compared with that of the individual models themselves (Thomas, 2020). Figure 2.7 provides a visual representation of the ensemble's architecture.

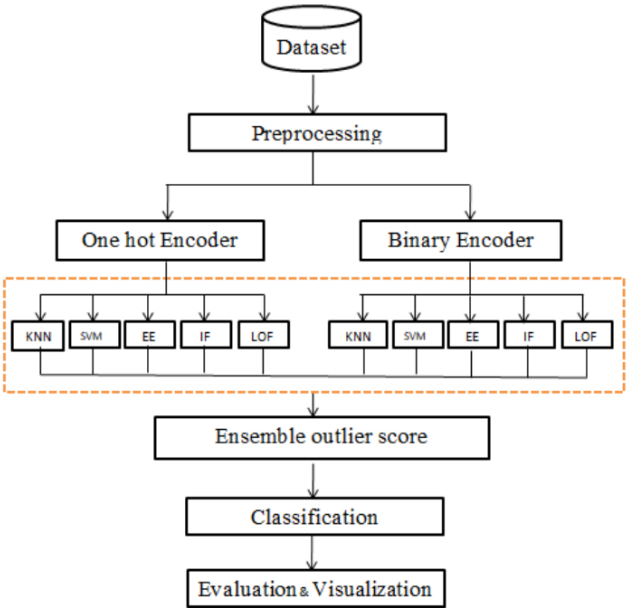

Figure 2.7: *Architecture of the Ensemble-Based AD used with categorical data (Thomas, 2020)*

2.7 Role of Anomaly Detection in Covid-19 detection

Even though Covid-19 is relatively very recent in terms of research interest, AD has already found its way to being employed in proposed ML/DL Covid-19 detection solutions, albeit in a seemingly limited amount. A DL model consisting of three components, specifically a backbone component based on an ImageNet dataset trained CNN, a classification component, and an AD component was suggested as a solution to detect Covid-19 using X-ray images. Due to the limited availability of Covid-19 images and the requirement for more data in DL, a mix of pneumonia confirmed images are also included.

The backbone extracts the necessary features, provided as input in classification and AD components, both of which then generate classification and anomaly scores. Statistically significant scores are assigned to Covid-19 X-ray images to optimize the model to identify Covid-19. This solution too was suggested as an alternative in circumstances where the availability of RT-PCR tests is limited (Zhang et al., 2020).

A wearable device was proposed as an option to detect Covid-19 early on by measuring physiological features such as body temperature, physical activity, breathing, and cough patterns. Both K-Means and IF techniques were utilized as AD-based symptom identification. The outliers would point to patients with deviating patterns that require a closer look (Ali et al., 2021).

Most of the ML/DL methods used in the detection of Covid-19 discussed so far did not consider the impact of AD. Those that did apply AD on quantitative data that was usually image-based data like X-ray, CT scans, and MRI. To the best of our knowledge, the impact of AD on ML's predictive power when used on symptomatic structured data that is predominantly categorical in nature, specifically a Covid-19 dataset, is yet to be explored.

2.8 Summary

In our literature review, we looked at research on ML-based Covid-19 research that is capable of assisting in the detection and mitigation of the disease. We also looked at how medical imaging (X-ray, MRI, CT scans) is being used in Covid-19 detection because of its prevalence in healthcare and how DL techniques such as CNN and LSTM are deployed to take advantage of the imaging data and provide highly accurate predictions. We then took a close look at Anomaly Detection (AD) and the different categories of ML-based techniques tailored for different data and labelled/unlabelled anomaly scenarios. We also looked at other AD techniques which are better suited for data with high dimensions. Ensemble-Based AD was particularly found to be versatile in its application across different kinds of datasets. We then reviewed AD techniques specifically employed for categorical data since it was established that AD in general was focused on quantitative data. Finally, our review looked at the scenarios in which AD is already being used to assist in the detection of Covid-19.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

In the methodology section, we look at some of the practical aspects of our study and how we plan on getting valid and reliable results that address the research aims and objectives. We also look at how the data was collected and the different aspects of the exploratory analysis of our data. We pay attention to the AD techniques that we plan to utilize and the specifics of how they will be implemented and evaluated. We also describe the model-building process, list the ML predictive models chosen as part of our proposed AD framework, and justify why they were picked. And finally, we look at how we will evaluate the performance of our framework.

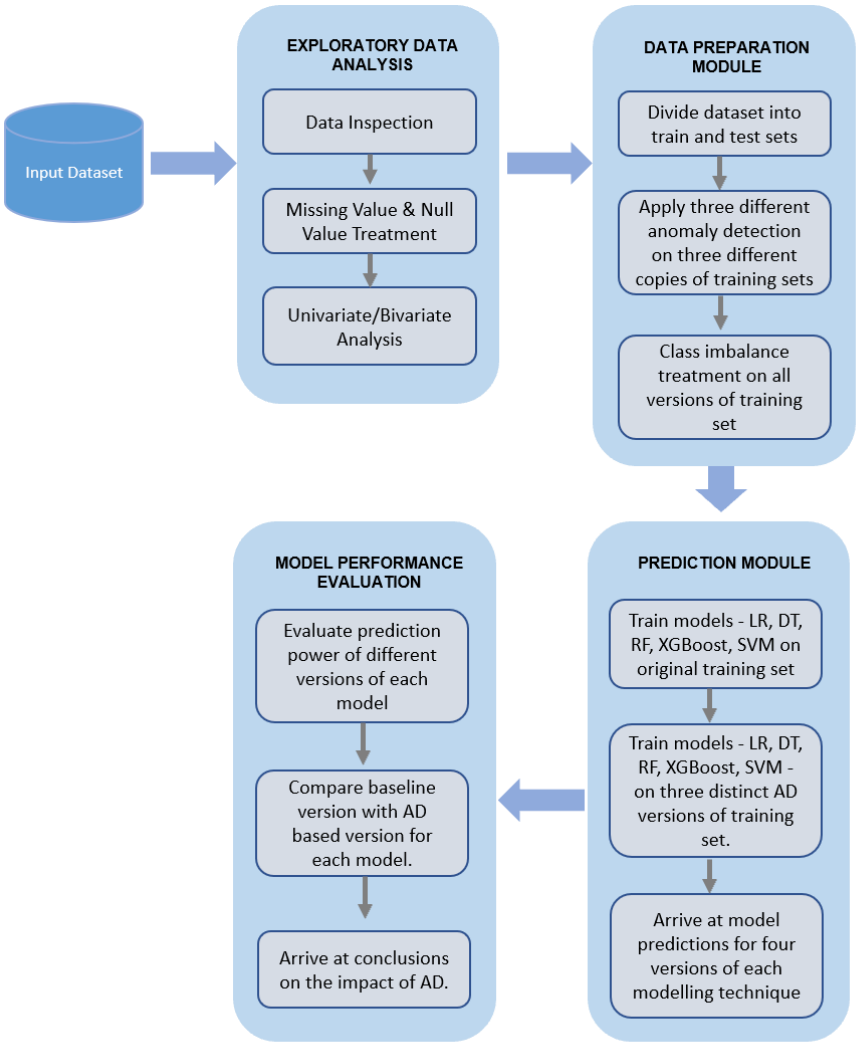

Figure 3.1: *Framework of Anomaly detection-based model development*

3.2 Research Approach

3.2.1 Data Description

The dataset has been made available by Big Data High-accuracy Center, Beijing University on their Github page as well as on Kaggle (Ahamad, 2021). The anonymized dataset consists of 6,512 patients from seven different provinces of China and was collected by the university from official channels of the Chinese national government's websites. Attributes include basic information (gender, age), symptoms (muscle soreness, cough, fever, runny nose), symptoms that resulted in hospital admission (diarrhoea, pneumonia), results from diagnostics (radiographic imaging, lung infection), and other information considered relevant (isolation treatment status, travel history). A target variable is also present which confirms if the patient has Covid-19 (1,572 cases) or not (4,970 cases). The dataset is categorical in nature, with the exception of the age attribute. Table 3.1 below provides further necessary details on the dataset.

Table 3.1: *Data Description*

Attribute	Data Type	Description
Fever	Boolean	Symptomatic with body temperature is greater than 38 degrees.
Pneumonia	Boolean	Symptomatic for Pneumonia resulting in hospital admission
Runny Nose	Boolean	Symptomatic for Runny Nose
Lung Infection	Boolean	CT Scan or Radiography images indicate lung infection
Cough	Boolean	Symptomatic for dry cough or cough with sputum
Diarrhoea	Boolean	Symptomatic for Diarrhoea resulting in hospital admission
Muscle Soreness	Boolean	Symptomatic for limb pain or soreness of muscle
Isolation	Boolean	Status of isolation treatment
Travel History	Boolean	Travelled to one or more places within China or abroad
Age	Integer	Age within range of 1-96
Gender	String	Indicates whether patient is Male or Female.
SARS-CoV-2 Positive	Boolean	Confirms whether the patient is Covid-19 positive or not

3.2.2 Exploratory Data Analysis

Datasets generally consist of instances that can be attributed to noise as well as a high number of features, missing values, null values, and inconsistencies likely due to the manner in which the data was sourced. Exploratory Data Analysis (EDA) would be required to understand the intricacies of the data and will, in general, consist of the following steps:

1. Data inspection to get an overview of the dataset.
2. Missing Value Treatment
3. Null Value Treatment
4. Univariate and/or Bivariate analysis of the available features.
5. Where necessary, derive variables and/or modify existing ones for better interpretation.
6. Use of data visualization packages to assist in the above steps.

3.2.3 Data Preparation Module

This module is focused on preparing the data prior to initiating the steps required for model building. This includes splitting the data into training and testing sets, applying the AD techniques to the training set, and class imbalance treatment for the training set.

3.2.3.1 Splitting the dataset

Prior to model building, we will split the data into training and testing sets. The former will be used for model building and will also be subject to further modifications as described in the subsequent subsections. The latter will be masked from the model so that it can be used to play the role of a real-world dataset. The testing set will only be used for the evaluation of the performance of our models.

3.2.3.2 Anomaly Detection

In this step, three different AD techniques will be employed on three different copies of the training set. Each of the AD techniques will be used to label and remove outliers in their respective copies. This is done in order to separately evaluate each of the AD techniques' impacts on the overall model prediction. The specifics of how these copies will be used in model building are explained in the Prediction Module. The three AD techniques are detailed below:

1. Isolation Forest (IF): Based on our work in the literature review, we found that it is a suitable technique to handle binary/categorical data. This is doable if categorical data is represented in numeric values which is already the case in our dataset (Liu et al., 2008). IF returns anomaly labels for all trainset datapoints where '1' indicates they are inliers and '-1' means they are outliers. The labels will be used to remove outliers.

2. Extended Isolation Forest (EIF): A special case of IF that has improved on some of the shortcomings in the detection of anomalies and is also capable of dealing with binary/categorical data (Hariri et al., 2021). EIF does not return anomaly labels but returns anomaly scores. A score close to 1 would indicate a high chance of the datapoint being an anomaly and a score less than 0.5 would mean it to be a normal instance. Based on the anomaly scores generated, an anomaly score threshold can be arrived at and datapoints with higher than said threshold would be classified as anomalies.

3. Histogram-based Outlier Detection (HBOS): A histogram-based technique identified in our review which is capable of handling categorical data (Goldstein and Dengel, 2012). Like IF, HBOS also returns anomaly labels that categorize datapoints into inliers and outliers. The latter would then be removed from the trainset.

To deploy HBOS, we will use an open-source Python package called PyOD (Zhao et al., 2019). This package provides access to several AD techniques, including HBOS. For our work, IF will be deployed using the scikit-learn package (Pedregosa FABIANPEDREGOSA et al., 2011). EIF is a relatively new technique and therefore has limited options that have been developed for deployment. H2O is an open-source ML framework developed by H2O.ai that offers EIF (Candel et al., 2020; Extended Isolation Forest — H2O 3.34.0.4 documentation, 2021). H2O requires the latest version of Java to be installed prior to deployment as well as dependency packages to be installed, which are listed below:

- requests
- tabulate
- future

3.2.3.3 Class Imbalance Treatment

The number of Covid-19 positive cases in our dataset is 1,572 while that negative is 4,970. That is about a ratio of 1:3 with the positive cases being the minority class. Class imbalance in the target variable can be challenging as machine learning models and methods of evaluation assume that the dataset has a balanced distribution (Brownlee, 2020). For this reason, class imbalance treatment will be implemented only on the training set.

In the case of models like RF, DT, and SVM which are deployed using the sklearn package, the class imbalance will be handled during model building using the parameter 'class_weight' that will automatically adjust weights inversely proportional to class frequencies in the training data. The formula for class_weight is given below:

$$Weight\ of\ Unique\ Class = \frac{Total\ Number\ of\ Samples}{Total\ Unique\ Classes * Total\ Samples\ in\ Unique\ Class)} \quad \textbf{(3.1)}$$

In the case of a model like XGBoost, we will use a hyperparameter specific to the model called 'scale_pos_weight' which controls the weights for unbalanced classes. The hyperparameter will be tuned during the model building phase to arrive at the appropriate values for a balanced class. For models like LR, which will be deployed using the statsmodels package, Synthetic Minority Class Oversampling Technique (SMOTE) will be used prior to the model building as statsmodels does not provide options to handle imbalance. SMOTE creates synthetic datapoints to artificially balance out the minority class.

3.2.4 Prediction Module

In this module, we will apply five popular ML classification models on the Covid-19 dataset to predict Covid-19 at an early stage. The reason we choose five models is to verify the consistency of the impact of the AD module on the prediction power before and after the removal of anomalies. All of the below listed models are considered to be benchmarks in terms of performance and are capable of handling categorical data. We also stick to only shallow ML models as DL models require a lot more data for training and therefore are not viable options. Each model will be used to create four different versions based on four different copies of the training set. One version will be based on the original training set prior to any AD technique being applied and will help us understand the baseline performance of a model. The other three versions will be based on the three distinct training set copies, each modified by a distinct AD technique. Once built, their respective performances will be evaluated and compared using the testing set. The models that we plan to use are listed and described below.

3.2.4.1 Logistic Regression (LR)

Logistic Regression (LR) is an ML model based on a technique from the field of statistics. It is a popular method for classification problems that are binary in nature. In this method, each instance is modelled for its probability of falling into a certain class. A statistics-based function, namely the Sigmoid function, is used to arrive at the probability, ranging from 0 to 1 (Cramer, 2003). We have picked LR as one of our models since it is well known to be capable of handling binary/categorical independent variables (Fleiss et al., 1986; WALDMAN et al., 1999).

3.2.4.2 Decision Tree (DT)

Decision Tree (DT) models are utilized for both regression and classification. The model is based on a tree representation where the leaf node is associated with features and branches are associated with values. DT is based on the idea that a tree can be built using the data available and at every leaf, a unique output is represented. A statistical concept called information gain (IG) is used to measure how well a feature can assist in classifying the data. A feature present at a node with high IG can effectively split the dataset thereby assisting in improving the accuracy of classification. DT is also capable of handling binary/categorical independent variables (Ahamad et al., 2020).

3.2.4.3 Random Forest (RF)

The Random Forest (RF) (Breiman, 2001) is a classification technique based on the DT model. It is an ensemble of decision trees, each of which is trained on a distinct subset of the original dataset. Each of these trees makes its prediction on the classification of samples in the test set. The final classification is decided based on prediction from multiple sets of trees through a mechanism that is based on majority voting called Bagging. This results in the overall model being more reliable as the predictions made by an ensemble of trees will likely be more consistent than that made by individual DTs. Since DT is capable of handling categorical data, RF being an ensemble of DTs is also, by extension, capable of managing it (Ahamad et al., 2020)

3.2.4.4 Extreme Gradient Boosting or XGBoost

XGBoost is another DT ensemble-based technique, like RF (Chen and Guestrin, 2016). The difference is that it is based on a mechanism called boosting where DT models are built sequentially with each model learning from the mistakes of the previous one.

Models which perform well have more influence on the final outcome or are 'boosted'. Additionally, the technique also uses Gradient Descent to reduce errors in the sequentially created models. XGBoost, similar to the other ensemble-based technique RF, is able to deal with binary/categorical data (Ahamad et al., 2020).

3.2.4.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a very popular ML model, primarily due to the fact that it is a high-performance technique with relatively little tuning required (Cortes et al., 1995). SVM classifies data by plotting each instance in k-dimensions where k is the total attributes in our dataset. Each datapoint is plotted as a coordinate based on the values of its corresponding attributes. The SVM then identifies the hyperplane that can separate the two classes, thus enabling the classification of the data points. SVM too is able to manage independent binary/categorical variables for its predictions (Ahamad et al., 2020).

3.2.5 Model Performance Evaluation

To quantify the performance of our models in the prediction of Covid-19, we will use well-known metrics in the ML field. The performance evaluation will be based on the model's predictions on the test dataset.

3.2.5.1 Precision

Precision helps us understand how many of the positive classes predicted by the model are positive. This metric is helpful when the cost of a false positive is high. The formula is given below:

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad \textbf{(3.2)}$$

3.2.5.2 Recall

Recall is important when the cost of a false negative is high. It tells us how many of the actual positives the model was able to predict correctly. In domains such as healthcare, this is a crucial metric since the cost of not taking action due to a false negative could be high. The formula is given on the next page.

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \quad \textbf{(3.3)}$$

3.2.5.3 Accuracy

This metric informs us of the model's general performance and if it is being trained correctly overall. Accuracy tells us the overall ratio of the correctly predicted to the total observations. However, it will not be able to get into any specifics of the model's performance like how Precision & Recall does. The formula is as follows:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad \textbf{(3.4)}$$

3.2.5.4 F1 Score

F1 score is a balance between Precision and Recall. It takes the weighted average between the two metrics. The score ranges between 0 and 1 with higher being better. While not as intuitive as accuracy, it is more useful in cases where the class distribution is uneven. It is a more useful metric than accuracy when the cost of fall positives and false negatives are uneven. The formula is described below:

$$F1\ Score = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \quad \textbf{(3.5)}$$

3.3 Summary

Our methodology section presents details on how we propose to evaluate the impact of AD on symptomatic data for Covid-19. At a high level, we start with EDA to understand the nature and quirks of the available data through univariate/bivariate analysis and to handle any null and missing values. This is followed by data preparation where the focus is on steps prior to initiating model building. This includes splitting the dataset into train and test sets, applying three different AD techniques on three distinct copies of the training set, and class imbalance treatment on all copies of the training sets. We then look at the specifics of the model building where we list the five models that we will be training and evaluating.

We also look at how each model will have four versions, one which will be trained using the original training set and serves as a baseline while the remaining three will be built on the AD modified training sets. We finally describe the evaluation metrics that we will use to compare the performance of the different versions of the classification models.

CHAPTER 4

MODEL DEVELOPMENT

4.1 Introduction

This chapter looks at the different stages of developing the proposed framework and the following sections and subsections get into the details of the methodology undertaken as part of this study. The chapter begins with the Exploratory Data Analysis, which consists of several important steps, including data cleaning and univariate/bivariate analyses. In the former, we make modifications to the data for ease of analysis and in the latter, we look at the attributes of our dataset both individually and in a combined manner in order to reveal useful patterns. This is followed by the Data Preparation stage where preparation of the data prior to model building is the focus. The specifics of the data split into training and testing sets are explained in this section. We also study the three AD techniques we are employing, the hyperparameters that were chosen, their respective approaches to detecting anomalies, and their impacts on the training set. The implementation of class imbalance treatment specific to each model is also explained here. Finally, the chapter looks at the Model Building phase where the specifics of building each modelling technique, both the baseline and AD-based versions of the model, are looked at.

4.2 Exploratory Data Analysis

The Covid-19 dataset that we used for our study is in a structured format and its attributes and their respective properties have already been detailed in the previous chapter and Table 3.1. The dataset, however, required going through several steps prior to being employed in the model building that is part of our AD framework. The following sub-sections will detail steps that were taken as part of EDA including preprocessing steps and univariate/bivariate analysis among others.

4.2.1 Missing Value Treatment

As part of our data inspection, we confirmed that the dataset had no missing values in any of its attributes and therefore did not require any treatment for any missing values.

4.2.2 Null Value Treatment

Our inspection of the data also informed us that none of the attributes have any null values. Null value treatment was therefore not necessitated.

4.2.3 Data Cleaning

The following changes were made to the dataset for ease of analysis:

- The ‘gender’ attribute was converted from a categorical datatype to a numeric type that is binary in nature where Male = 1 and Female = 0.
- A new variable ‘age_group’ was derived from ‘age_year’ by binning the patients into different groups for use later in univariate and bivariate analyses.

4.2.4 Univariate Analysis

To understand the patterns of each of our dataset attributes, univariate analysis was carried out on each of the variables in our data. The following subsections detail the steps that were taken.

4.2.4.1 SARS-Cov-2 Positive (Infected)

We started with our important attribute, SARS-Cov-2 Positive, which is considered the target variable for our model building. Understanding its distribution helped decide further steps prior to model building. For ease of use, the attribute was renamed to ‘Infected’ and has been referenced with this name in any subsequent visualizations.

Table 4.1 *Class distribution of target variable*

SARS – Cov2 - Positive		
Value	**Total Instances**	**Percentage**
1	1572	24%
0	4940	76%

Table 4.1 shows that the class distribution of our target variable was initially quite skewed towards non-infected patients (76%). This was corrected later using data imbalance treatment since an imbalance of the dependent variable can skew the results of our models as well.

4.2.4.2 age_year

Figure 4.1 shows a density plot of patient age in our dataset. We observed that age follows a pattern that is similar to a normal distribution with middle-aged patients having a higher proportion in our dataset.

Figure 4.1 *Density plot showing distribution of patient age*

Table 4.2 shows the distribution of patient age when they are binned into different age groups. From this analysis, it was clear that patients aged between 40-60 and 20-40 are the highest and second-highest respectively, and cumulatively add up to nearly 80% of the dataset. Based on this, a bivariate analysis of age groups and our target variable, discussed later, looked into whether the higher share of middle age groups has any relation with the likelihood of Covid-19 infection.

Table 4. 2 *Distribution of patient count basis age group*

Age Group		
Age Group	**Total**	**Percentage**
0 – 10	123	1.9%
10 – 20	248	3.8%
20 – 40	2458	37.7%
40 – 60	2666	40.9%
60 – 80	912	14%
80 – 100	105	1.6%

4.2.4.3 Independent Binary Features

We looked at the remaining independent features which include gender, various patient symptoms, travel history, and isolation treatment.

Table 4.3 *Distribution and respective percentages of independent binary variables*

Binary Feature	Value = 1	Value = 1 (% of total)	Value = 0	Value = 0 (% of total)
gender	3367	52%	3145	48%
fever	2675	41%	3837	59%
cough	1975	30%	4537	70%
runny_nose	549	8%	5963	92%
muscle_soreness	26	0.1%	6486	99.9%
pneumonia	487	7%	6025	93%
diarrhea	37	1%	6475	99%
lung_infection	855	13%	5657	87%
travel_history	4239	65%	2273	35%
isolation_treatment	1413	22%	5099	78%

Table 4.3 shows the total count of instances and the percentage of the total for each of these features within our dataset. From this analysis, we were able to gather several aspects of our patients' profiles. We saw that the ratio of male to female patients in our dataset is almost evenly balanced with the percentage of male patients being slightly higher.

We also observed that patients with fever or cough make a large number with 41% and 30% respectively of the total displaying these symptoms. However, data available on all other symptoms were highly skewed with a significant number of patients not exhibiting said symptoms. 13% and 10% of all patients had lung infection and runny nose respectively. The number of patients with pneumonia was even lesser at 7%. Patients with diarrhea and muscle soreness were negligible in numbers. As part of the treatment protocol for Covid-19, a large number of the patients were isolated (78%). It was also noted that 65% of the tested patients had a travel history.

4.2.5 Bivariate Analysis

This section looks at the different combinations of features that were analyzed to understand how they influenced each other and the patterns that emerged from the analyses.

4.2.5.1 SARS-Cov-2 Positive (Infected) vs Independent Binary Features

We looked at all independent binary attributes - gender, disease symptoms, travel history, and isolation treatment - in comparison to the target variable 'Infected'. The features mentioned were binary in nature (ones and zeros) and by calculating the mean for each of them, we were able to depict the percentage of total patients for whom the value of the variables was equal to one. To accommodate all binary features adequately, they were split into two sets of visualizations. Figures 4.2 and 4.3 are bar charts that make the comparisons.

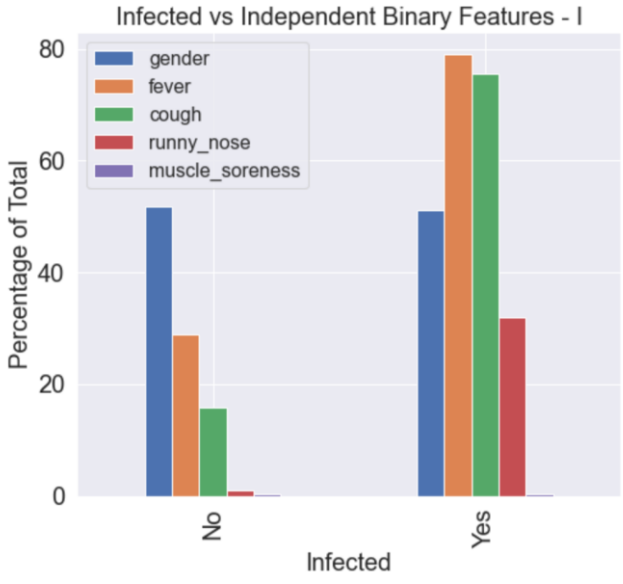

Figure 4.2 *Bivariate Analysis - Infected vs Independent Binary Variables - I*

In Figure 4.2, we can see that Covid-19 infection did not affect one gender differently than the other since 50 % of the patients, both infected and non-infected, were males. However, in the case of symptoms, a far higher percentage of patients who were infected had fever and cough (nearly 80% each) when compared to those who were not infected (about 30% and 20% respectively). Runny nose was also another symptom that was skewed towards patients who were infected with about 30% of the total having had it as a symptom while it was negligible in those who were not infected. Muscle soreness was practically nonexistent in both types of patients which was not surprising given our observations from the univariate analysis.

Figure 4.3 continues our analyses of other patient symptoms. Pneumonia seemed to be a clear differentiator with 30% of infected patients contracting it while none of the non-infected had the symptom. Lung infection too was significantly higher for Covid-19 positive patients (50%) compared to those not infected (about 2%). Diarrhea, similar to our findings from univariate analysis, was close to zero in both types of patients. We also saw that travel history was not indicative of an increased risk of infection. In fact, over 70% of patients who were not infected had a travel history while only about 35% of those infected had any history of travel. Isolation treatment was also carried out more consistently with infected patients (nearly 40%) than non-infected (over 15%).

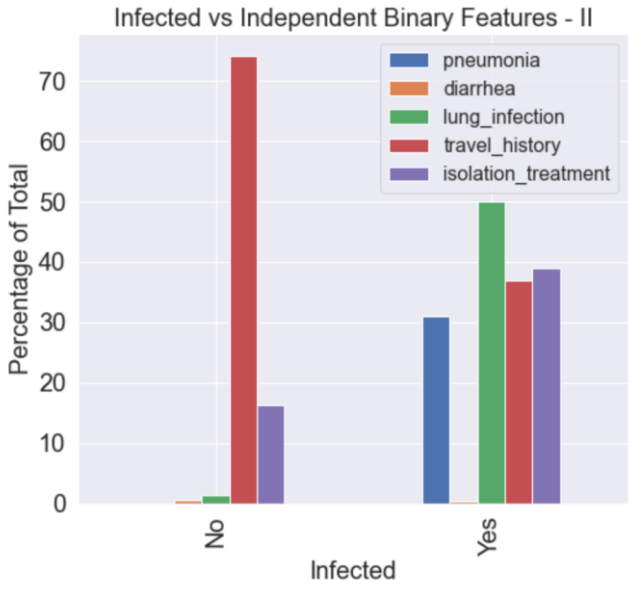

Figure 4.3 *Bivariate Analysis - Infected vs Independent Binary Variables - II*

4.2.5.2 age_year vs Binary Features

In this analysis, we looked at how patient age relates to Covid-19 infection, patient's gender, disease symptoms, travel history, and isolation treatment. Once again, for ease of interpretation, we split the binary features between Figure 4.4 and Figure 4.5. Like the previous bivariate analysis, the mean for each of the binary features represented the percentage of total patients for whom the value of the variables was equal to one.

As seen in Figure 4.4, our data was more skewed towards males for the age groups 0-10 and 10-20 (65% -70%) with gender more balanced out for the remaining age groups. In the case of fever, we saw that, with the exception of age group 10-20, all age groups had between 40%-50% of patients reporting the symptom. However, the trend was generally on the rise indicating increased age was likely to increase chances of fever. This was even more evident with cough where the symptom generally affected a higher percentage of patients as the age bracket increased. Muscle soreness was once negligible or zero across all age groups. What was interesting to see was that Covid-19 affected a higher percentage of patients of the lower range of age before hitting a trough at the age group 20-40 and then rising and peaking with older patients of age 80-100. One possible interpretation was that the young and old aged patients of our dataset were more susceptible to Covid-19 infection. However, from our univariate analysis, we knew that the total number of patients in those age groups only made up about three percent of our dataset.

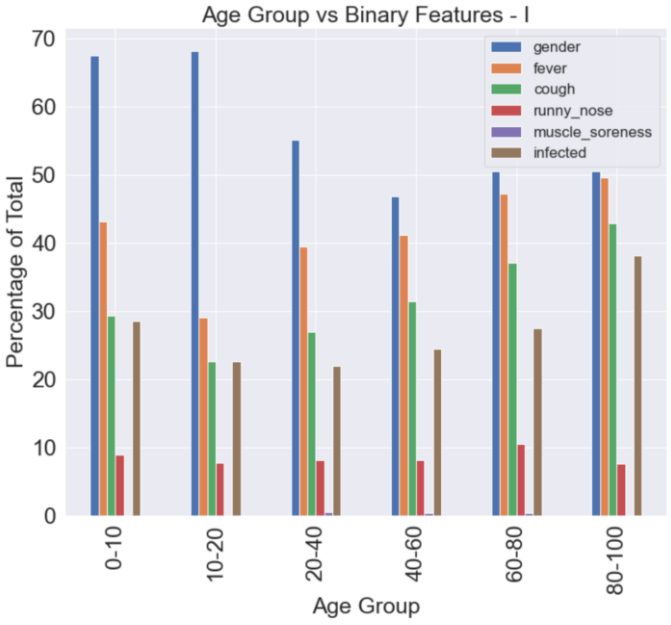

Figure 4.4 *Bivariate Analysis - Age Group vs Binary Features - I*

In Figure 4.5, we saw that pneumonia affected all age groups similarly with a similar percentage of patients showing symptoms. Diarrhea symptoms had a minute share across all age groups. Lung infection was of a relatively higher percentage for age groups 80-100 and 0-10 indicating possible vulnerability for these patients. Isolation treatment was administered to a similar percentage of patients irrespective of age group. Travel history was a high percentage for all age groups with patients aged 20-40 being the highest.

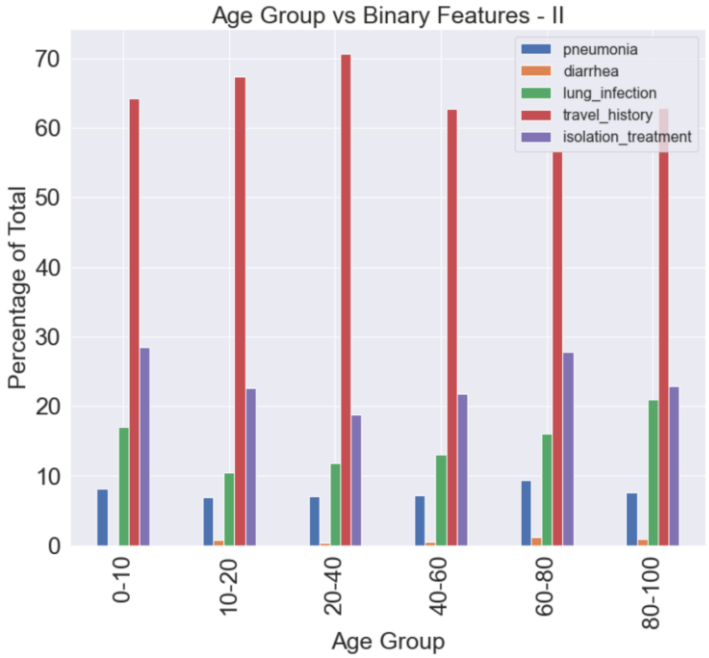

Figure 4.5 *Bivariate Analysis - Age Group vs Binary Features - II*

4.3 Data Preparation

This section gets into the details of the steps taken prior to building our models. The following subsections look at how the data was split for use in model building, treatment of data imbalance, and detail the anomaly detection techniques that were employed.

4.3.1 Splitting the dataset

The dataset was split into training and testing tests using a 70:30 ratio using sklearn's model_selection package. The following parameters were employed here:

1. train_size = 0.7
2. test_size = 0.3

The split ratio used here was based on standard practices that allow a sizeable training set to be employed in training our predictive models and a sufficiently sized testing set to evaluate the model performance. The following were the number of samples in our dataset prior to and post splitting:

1. Complete dataset = 6,512 samples
2. Training set = 4,558 samples
3. Testing set = 1,954 samples

Post splitting the dataset, normally a feature scaling technique is employed on the independent numeric variables to bring them all on the same scale, making them useful for meaningful comparisons which also helps model building and interpretability. However, this is required only when several numeric variables are present in the dataset and ours only had one (age_year) while all others were binary/categorical variables. We, therefore, did not employ this technique.

4.3.2 Anomaly Detection

This section looks at how AD was used to detect anomalies in our data. Our proposed framework employed AD only on the training set since the goal is to evaluate its impact on model prediction. The testing set was hence not exposed to our chosen AD techniques. The following subsections detail each of the AD techniques that we used – IF, EIF and HBOS. As detailed in Chapter 3, in order to measure the impacts of AD, copies of the trainset were made for each of the AD techniques, all of which were later used separately for training the ML models.

4.3.2.1 Isolation Forest (IF)

We used the sklearn package to deploy IF. Table 4.4 lists the hyperparameters of IF and the corresponding values that were used.

Table 4.4 *Isolation Forest - Hyperparameters used and corresponding values*

Isolation Forest	
Hyperparameters	**Values**
n_estimators	100
max_samples	'auto'
max_features	1.0
contamination	0.05
bootstrap	False

We used an ensemble of 100 tree-based estimators, which is also the default value prescribed for sklearn based IF. 'max_samples' was set to 'auto' which defaults to 256 samples of the total being used. This was decided as IF performs better when a smaller sample, rather than the full dataset, is employed for training for identification of anomalies. 'max_features' decides the number of attributes of the dataset used for training IF, and here we used all of the available ones. 'contamination' is used to define the proportion of anomalies in our dataset. Since our dataset is unlabeled, we went with a conservative estimate of 0.05 or five percent. 'bootstrap' was set to False so that IF samples the training data with replacement, which allows for better isolation of anomalies.

Using the above hyperparameters, we used two of the IF methods to return anomaly scores and anomaly labels for each of the datapoints in our trainset. Figure 4.6 depicts the spread of the said anomaly scores of our trainset using a boxplot.

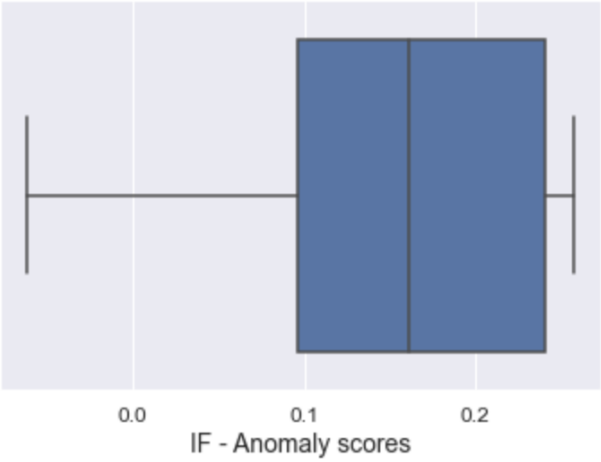

Figure 4.6 *Isolation Forest – Boxplots depicting anomaly scores of training set*

Datapoints with lower scores were more likely to be anomalies. Datapoints with negative scores were confirmed anomalies and can be seen in Figure 4.6 as the instances closer to the lower fence of the boxplot are less than zero. The anomaly labels returned by IF categorized the trainset instances into outliers or inliers based on the scores. A copy of the trainset was created with the outliers removed, which was five percent of the trainset, and 4330 instances remained.

4.3.2.2 Extended Isolation Forest (IF)

The H2O package was used to deploy EIF in our experiments, which was done by importing the package followed by the initialization of a local H2O instance. During the analysis, the following limitations were discovered, and workarounds were administered:

1. H2O only allowed for importing of csv files but not a Pandas DataFrame. Since the trainset was already preprocessed and stored as a DataFrame, we temporarily saved it as a local csv file followed by importing the trainset.
2. The results returned by EIF was an H2OFrame, a data store similar to pandas DataFrame but not directly usable in our pandas-based analysis. The H2OFrame was converted into a separate pandas DataFrame and later added to our trainset using pandas merge.
3. Unlike IF, H2O-based EIF only returned anomaly scores but not anomaly labels, which left it to the user to define the threshold for classifying a datapoint as an anomaly. The threshold for our trainset was determined based on our analysis that follows shortly.

The hyperparameters used for EIF and their corresponding values are listed in Table 4.5.

Table 4.5 *Extended Isolation Forest - Hyperparameters used and corresponding values*

Extended Isolation Forest	
Hyperparameters	**Values**
ntrees	100
sample_size	256
extension_level	P – 1 (P = Number of features)

Since one of the objectives of our research is to recommend an AD technique, the values of the hyperparameters used for EIF were similar to that of IF so that the end results are comparable. Once again, an ensemble of 100 tree-based estimators was employed to build EIF. A smaller sample of size equal to 256 was used for improved anomaly detection as opposed to using the whole trainset. 'extension_level' was dependent on the number of features of the trainset planned for training EIF. The maximum value is P−1, which stands for a full extension. As the 'extension_level' is increased, the bias of EIF is reduced. Additionally, since we used all the features of the trainset in IF, we did the same for EIF as well and assigned the maximum value for 'extension_level'.

As mentioned earlier, H2O-based EIF returned anomaly scores but not anomaly labels. We, therefore, defined our own anomaly threshold by studying the anomaly scores. Figure 4.7 shows a boxplot that visualizes the spread of the anomaly scores of our training set.

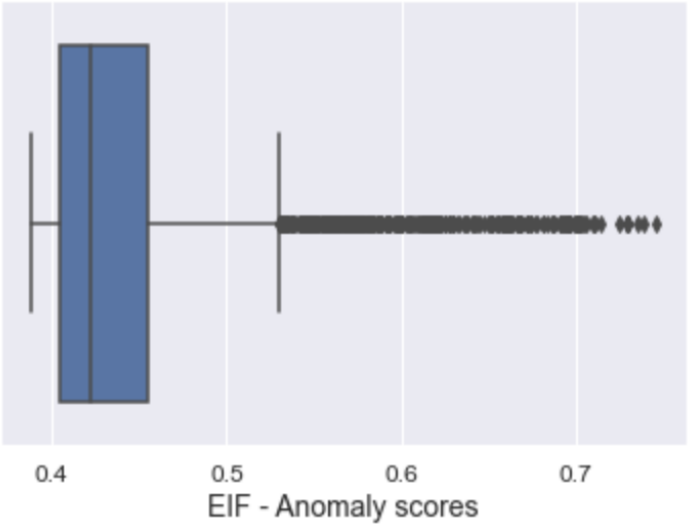

Figure 4.7 *Extended Isolation Forest – Boxplots depicting anomaly scores of training set*

As discussed in the literature review on EIF in Chapter 3, the anomaly scores were interpreted as follows:

- Instances with scores close to 1 were definitely anomalies.
- Instances with scores lesser than 0.5 could safely be considered normal instances.

We found that over 14% of our trainset instances had anomaly scores greater than 0.5 in the trainset. Excluding that many data points could potentially compromise model-building effectiveness. We instead took a more conservative approach where only five percent of the data points with the highest anomaly scores) would be considered as anomalies. This was a safer approach since the boxplot itself depicts the datapoints with these scores as outliers in the spectrum. It is also consistent with the contamination_factor = 0.05 defined in IF except that with EIF we had to manually define and exclude. We arrived at an anomaly threshold equal to 0.591. Another copy of the trainset was created with all datapoints with scores greater than 0.591 removed, which was about five percent of train instances. The remaining number of training instances was equal to 4,330.

4.3.2.3 Histogram-Based Outlier Detection (HBOS)

HBOS was deployed using the PyOD package and had the hyperparameters and corresponding values shown in Table 4.6.

Table 4.6 *HBOS - Hyperparameters used and corresponding values*

Histogram-Based Outlier Detection	
Hyperparameters	**Values**
n_bins	'auto'
alpha	0.1
contamination	0.05

The number of bins ('n_bins') for each feature in the trainset was set to 'auto'. This allowed the optimal number of bins to be calculated in accordance with the Birge-Rozenblac method (Birgé and Rozenholc, 2006). 'alpha' is the regularization parameter for HBOS and was set to the default value of 0.1. 'contamination', like the hyperparameter in IF, defines the proportion of anomalies in our dataset. We set it to 0.05 to keep results comparable. Similar to IF, PyOD based HBOS returned both anomaly scores and anomaly labels. Figure 4.8 shows the spread of anomaly scores calculated using HBOS for our trainset.

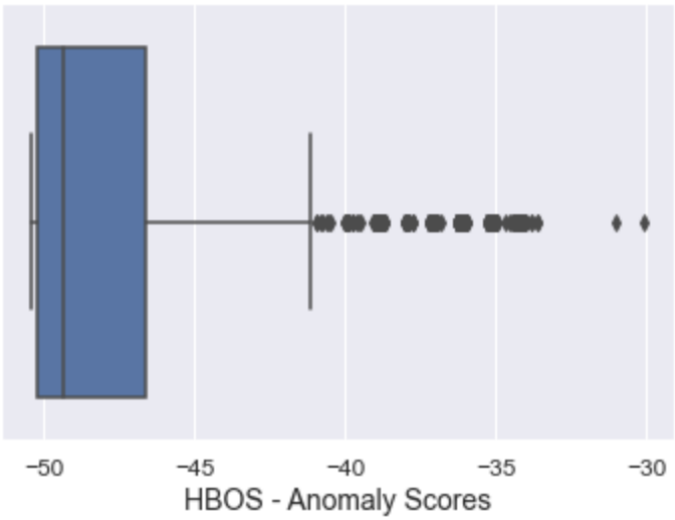

Figure 4.8 *HBOS – Boxplots depicting anomaly scores of training set*

Similar to EIF, higher anomaly scores (closer to the upper fence of the boxplot) meant that the datapoints were more likely to be anomalies. The anomaly labels were used to create a copy of the trainset sans outliers, which accounted for five percent of the original trainset. The remaining instances in the HBOS trainset were 4,331 at this stage.

4.3.3 Class Imbalance Treatment

Prior to building an LR model, SMOTE was employed for treating the imbalance in the target variable. We had four distinct versions of the trainset since model prediction had to be evaluated before and after the AD techniques were employed separately. Table 4.7 shows a detailed comparison of the trainset imbalance for all versions of the trainset (prior to and post AD).

Table 4.7 *Class imbalance of training set before and after SMOTE basis framework stage*

Class Imbalance – A Comparison			
Framework Stage	**Trainset Attributes**	**Before SMOTE**	**After SMOTE**
Before AD	*Number of samples*	4,558	6,956
	Class Imbalance	0: 76%	0: 50%
		1: 24%	1: 50%
After AD/ IF	*Number of samples*	4,330	6,928
	Class Imbalance	0: 80%	0: 50%
		1: 20%	1: 50%
After AD/ EIF	*Number of samples*	4,330	6,662
	Class Imbalance	0: 77%	0: 50%
		1: 23%	1: 50%
After AD/ HBOS	*Number of samples*	4,331	6,954
	Class Imbalance	0: 80%	0: 50%
		1: 20%	1: 50%

As shown in Table 4.7, the number of training samples depended on whether an AD technique was applied or not, which AD technique was applied, and whether the data imbalance technique SMOTE had already been applied.

Each of the AD techniques removed about five percent of their respective trainset's instances that were deemed anomalous in nature. In each trainset, the proportion of class imbalance varied prior to SMOTE but was equally balanced out once the technique was applied.

For DT, RF, and SVM which were built using sklearn, we used the in-built parameter 'class_weights' = 'balanced' which optimized the scoring for the minority class. XGBoost had its own unique hyperparameter 'scale_pos_weight' that controlled the weights for unbalanced classes. In order to separate the class imbalance treatments, we used separate copies of the trainset when building the LR model and the other models.

4.4 Prediction Module

This section looks at the steps taken for building different predictive models. It also looks at specific details such as hyperparameters that were used for tuning the performance of these models. In order to compare the impact of AD on model prediction, each model was built four different times on four distinct trainsets. The first build was prior to any AD technique being employed while the next three builds were using each of the trainsets processed by the three AD techniques – IF, EIF and HBOS.

4.4.1 Logistic Regression

We chose the statsmodels package to build our LR models from start to finish. The reason for this was statsmodels enabled LR models to be built iteratively while eliminating features that were either statistically insignificant or contributed to multicollinearity. Feature elimination was carried out one at a time with each iteration of model building, and the remaining features were used to build subsequent iterations. This was because the significance and multicollinearity of the remaining features were likely to stabilize each time a feature was dropped. As a result, we employed SMOTE to solve class imbalance for LR as the statsmodels package does not have an inbuilt class imbalance parameter, unlike sklearn. Once the final iterations of the models were arrived at, probability-based predictions of the target variable were generated and stored. The LR models were fine-tuned by calculating the optimal probability threshold which allowed a balance between different performance metrics. Statsmodels was used with default parameter settings to build the LR models.

4.4.2 Decision Tree

For building DT, we used GridSearchCV to assist in hyperparameter tuning as well as cross-validation of our trainset. Table 4.8 shows the hyperparameters and the corresponding values that were finalized for all four versions of DT (before and after AD).

Table 4.8 *Decision Tree - Hyperparameters used and corresponding values*

Decision Tree – Hyperparameters				
Hyperparameters	**Before AD**	**IF**	**EIF**	**HBOS**
max_depth	8	8	8	8
min_samples_split	120	160	160	160
min_samples_leaf	15	15	13	15
cv	4	4	4	4
scoring	'f1'	'f1'	'f1'	'f1'

The scoring hyperparameter was set to F1-score as that allowed for the DTs to be tuned for balance between the performance measures, precision, and recall. Four-fold cross-validation was used on the training sets to validate the efficiency of the model while tuning hyperparameters. A large range of values was used to experiment with the maximum depth of the DTs ('max_depth'), the minimum samples necessary prior to splitting a node ('min_samples_split'), and the minimum samples required in a leaf node ('min_samples_leaf'). Upon comparison of the values post tuning for each type of hyperparameter, we see a significant difference only in 'min_samples_split' with all other hyperparameters having similar values.

4.4.3 Random Forest

GridSearchCV was once again used for hyperparameter tuning and training using cross-validation for RF. The hyperparameters used in RF were the same as those used in DT with the exception of 'n_estimators', which specified the number of trees in the ensemble-based RF. Table 4.9 can be referred to for the finalized values of the hyperparameters.

Table 4.9 *Random Forest - Hyperparameters used and corresponding values*

Random Forest– Hyperparameters				
Hyperparameters	**Before AD**	**IF**	**EIF**	**HBOS**
max_depth	15	15	12	12
min_samples_split	28	5	8	8
min_samples_leaf	8	3	3	3
n_estimators	10	130	13	6
cv	4	4	4	4
scoring	'f1'	'f1'	'f1'	'f1'

The number of estimators varied significantly depending on whether AD was administered or not and if yes, the type of AD technique. All the other hyperparameters were more or less similar in values. It was observed that the 'min_samples_leaf' had a tendency to move towards very low values and therefore the range of values attempted was restricted to avoid overfitting.

4.4.4 XGBoost

In building XGBoost models as well, GridSearchCV was employed for hyperparameter tuning. Table 4.10 details the hyperparameters that were used across the different XGBoost builds.

Table 4.10 *XGBoost - Hyperparameters used and corresponding values*

XGBoost– Hyperparameters				
Hyperparameters	**Before AD**	**IF**	**EIF**	**HBOS**
objective	'binary:logistic'	'binary:logistic'	'binary:logistic'	'binary:logistic'
subsample	0.9	0.9	0.9	0.9
colsample_bytree	0.5	0.5	0.5	0.5
max_depth	4	4	4	5
learning_rate	0.1	0.7	0.5	0.15
gamma	0.5	0.1	0.65	0.25
reg_lamba	0.5	0.7	1.0	1.0
scale_pos_weight	2	2	2	2
cv	4	4	4	4
scoring	'f1'	'f1'	'f1'	'f1'
early_stopping_rounds	20	20	20	20
eval_metric	'auc'	'auc'	'auc'	'auc'

Most hyperparameters are different from those used in the previous ensemble-based models. ‘objective’ defined the learning task and the corresponding learning objective of our models, We set it to ‘binary:logistic’ which allowed our models to use a logistic regression approach for binary classification. ‘subsample’ set the percentage of random trainset samples that were used for training at 90%, enabling improved speed in cross-validation. Similarly, ‘colsample_bytree’ allowed a random 50% of trainset attributes to be used, which improved speed and helped prevent overfitting. ‘max_depth’ specified the depth of trees built in our models.

‘learning_rate’ controlled the step size between consecutive XGBoost trees built making the boosting process more conservative. ‘gamma’ and ‘reg_lambda’ were two other hyperparameters that controlled how conservatively the models were built. The former specified the minimum reduction in loss required to partition a leaf node and the latter was a regularization term for added weights. ‘scale_pos_weight’ was used for balancing the weights given to the imbalanced trainset. Unlike the previous ensemble-based models, the class imbalance could be improved here using hyperparameter tuning. Four-fold cross-validation was used and the F1 score was the choice of metric to evaluate improvements in the models while tuning (‘cv’ = 4 and ‘scoring’ = ‘f1’).

Unlike in RF where the number of trees was specified, we used a unique hyperparameter called ‘early_stopping_rounds’ in XGBoost that controlled the number of trees built based on improvements in predictions. In our case, we set it to 20 which meant that once the trees stopped improving, XGBoost built another 20 more to see if further improvements were possible. The improvements were measured based on the ‘eval_metric’ which was set to ‘auc’ (Area under the Curve for ROC) and the parameter ‘eval_set’ which stored our testing dataset. This allowed XGBoost models to be trained using the training set but evaluate the number of trees required using the performance against the testing set. As per the recommendation available in the XGBoost documentation for imbalanced datasets, we used ‘auc’ as the evaluation metric.

4.4.5 Support Vector Machines

The hyperparameters used in SVM can be referred to in Table 4.11. With SVM as well, we used GridSearchCV for tuning the hyperparameters. 'C' or 'Cost' is a regularization hyperparameter that we used to penalize misclassified trainset datapoints. It was interesting to note that the values were on the increasing side with SVM models built after AD. This perhaps indicated a higher tendency for misclassification requiring higher penalties.

Table 4.11 *Support Vector Machines - Hyperparameters used and corresponding values*

SVM – Hyperparameters				
Hyperparameters	**Before AD**	**IF**	**EIF**	**HBOS**
C	8	15	14	27
gamma	0.1	0.08	0.08	0.03
kernel	'rbf'	'rbf'	'rbf'	'rbf'
cv	4	4	4	4
scoring	'f1'	'f1'	'f1'	'f1'

We used a commonly employed 'kernel' called radial basis function (RBF), which is preferred to separate datapoints that are not linearly separable. 'gamma' was used to control the similarity radius that decided how rigid or lenient SVM was in grouping similar datapoints together. 'gamma' had low values, indicating leniency, across the different SVM models and post-AD SVM models had reduced values compared to pre-AD. A four-fold cross-validation was once again used and the improvements in tuning were based on the 'scoring' metric, which was the F1 score.

4.5 Summary

In this chapter, we looked at the several steps taken to build our models as per the proposed methodology framework. We started with the EDA where we noted that our dataset did not require missing and null value treatments. This was followed by data cleaning where we discussed the changes made in the 'gender' variable and the derived variable 'age_group' for ease of analysis. We then discussed the patterns that emerged from the univariate and bivariate analyses of the attributes in our dataset.

In the univariate analyses, we looked at the distribution of patient age, the percentage of patients falling into different age groups, the distribution of our target variable between the two different classes and repeated a similar analysis on the remaining binary attributes to understand their place and importance in our dataset. In the bivariate analyses, we paid attention to the differences that arise in binary attributes such as patient symptoms, isolation treatment, and travel history when compared against the target variable (infected vs non-infected patients). We similarly looked at how the patient's age group influences symptoms, Covid-19 infection, isolation treatment, and travel history.

We then dived into the steps taken for preparing the data prior to model building. The split ratio for the training and testing sets and their respective outcomes were specified. We looked at the hyperparameters that were used to build the three different AD techniques – IF, EIF and HBOS. The anomaly score distribution for each AD technique was visualized using boxplots. It was also explained how the AD techniques would be implemented on separate copies of the training set that would be used to create separate versions of each modelling technique. This enabled the comparison of performance against a baseline version of the modelling technique that used the original training set. Class imbalance treatment using SMOTE was then carried out prior to building LR.

We finally looked at the model building phase where the steps taken to build all four versions of each model, specifically the baseline model and three versions using AD, were discussed. We explored the different hyperparameters used, their role in tuning our modelling techniques, and the differences in the final values of each of the hyperparameters between the four versions were compared.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Introduction

This chapter primarily looks at the results from our impact analysis of the AD techniques that were employed on the different predictive models – LR, DT, RF, XGBoost, and SVM. We first look at each of the predictive models individually, prior to as well as post-application of AD. We then see how the performance of each predictive model is impacted in the training set as well as the testing set by comparing the baseline version of the model with AD-based versions of the model (IF, EIF, and HBOS versions). This is done based on our evaluation metrics, namely accuracy, precision, recall, and F1 score. After this, all the predictive models are compared together along with the best performing AD to see if the impacts, if any, in performance are consistent. The impact analysis is followed by a discussion and interpretation of the results where we condense our findings, look at some of the limitations that affected our research, and interpret how our results impact our objectives.

5.2 Impact of Anomaly Detection on Predictive Models

One of our objectives is to evaluate the impact AD techniques have on the predictive power of machine learning models. The remaining two objectives are answered based on the results of this evaluation. To analyze the impact, we built four different versions of each modelling technique, one version before AD and three versions using three different AD techniques. The performances of the different versions were mainly compared using four evaluation metrics, namely accuracy, precision, recall, and F1 score. In our experiments, the tuning of the models was done using the F1 score since early detection of Covid-19 required a balance between precision and recall. We wanted the models to generate relevant predictions (a good recall score) but not be overly conservative while predicting which patients were likely to have Covid-19 (an acceptable precision score).

In the following subsections, we will look at each model and compare the four different versions and their corresponding performance metrics.

5.2.1 Logistic Regression

Table 5.1 compares the different performance metrics for LR prior to and post AD. From a first look, it becomes evident that all three AD techniques have improved performance almost consistently across all four metrics. With accuracy, the improvement was highest in the training set when LR with EIF was used for predictions, increasing the metric from 0.821 to 0.867. With testing set though, it was LR with IF that improved the score the most from 0.833 to 0.858 with EIF and HBOS coming in at a very close second and third respectively. Recall however is the metric where significant improvement was achieved. LR with EIF improved the metric the most, increasing the score by over 10% points, from 0.781 to 0.894 and 0.770 to 0.896 in training and testing sets respectively. Precision was a different story where there was either no improvement or a slight drop in the training set scores but a two to three percentage points improvement in the testing set scores.

Table 5.1 *Logistic Regression – Comparison of performance metrics*

Logistic Regression – Evaluation Metrics					
Classifier	**Dataset Type**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
LR w/o AD	**Train**	0.821	0.848	0.781	0.813
	Test	0.833	0.639	0.770	0.699
LR with IF	**Train**	0.846	0.846	0.847	0.846
	Test	0.858	0.663	0.888	0.759
LR with EIF	**Train**	0.867	0.848	0.894	0.870
	Test	0.854	0.652	0.896	0.755
LR with HBOS	**Train**	0.842	0.842	0.842	0.842
	Test	0.853	0.652	0.888	0.752

F1 score being a balancing metric between precision and recall averaged out their gains. LR with EIF once again made the most improvement in the training set with F1 score going up from 0.813 to 0.870 and LR with IF topped the improvements in the testing set with an increase from 0.699 to 0.759. It was noted that although LR with HBOS made improvements compared to the baseline model, in comparison to the other AD techniques the gains were among the lowest.

5.2.2 Decision Tree

A comparison of performance metrics of the different versions of DT can be referred to in Table 5.2. A quick overview of the metrics told us that all of the AD techniques had generally either not improved or had a detrimental effect on the performance of DT across all metrics.

Table 5.2 *Decision Tree – Comparison of performance metrics*

Decision Tree– Evaluation Metrics					
Classifier	**Dataset Type**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
DT w/o AD	**Train**	0.884	0.698	0.899	0.786
	Test	0.877	0.708	0.872	0.781
DT with IF	**Train**	0.882	0.656	0.861	0.745
	Test	0.885	0.727	0.868	0.791
DT with EIF	**Train**	0.883	0.691	0.892	0.779
	Test	0.880	0.714	0.872	0.785
DT with HBOS	**Train**	0.880	0.648	0.859	0.739
	Test	0.885	0.727	0.868	0.791

Accuracy for the training set had more or less stayed the same across the different versions. In the case of the testing set accuracy improved very slightly across all DT models post AD, with the IF and HBOS versions tying at the top by increasing the score from 0.877 to 0.885. The inconsistency in performance became evident when we looked at precision and recall. In the training set, DT with EIF was the only version that managed to score a precision close to the baseline model. The IF and HBOS versions dropped the scores by four to five percentage points. In the testing set, the story was the other way around. The HBOS and IF versions tied and made the most improvements in precision by increasing the score from 0.708 to 0.727 while DT with EIF made relatively marginal improvements.

In the case of recall, the story was similar for the training set where the baseline model did better than all three AD versions. DT with EIF dropped the recall score from 0.899 to 0892 making it the least reduction in performance while the IF and HBOS versions experienced a three to four percent reduction. With the testing set, none of the AD versions improved recall over the baseline model but they hardly experienced any drops in the score either. All the versions of the DT models were tuned to maximize the F1 score and yet all of the AD versions saw a reduced metric. DT with EIF once again saw a minor drop from 0.786 to 0.779 while the IF and HBOS versions saw four and five percentage points reductions respectively.

5.2.3 Random Forest

The different versions of RF and their respective performance metrics have been compared in Table 5.3. In terms of improvements, the results for RF overall were a mixed bag. In the training set, all the AD-based models improved accuracy compared to the baseline model, with the highest improvement being 0.869 to 888 by RF with EIF. In the case of precision, only the EIF version managed an improvement in comparison to the baseline model (0.664 to 0.698) while both IF and HBOS versions saw drops (one percent and three percentage points respectively). Recall on the other hand saw a marginal decrease with the IF version and a two percentage points drop with HBOS. Only RF with EIF equaled the baseline model's score but did not improve over it. With the F1 score as well, RF with EIF was the only model that improved compared to the baseline model (0.767 to 0.789). RF with IF saw a marginal drop (0.767 to 0.759) and RF with HBOS saw more than three percentage points drop (0.767 to 0.731).

Table 5.3 *Random Forest – Comparison of performance metrics*

Random Forest– Evaluation Metrics					
Classifier	**Dataset Type**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
RF w/o AD	**Train**	0.869	0.664	0.909	0.767
	Test	0.865	0.678	0.888	0.769
RF with IF	**Train**	0.885	0.654	0.903	0.759
	Test	0.872	0.696	0.874	0.774
RF with EIF	**Train**	0.888	0.698	0.908	0.789
	Test	0.876	0.706	0.866	0.778
RF with HBOS	**Train**	0.871	0.622	0.888	0.731
	Test	0.863	0.678	0.868	0.761

In the case of the testing set, the accuracy metric saw a marginal drop with HBOS, a small increase with IF, and, in relative terms, the most improvement was once again by RF with EIF (0.865 to 0.876). Precision improved with both the IF and EIF versions (from 0.678 to 0.696 and 0.706 respectively) but HBOS tied with the baseline score. Recall on the other hand saw decreases across all AD versions compared to the baseline score with the lowest coming from RF with EIF (0.888 to 0.866). F1 score saw improvements from both the IF and EIF versions, the latter managing the highest (0.769 to 0.778) while RF with HBOS dropped in performance by almost one percentage point.

5.2.4 XGBoost

Performance metrics for XGBoost were compared across versions and can be referred to in Table 5.4. Overall, the AD-based XGBoost models have made only marginal improvements compared to the baseline model. In the training set, only XGBoost with EIF managed to increase the accuracy score compared to the baseline from 0.888 to 0.894. The IF version matched the baseline while the HBOS version marginally decreased the score. With precision, once again it was XGBoost with EIF that improved the training score from 0.706 to 0.720 while the other two AD versions dropped in performance by three to four percent.

Table 5.4 *XGBoost – Comparison of performance metrics*

XGBoost– Evaluation Metrics					
Classifier	**Dataset Type**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
XGBoost w/o AD	**Train**	0.888	0.706	0.900	0.791
	Test	0.877	0.707	0.874	0.782
XGBoost with IF	**Train**	0.888	0.673	0.859	0.755
	Test	0.886	0.734	0.862	0.793
XGBoost with EIF	**Train**	0.894	0.720	0.883	0.793
	Test	0.885	0.741	0.835	0.785
XGBoost with HBOS	**Train**	0.885	0.659	0.863	0.747
	Test	0.880	0.720	0.858	0.783

However, the recall was a metric where we could see the use of the AD techniques resulted in reduced performance. XGBoost with EIF reduced the score by nearly two percentage points (0.900 to 0.883) while the IF and HBOS versions had their scores reduced by nearly four percentage points. With the F1 score, once again only XGBoost with EIF made improvements, albeit marginal (0.791 to 0.793). Accuracy for IF and HBOS versions dropped by four and five percentage points respectively.

In the testing set, the improvements in the performance metrics became more evident. Accuracy showed marginal improvement across all versions of AD-based models with the IF version taking the top score (increased from 0.877 to 0.886). Precision scores too showed improvement across all versions and XGBoost with EIF made the largest increase (from 0.707 to 0.741). With recall, we saw results to be similar to that observed in the training set where all three AD-based models dropped in performance. Among the three, XGBoost with IF managed a relatively smaller decrease in the score (from 0.874 to 0.862), and XGBoost with EIF had the largest decrease (from 0.874 to 0.835). Unlike in the training set, the F1 score saw XGBoost with IF making a relatively larger improvement (from 0.782 to 0.793) and both EIF and HBOS versions making marginal improvements (0.785 and 0.782 respectively).

5.2.5 Support Vector Machines

We once again compared the performance metrics between all the versions of our model. Table 5.5 provides specifics of the comparison for all SVM-based models. It was observed that all of the AD techniques overall had a detrimental effect on nearly every metric that was measured. In the training set, accuracy was either very close to the baseline model's (0.880 by SVM with EIF vs baseline score of 0.881) or reduced with both the IF and HBOS versions achieving scores of 0.876 and 0.857 respectively. Precision saw a more dramatic reduction with HBOS managing only a score of 0.590 vs the baseline of 0.686. SVM with IF too only achieved a reduced score of 0.633. The EIF version too lost nearly one percentage point (0.677) against the baseline score. While the reductions were not as significant, recall too saw drops in performance. The largest reduction was once again seen with the HBOS version (0.919 to 0.898). SVM with EIF saw the least reduction from 0.919 to 0.913.

Table 5.5 *Support Vector Machines – Comparison of performance metrics*

Support Vector Machines– Evaluation Metrics					
Classifier	**Dataset Type**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
SVM w/o AD	**Train**	0.881	0.686	0.919	0.785
	Test	0.861	0.673	0.874	0.760
SVM with IF	**Train**	0.876	0.633	0.905	0.745
	Test	0.856	0.669	0.846	0.747
SVM with EIF	**Train**	0.880	0.677	0.913	0.777
	Test	0.854	0.671	0.821	0.738
SVM with HBOS	**Train**	0.857	0.590	0.898	0.712
	Test	0.852	0.653	0.884	0.751

F1 score saw a pattern similar to that observed with precision. Once again, a significant reduction from the baseline score was seen with the HBOS version (0.785 vs 0.712). SVM with IF saw reduced performance (0.745). SVM with EIF too lost nearly one percentage point (0.777).

In the testing set, the loss of performance was relatively to a less degree. All three AD-based models saw drops in performance by up to one percentage point when compared to the baseline model (0.861). Losses in precision were to a similar degree with the IF and EIF versions scoring 0.669 and 0.671 against the baseline score of 0.673 while SVM with HBOS lost two percentage points. For recall, surprisingly only SVM with HBOS gained in performance (from 0.874 to 0.884) while IF and EIF dropped in performance by three and five percent respectively.

For the F1 score, the least drop was observed with the HBOS version compared to the baseline (reduced from 0.760 to 0.751). The IF and EIF versions saw performance drop by over one and two percentage points respectively.

5.2.6 Comparison of Best Performing AD Technique between Predictive Models

In the previous subsections, we analyzed the impact of all three AD techniques individually on each of our chosen predictive models. We saw that, in general, AD with EIF performed better in most metrics than the IF and HBOS versions. Based on these results, we decided to pick the best AD-based models and compare them against their corresponding baseline models. Since all versions of our models were tuned to perform better with the F1 score, the best among the three AD-based models was also picked using this metric. Table 5.6 makes this comparison for the training set. The best F1 score for each model has been highlighted.

Table 5.6 *Comparison of Models – Training Set – Baseline Model vs Best AD based Model*

Comparison of Predictive Models – Training Set - Baseline Model vs Best AD based Model					
Classifier	**Without AD/ Best AD Method**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
LR	Without AD	0.821	0.848	0.781	0.813
	AD With EIF	0.867	0.848	0.894	**0.870**
DT	Without AD	0.884	0.698	0.899	**0.786**
	AD With EIF	0.883	0.691	0.892	0.779
RF	Without AD	0.869	0.664	0.909	0.767
	AD With EIF	0.888	0.698	0.908	**0.789**
XGBoost	Without AD	0.888	0.706	0.900	0.791
	AD With EIF	0.894	0.720	0.883	**0.793**
SVM	Without AD	0.881	0.686	0.919	**0.785**
	AD With EIF	0.880	0.677	0.913	0.777

In the training set, across all predictive models, it was the EIF version that performed the best among the three AD methods. Three out of the five models were seen to benefit from the use of AD on the training set. For LR, it is evident across all metrics and not just the F1 score (0.813 to 0.870) that LR with EIF performed significantly better than the baseline model. A nearly six percentage point increase in the F1 score made LR with EIF a clear winner. For DT, it was the baseline model that performed the best. Only the EIF version came close enough to the baseline F1 score but still witnessed a drop in performance (0.786 to 0.779). Similar patterns were seen with the other metrics as well, except for recall where EIF made marginal improvements. For RF, we once again saw that the EIF version performed better than the baseline F1 score (an increase from 0.767 to 0.789). The same applied for other metrics as well where EIF either improved or at least matched the baseline metric.

For XGBoost, EIF managed only a marginal increase compared to the baseline F1 score indicating that the impact of AD was hardly significant. Accuracy and precision saw relatively larger increases while recall saw a drop compared to the baseline model. With SVM, the baseline model performed better in all of the metrics and while the margin of difference between the baseline and EIF was not significantly high, it was clear that the AD technique had a detrimental effect on performance.

A similar comparison was made for the testing set as well. Table 5.7 can be referred to for the specifics. Four out of the five predictive models were seen to benefit from the use of AD on the testing set. However, this time the best performing AD technique varied based on the modelling technique. For LR, it was the IF version that made significant gains. There were once again clear improvements visible with the F1 score improving from 0.699 to 0.759 as well as with the other metrics. Recall particularly stood out with an increase from 0.770 to 0.888. In the case of DT, unlike in the training set, both the IF and HBOS versions equally improved the F1 score from 0.781 to 0.791. In fact, the improvements were identical for both AD versions such that the scores were identical across all four metrics. Except for recall, the remaining metrics made marginal improvements compared to the baseline.

Table 5.7 *Comparison of Models – Testing Set – Baseline Model vs Best AD based Model*

Comparison of Predictive Models – Testing Set - Baseline Model vs Best AD based Model					
Classifier	**Without AD/ Best AD Method**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
LR	Without AD	0.833	0.639	0.770	0.699
	AD With IF	0.858	0.663	0.888	**0.759**
DT	Without AD	0.877	0.708	0.872	0.781
	AD With IF/HBOS	0.885	0.727	0.868	**0.791**
RF	Without AD	0.865	0.678	0.888	0.769
	AD With EIF	0.876	0.706	0.866	**0.778**
XGBoost	Without AD	0.877	0.707	0.874	0.782
	AD With IF	0.886	0.734	0.862	**0.793**
SVM	Without AD	0.861	0.673	0.874	**0.760**
	AD With HBOS	0.852	0.653	0.884	0.751

In the case of RF, AD with EIF delivered the highest performance compared to the baseline model with an F1 score increased from 0.769 to 0.778. Once again only recall saw a drop in performance, while accuracy and precision saw improvements. With XGBoost, it was once again IF that delivered the best improvements with an F1 score increase from 0.782 to 0.793. Here too, recall slightly dropped while accuracy and precision improved. SVM was the only model which did not see any of the AD techniques improve over the baseline model both in the training and testing sets. Surprisingly, SVM with HBOS managed the highest F1 score of the three AD techniques even though it didn't improve over the baseline (decreased from 0.760 to 0.751).

5.3 Results Discussion & Interpretation

The impact analysis of the three AD techniques was carried out in the previous section. In deciding whether AD had a positive impact or not on predictive power, focusing on the results of the testing set would better tell us how the models would perform in real-world data. As analyzed in the previous section using data from Table 5.7, we saw that LR significantly gained from the use of AD techniques while the gains are relatively smaller with DT, RF, and XGBoost. SVM was the only model that saw its performance drop because of AD although the decrease is less than a percentage point. This tells us that there isn't one ideal AD technique that suits all of our models equally.

In the testing set, it is IF-based models that brought about the highest improvements in model performance, with three out of five models benefiting the most from IF. This is an interesting turn of events since we saw that the EIF based models dominated performance in the training set. This was unexpected since the EIF is a modified version of IF with some of the drawbacks of the latter addressed. One explanation could be that the sklearn based IF implementation that we used is more mature than the H2O-based EIF, especially given that EIF itself is a relatively recently developed technique. A second reason could be that EIF itself may require hyperparameter tuning to arrive at tuned values for hyperparameters like 'n_trees' (number of estimators/trees) and 'extension_level', which defines how many of the available training set features should be used for identifying anomalies. However, in the testing set, it can be noted that the difference in the performance of the IF and EIF versions across all predictive models is quite small. Except for SVM, we can see that both IF and EIF have made comparable improvements in the F1 score when compared to the baseline model. It is worth pointing out at this juncture that HBOS rarely made improvements compared to the baseline model and often its performance lagged the IF and EIF versions of all models. This tells us that HBOS is not suitable for categorical/binary data.

At this point, we should note that there is a possibility that due to the limited number of records in our dataset, we may not have seen the full extent of the impact of our AD techniques on model performance. Our training set was already at 70% of our limited-sized dataset (6,512 instances) and a further 5% removal of the training set due to AD could have affected the predictive model's ability to learn.

The remaining 30% was used for testing (1,954 instances) which is a small number of instances that makes it difficult to arrive at conclusions with confidence.

With the detailed impact analysis in the previous section and summarized above, we have achieved the second objective of our research by analyzing the impact of three different AD techniques on five different predictive models.

The first objective of our research is to suggest a suitable AD technique for our categorical dataset. From the above summarization, we can recommend IF as the AD technique that can be used along with four of our five predictive models, namely LR, DT, RF, and XGBoost. For SVM, based on the current research we are unable to recommend an AD technique that improves its performance. Further analysis using a larger dataset and/or further research into other available AD techniques may be necessary before recommending AD for SVM.

The third objective has been partially fulfilled since we used AD to improve the prediction capabilities of four of our five proposed models. Since we were unable to improve SVM's capability using any of the three AD techniques, this objective remains incomplete for SVM alone.

5.4 Summary

In this chapter, we analyzed the impact of three different AD techniques on five distinct predictive models. We compared the evaluation metrics first for each model and discussed how each of the AD techniques had an impact on the performance compared to the baseline model's scores. We noted the variations in performance for each metric, how they were different between the training and testing sets, and picked a top performer for each metric from the baseline model and the three AD-based versions of the model. We saw that DT and SVM witnessed a reduced performance in the important F1 score for the training set but SVM alone consistently carried over the drop in performance to the testing set as well.

We then did a comparison of the metrics across all models, both in the training and testing sets, between the baseline scores for each model against the corresponding AD-based model that performed best among the three. It was observed that EIF was consistently the top performing AD technique in the training set but IF generally performed better than EIF in the testing set. The performance of the HBOS version was observed to lag IF, EIF, and even the baseline model and was therefore deemed inefficient for use in a categorical/binary dataset. Possible limitations of the deployed EIF technique and limitations of the size of the dataset were discussed. Finally, we reviewed each objective and the extent to which they were achieved with our analysis.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

In this chapter, we will begin by summarizing and discussing the work done as part of our research. We will formulate our conclusions and answer each of our objectives. This will be followed by detailing our contributions to existing research and recommendations for the future.

6.2 Discussion and Conclusion

In our research, we endeavoured to develop a methodology that took advantage of AD to improve the detection of Covid-19 at an early stage using a categorical dataset. As part of this effort, we first surveyed the available literature on ML and DL-based techniques that worked on the detection and mitigation of Covid-19. We also looked at the work done in ML-based AD and how they were employed in different types of anomaly scenarios. Ensemble-based AD techniques such as IF and EIF, in particular, were found to be useful in scenarios where categorical/binary data was employed. From our review of the literature, we found that research was yet to be done on the impact of AD on the prediction of Covid-19 using a categorical dataset. Post the survey, we developed a framework that helped us evaluate the impact of AD on ML models in predicting Covid-19 using a categorical/binary dataset. Our methodology made use of three different AD techniques (IF, EIF, and HBOS) that we found to be compatible for use with a categorical dataset and were separately employed to work with five distinct predictive models (LR, DT, RF, XGBoost, and SVM). To evaluate the performance of AD, we chose four different metrics (accuracy, precision, recall, and F1 score).

In order to implement the framework, we proceeded with our methodology where we began by carrying out EDA to derive meaningful insights from our data. This was followed by data preparation where we addressed the class imbalance and developed all three AD techniques for use in the training set. Subsequent to these steps, we built one version of each model without the use of AD to serve as a baseline and three more versions with each using a distinct AD technique so as to compare the difference in performance.

As part of the model development process, different hyperparameters were used for tuning the performance of the model and the differences between hyperparameter values were compared.

Post model development, as part of the last stage of our framework we evaluated the impact of the three AD techniques on the five predictive models. We used the evaluation metrics to first understand the impact of the AD techniques for each predictive model in the training and testing sets. This was followed by comparison across all models to better understand the impact of AD on the predictive power in a categorical/binary dataset.

With the implementation of our methodology and the evaluation that followed, we were able to achieve our objectives as follows. We carried out an impact analysis on the differences in performance of our five predictive models when AD is administered, by comparing their baseline models to their corresponding AD-based models. This analysis and evaluation fulfilled our second objective. The first objective was to recommend an AD technique suitable for our Covid-19 categorical dataset. For this, we had to evaluate whether the chosen AD technique did not have a detrimental effect on the quality of the dataset. We achieved this through our impact analysis, where we identified IF as an AD technique that is capable of finding anomalies in a categorical-natured dataset as well as enhancing the performance of four out of five predictive models.

The third objective was to improve the prediction of Covid-19 at an early stage using our dataset. We were able to achieve this with four of our predictive models (LR, DT, RF, XGBoost) with the help of the AD using IF. The impact analysis which compared the baseline versions of these models against the IF-based versions saw performance improvements in the latter when evaluated against the testing set (akin to real-world data). However, we were unable to improve the prediction of Covid-19 using SVM-based modelling. This leaves the third objective partially fulfilled.

6.3 Contribution to Knowledge

Our study reviewed literature that showed the use of AD was predominantly restricted to modelling in quantitative data. The research we undertook was able to establish that AD has a useful role to play in datasets that are categorical in nature as well. It was also demonstrated that integrating AD into the methodology of building models for categorical datasets, specifically IF in our case, could improve the prediction of Covid-19 at an early stage.

As stated earlier, detection and mitigation of Covid-19 at an early stage remain a crucial strategy for healthcare workers. Improvements to the use of model-based predictions to identify Covid-19 patients only serve to strengthen the strategy. AD can play an important role in this endeavour when dealing with categorical datasets as well.

6.4 Future Recommendations

The outcomes of the research were restricted by the limited size of the dataset (6,512 samples). It is recommended that a larger dataset be used to better understand the impact AD has on different modelling techniques. Apart from the already tried AD techniques, XGBOD was one other technique that seemed suitable for use in categorical datasets based on our literature review. It is an ensemble-based technique capable of combining inputs from varying AD techniques and applying weights to each of their inputs, similar to a feature selection process where inputs are pruned to control for complexity and increase accuracy. However, at this stage, there is no referenceable documentation on the limited implementations available for use. This could change in the near future and XGBOD's capabilities too can be tested on categorical datasets using our proposed framework. Further research into other AD techniques for the categorical dataset will need to be carried out to find compatible options for modelling techniques such as SVM (including XGBOD).

REFERENCES

Afm, B., Jl, M., Thr, D., Filho, C., André, A., De Moraes Batista, F., Luiz Miraglia, J., Dias, A. and Filho, P.C., (2020) COVID-19 diagnosis prediction in emergency care patients: a machine learning approach. [online] Available at: https://doi.org/10.1101/2020.04.04.20052092 [Accessed 8 Aug. 2021].

Aggarwal, C.C., (2017) High-Dimensional Outlier Detection: The Subspace Method. In: *Outlier Analysis*. [online] Cham: Springer International Publishing, pp.149–184. Available at: http://link.springer.com/10.1007/978-3-319-47578-3_5.

Ahamad, M., (2021) *Early stage symptoms of COVID-19 patients*. [online] Kaggle. Available at: https://www.kaggle.com/martuza/early-stage-symptoms-of-covid19-patients [Accessed 12 Dec. 2021].

Ahamad, M.M., Aktar, S., Rashed-Al-Mahfuz, M., Uddin, S., Liò, P., Xu, H., Summers, M.A., Quinn, J.M. and Moni, M.A., (2020) A machine learning model to identify early stage symptoms of SARS-Cov-2 infected patients. *Expert Systems with Applications*, [online] 160, p.113661. Available at: https://doi.org/10.1016/j.eswa.2020.113661 [Accessed 28 Jul. 2021].

Ali, O., Ishak, M.K. and Bhatti, M.K.L., (2021) Early COVID-19 symptoms identification using hybrid unsupervised machine learning techniques. *Computers, Materials and Continua*, 691, pp.747–766.

Anello, E., (2021) *Anomaly Detection With Isolation Forest*. [online] Medium. Available at: https://betterprogramming.pub/anomaly-detection-with-isolation-forest-e41f1f55cc6 [Accessed 19 Oct. 2021].

Anon (2021) *Extended Isolation Forest — H2O 3.34.0.4 documentation*. [online] H2O.ai, Inc. Available at: https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/eif.html [Accessed 4 Dec. 2021].

Birgé, L. and Rozenholc, Y., (2006) How many bins should be put in a regular histogram. *ESAIM: Probability and Statistics*, [online] 10, pp.24–45. Available at: http://www.esaim-ps.org/10.1051/ps:2006001.

Breiman, L., (2001) *Random Forests*.

Brownlee, J., (2020) *Why Is Imbalanced Classification Difficult?* [online] Available at: https://machinelearningmastery.com/imbalanced-classification-is-hard/ [Accessed 12 Aug. 2021].

Campos, G.O., Zimek, A. and Meira, W., (2018) An Unsupervised Boosting Strategy for Outlier Detection Ensembles. [online] pp.564–576. Available at: http://link.springer.com/10.1007/978-3-319-93034-3_45.

Candel, A., Ledell, E. and Bartz, A., (2020) *Deep Learning with H2O*. [online] *H2O.ai, Inc*. Available at: http://h2o.ai/resources/ [Accessed 4 Dec. 2021].

Chen, T. and Guestrin, C., (2016) XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [online] New York, NY, USA: ACM, pp.785–794. Available at: http://dx.doi.org/10.1145/2939672.2939785.

Cortes, C., Vapnik, V. and Saitta, L., (1995) Support-Vector Networks. *Machine Leaming*, 20,

pp.273–297.

Cramer, J.S., (2003) The Origins of Logistic Regression. *SSRN Electronic Journal*. [online] Available at: http://www.ssrn.com/abstract=360300.

Dogan, O., Tiwari, S., Jabbar, M.A. and Guggari, S., (2021) A systematic review on AI/ML approaches against COVID-19 outbreak. *Complex & Intelligent Systems*. [online] Available at: https://doi.org/10.1007/s40747-021-00424-8.

Else, H., (2020) How a torrent of COVID science changed research publishing — in seven charts. *Nature*, [online] 5887839, pp.553–553. Available at: http://www.nature.com/articles/d41586-020-03564-y.

Fleiss, J.L., Williams, J.B.W. and Dubro, A.F., (1986) The logistic regression analysis of psychiatric data. *Journal of Psychiatric Research*, [online] 203, pp.195–209. Available at: https://linkinghub.elsevier.com/retrieve/pii/0022395686900038.

Flowick, V., (2018) *How to use machine learning for anomaly detection and condition monitoring | by Vegard Flovik | Towards Data Science*. [online] Available at: https://towardsdatascience.com/how-to-use-machine-learning-for-anomaly-detection-and-condition-monitoring-6742f82900d7 [Accessed 12 Aug. 2021].

Goldstein, M. and Dengel, A., (2012) Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm. [online] Available at: http://madm.dfki.de/rapidminer/anomalydetection. [Accessed 15 Oct. 2021].

Hariri, S., Kind, M.C. and Brunner, R.J., (2021) Extended Isolation Forest. *IEEE Transactions on Knowledge and Data Engineering*, [online] 334, pp.1479–1489. Available at: https://ieeexplore.ieee.org/document/8888179/.

Huang, H., Mehrotra, K. and Mohan, C.K., (2013) Rank-based outlier detection. *Journal of Statistical Computation and Simulation*, [online] 833, pp.518–531. Available at: http://www.tandfonline.com/doi/abs/10.1080/00949655.2011.621124.

Islam, M.Z., Islam, M.M. and Asraf, A., (2020) A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. *Informatics in Medicine Unlocked*, [online] 20, p.100412. Available at: https://doi.org/10.1016/j.imu.2020.100412.

Liu, F.T., Ting, K.M. and Zhou, Z.-H., (2008) Isolation Forest. In: *2008 Eighth IEEE International Conference on Data Mining*. [online] IEEE, pp.413–422. Available at: http://ieeexplore.ieee.org/document/4781136/.

Mascaro, S., Nicholson, A. and Korb, K., (2014) Anomaly detection in vessel tracks using Bayesian networks. *International Journal of Approximate Reasoning*, 551 PART 1, pp.84–98.

Mohamed, M.S. and Kavitha, T., (2011) Outlier Detection Using Support Vector Machine in Wireless Sensor Network Real Time Data. *International Journal of Soft Computing and Engineering (IJSCE)*, 12, pp.68–72.

Mohammed, M.A., Abdulkareem, K.H., Garcia-Zapirain, B., Mostafa, S.A., Maashi, M.S., Al-Waisy, A.S., Subhi, M.A., Mutlag, A.A. and Le, D.N., (2021) A comprehensive investigation of machine learning feature extraction and classification methods for automated diagnosis of COVID-19 based on X-ray images. *Computers, Materials and Continua*, 663, pp.3289–3310.

Pedregosa FABIANPEDREGOSA, F., Michel, V., Grisel OLIVIERGRISEL, O., Blondel, M., Prettenhofer, P., Weiss, R., Vanderplas, J., Cournapeau, D., Pedregosa, F., Varoquaux, G.,

Gramfort, A., Thirion, B., Grisel, O., Dubourg, V., Passos, A., Brucher, M., Perrot andÉdouardand, M., Duchesnay, andÉdouard and Duchesnay EDOUARDDUCHESNAY, Fré., (2011) Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot. *Journal of Machine Learning Research*, [online] 12, pp.2825–2830. Available at: http://scikit-learn.sourceforge.net. [Accessed 16 Oct. 2021].

Rayana, S., Zhong, W. and Akoglu, L., (2016) Sequential Ensemble Learning for Outlier Detection: A Bias-Variance Perspective. In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. [online] IEEE, pp.1167–1172. Available at: http://ieeexplore.ieee.org/document/7837967/.

Ripan, R.C., Sarker, I.H., Hossain, S.M.M., Anwar, M.M., Nowrozy, R., Hoque, M.M. and Furhad, M.H., (2021) A Data-Driven Heart Disease Prediction Model Through K-Means Clustering-Based Anomaly Detection. *SN Computer Science*, [online] 22, pp.1–12. Available at: https://doi.org/10.1007/s42979-021-00518-7.

Ruff, L., Kauffmann, J.R., Vandermeulen, R.A., Montavon, G., Samek, W., Kloft, M., Dietterich, T.G. and Muller, K.-R., (2021) A Unifying Review of Deep and Shallow Anomaly Detection. *Proceedings of the IEEE*, [online] 1095, pp.756–795. Available at: https://www.statista.com/.

Sharma, N., (2021) *Bernoulli Naive Bayes and it's implementation*. [online] Available at: https://medium.com/@nansha3120/bernoulli-naive-bayes-and-its-implementation-cca33ccb8d2e [Accessed 13 Aug. 2021].

Song, H., Jiang, Z., Men, A. and Yang, B., (2017) A Hybrid Semi-Supervised Anomaly Detection Model for High-Dimensional Data. *Computational Intelligence and Neuroscience*, [online] 2017, pp.1–9. Available at: https://www.hindawi.com/journals/cin/2017/8501683/.

Sun, L., Versteeg, S., Boztas, S. and Rao, A., (2016) Detecting Anomalous User Behavior Using an Extended Isolation Forest Algorithm: An Enterprise Case Study. [online] Available at: http://arxiv.org/abs/1609.06676.

Suri, N.N.R.R., Murty, M.N. and Athithan, G., (2013) A ranking-based algorithm for detection of outliers in categorical data. *International Journal of Hybrid Intelligent Systems*, [online] 111, pp.1–11. Available at: https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/HIS-130179.

Taha, A. and Hadi, A.S., (2019) Anomaly Detection Methods for Categorical Data: A Review. *ACM Computing Surveys*, [online] 522, pp.1–35. Available at: https://doi.org/10.1145/3312739.

Thomas, R., (2020) A Novel Ensemble Method for Detecting Outliers in Categorical Data. *International Journal of Advanced Trends in Computer Science and Engineering*, [online] 94, pp.4947–4953. Available at: http://www.warse.org/IJATCSE/static/pdf/file/ijatcse108942020.pdf.

Turkoglu, M., (2021) COVIDetectioNet: COVID-19 diagnosis system based on X-ray images using features selected from pre-learned deep features ensemble. *Applied Intelligence*, [online] 513, pp.1213–1226. Available at: https://doi.org/10.1007/s10489-020-01888-w [Accessed 30 Jul. 2021].

Verbus, J., (2019) *Detecting and preventing abuse on LinkedIn using isolation forests*. [online] Linkedin Engineering. Available at: https://engineering.linkedin.com/blog/2019/isolation-

forest [Accessed 19 Oct. 2021].

WALDMAN, I.D., ROBINSON, B.F. and ROWE, D.C., (1999) A logistic regression based extension of the TDT for continuous and categorical traits. *Annals of Human Genetics*, [online] 634, pp.329–340. Available at: http://doi.wiley.com/10.1046/j.1469-1809.1999.6340329.x.

Wang, H., Bah, M.J. and Hammad, M., (2019) Progress in Outlier Detection Techniques: A Survey. *IEEE Access*, [online] 7, pp.107964–108000. Available at: https://ieeexplore.ieee.org/document/8786096/.

Xu, X., Liu, H. and Yao, M., (2019) Recent Progress of Anomaly Detection. *Complexity*, [online] 2019, pp.1–11. Available at: https://www.hindawi.com/journals/complexity/2019/2686378/.

Zhang, J., Xie, Y., Pang, G., Liao, Z., Verjans, J., Li, W., Sun, Z., He, J., Li, Y., Shen, C. and Xia, Y., (2020) COVID-19 Screening on Chest X-ray Images Using Deep Learning based Anomaly Detection. [online] Available at: http://arxiv.org/abs/2003.12338 [Accessed 9 Aug. 2021].

Zhao, Y. and Hryniewicki, M.K., (2018) XGBOD: Improving Supervised Outlier Detection with Unsupervised Representation Learning. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. [online] IEEE, pp.1–8. Available at: https://ieeexplore.ieee.org/document/8489605/.

Zhao, Y., Nasrullah, Z. and Li, Z., (2019) PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research*, [online] 20, pp.1–7. Available at: https://pyod.readthedocs.io [Accessed 16 Oct. 2021].

APPENDIX A: RESEARCH PROPOSAL

Abstract

Our world is in the middle of a pandemic due to Covid-19, a highly contagious disease with limited options for treatment. Early detection and isolation are considered the primary strategy in containing its spread. This will require a capacity to make early and informed decisions with the highest accuracy possible from available data on patient characteristics and other data points related to diagnostics and early symptoms. As part of this endeavour, our study will apply an Anomaly Detection (AD) model/technique on a Covid-19 dataset that is categorical in nature and evaluate its effect on the performance of several Machine Learning (ML) models such as Logistic Regression, Naïve Bayes & Random Forest. The evaluation of ML models is based on their performance prior to and post-removal of anomalies using the AD model/technique. The expectation is that AD based ML model training is a more efficient process and can result in significant gains in prediction performance. While similar or related methodologies may have been previous conducted on quantitative data, our study will test the effects of AD on a categorical dataset.

1. Background

The advent of the Covid-19 pandemic resulted in a worldwide disruption of life as we know it. Wide ranging impacts have been observed in healthcare systems, economic health, gender equality, public health and many others. Research work in the year of 2020 revolved significantly around Covid-19. One estimate suggested that around 4% of the world's research was devoted to coronavirus with the bulk of it initially focused on disease spread, hospitalization outcomes and diagnostics and mitigatory strategies such as testing (Else, 2020).

Artificial Intelligence (AI)/ML has been applied significantly in the epidemiological research of Covid-19. ML has been used to identify variables, explain interconnections that help understand the key reasons behind Covid-19 cases and related deaths. Deep Learning (DL) techniques, such as Long short-term memory (LSTM), were found effective in detecting Covid-19 using time series data. Chest X-rays and CT scans are popular in screening for Covid-19 using Convolutional Neural Network (CNN) or hybrid DL techniques. AI/ML has also been applied in other interesting ways such as finding new molecules that help confirm Covid-19 cases, embedded AI in cameras and smartphones to detect infected people and even use of drones for transportation of food and medicines in areas with infection (Dogan et al., 2021).

The research into AI/ML approaches to solve issues related to Covid-19 grows by the day but the use of AD to assist in the process remains relatively low. Anomalies or outliers are objects that are not consistent with the pattern of the majority of the instances in the dataset. It can be quite useful to detect anomalies in data because they are capable of distorting analysis and predictive power of trained models thereby affecting decision making. Significant research has been done in the field of anomaly/outlier detection since its study is of importance to multiple disciplines which include statistics, data mining and machine learning among others. AD has innumerable applications in various domains such as cybersecurity, fraud detection, healthcare, medical diagnosis and disease outbreak detection to name a few (Ruff et al., 2021). AD therefore can be considered as a suitable technique to be employed in improvement ML models for prediction of Covid-19.

In our study, we propose to develop a methodology that applies AD as an integral step in the training of ML models to predict Covid-19 early on basis a symptomatic dataset. As part of

the research, we will study the impact of AD on a categorical Covid-19 dataset and how it affects the predictive power of the trained models. Given that detection and containment of Covid-19 remains an important strategy, ML models with high accuracy could provide the necessary advantage for frontline healthcare workers to act early on.

2. Related Works

Much research has been published that involves Data Science and Machine Learning approaches in the detection and mitigation of Covid-19. A review of the AI/ML methods being used in the detection of Covid-19 found that over 50% of studies chose Random Forest (RF) as at least one of their ML classifiers because of its capacity to pick good features for classification. Support Vector Machines (SVM) was another ML method that was found popular in Covid-19 studies for similar reasons (Dogan et al., 2021). In an endeavour to speed up decision making on treatments and isolation strategies, machine learning classification algorithms, specifically RF ,Gradient Boosting Machine (GBM), XGBoost, SVM and Decision Tree (DT) were deployed to identify which symptoms of a patient were significant predictors of Covid-19 (such as fever, cough, runny nose) and also to understand which among the models were suitable to consistent predictions across age brackets (Ahamad et al., 2020). Another study proposed use of ML algorithms to help with prioritization and targeted use of RT-PCR tests to identify Covid-19. The study suggested use of ML to predict the risks of a positive diagnosis using data from routine tests in situations where the RT-PCR test results took too long. In this case too, RF and SVM featured along with others like Logistic Regression (LR), GBM, and Neural Networks (Afm et al., 2020).

Because of the prevalence of medical image data such as X-rays, CT scans and MRI, DL techniques such as CNN models have been a popular choice either by themselves or together with other methods like AlexNet and LSTM. One of them was a combined DL system that was tested on an X-ray dataset which included samples of Covid-19 patients among others. The system consisted of a CNN network and LSTM, the former of which was used to extract features from the X-ray images and the latter to make predictions based on said features and was able to achieve a significant accuracy of 99.4% (Islam et al., 2020). Another proposed solution was COVIDetectioNet, a novel system that used a three-stage process. A pretrained AlexNet architecture was employed as a feature extractor followed by the use of the Relief

algorithm in identifying the most efficient of these derived features. Finally, the SVM method was used to classify these features (Turkoglu, 2021). A detailed study of several traditional ML models (such as SVM, KNN, RF, DT) as well as DL models (such as GoogleNet, ResNet, Xception) was conducted to understand automatic identification of Covid-19 using X-ray images. The novelty here was a unique process that was used to select the ideal transfer learning model which was then employed to extract features and train both ML and DL models (Mohammed et al., 2021). But none of the research discussed so far have considered using Anomaly Detection (AD) methods to improve model predictions.

In the area of machine learning, at a high level, one way to look at AD techniques is to categorize them based on the traits of the data - into supervised, semi-supervised and unsupervised.

In supervised AD, all instances of the dataset are initially labelled to indicate whether they are outliers or not. The dataset is then divided into train and test followed by the classification models being trained to predict the labels on the test set. SVM and Bayesian Networks are a couple of models used in such cases. SVM was employed as part of an AD method in wireless sensor networks where the model was able to classify node data as a local outlier, network outlier or cluster outlier (Mohamed and Kavitha, 2011). Maritime security was another interesting domain where supervised AD was found useful. With the help of data available via Automated Identification System, Bayesian Networks could detect deviations from regular movement patterns of vessels and thereby assist in identification of security risks or illegal trafficking (Mascaro et al., 2014). The disadvantage in the supervised approach is they require labelled instances which is a manual process, is often scarcely available and is expensive to make. Furthermore, identifying new patterns of outliers becomes difficult when the existing models are trained only on the known ones.

Semi-supervised AD typically involves the use of training datasets where (i) both regular samples as well as outliers are made available with labels along with unlabelled data or (ii) normal data is exclusively made available. The model learns from the labelled data and is then expected to find deviating instances from an unlabelled training set. An example of this is a

hybrid model consisting of a Deep AutoEncoder and an ensemble of KNN based outlier detectors that was proposed in which the former would transform a high dimensional dataset into a compact form post which the latter handled the task of detecting outliers (Song et al., 2017). The hybrid setup provided the advantage of using only a portion of training set due to its compact nature of distribution post transformation as well as helped reduce computational costs.

However, Unsupervised learning has been the leading approach to AD. This is because in typical circumstances the availability of outlier data with indicative labels are scarce. In the event that labelled data is actually available, they are hardly sufficient to cover different characteristics of outliers thus limiting the effectiveness of the supervised approach (Ruff et al., 2021). An example would be the use of Isolation Forest employed as an AD solution in the Information Security domain. In one such scenario, an anomalous user detection system was devised by developing a baseline model of an enterprise's user activity by using Isolation Forest to calculate a threshold anomaly score. When a new user enters the system, a score is calculated and compared with the threshold score, which if crossed results in the user being flagged (Sun et al., 2016). K-Means clustering was used to improve heart disease prediction by employing it as an AD solution. This was proven to be effective by comparing accuracies of various classification models - namely SVM, Naïve Bayes (NB), LR, RF and KNN – prior to and post removal of outliers tagged by K-Means (Ripan et al., 2021).

What is notable about all the AD solutions discussed so far is that they are applicable only in cases where data is quantitative. Categorical data has generally received relatively less consideration. This is as a result of AD in categorical data being a challenge due to a number of reasons. Many AD techniques depend on distance functions to identify anomalous observations that deviate from a representative pattern of normal data. But neither is identifying such patterns nor measuring distance in categorical data a straightforward task. Furthermore, varying definitions of outliers in categorical data exist in the literature. Depending on the definition adopted, different AD methods may end up choosing different instances as outliers (Taha and Hadi, 2019). Fortunately, despite these challenges there has been relatively recent progress in research on viable options to execute AD in categorical data. Frequency-Based, Density-Based, Distance-Based and Clustering-Based AD methods

are some of them. An example would be the Ranking-based Outlier Analysis and Detection method (ROAD) which is a two phased clustering algorithm. The first step involves computing the density of datapoints/objects as well exploring the clustering of the dataset. The second step calculates a frequency-based rank as well as a clustering-based rank of each datapoint. The two separate rankings are then used to then determine the outliers (Suri et al., 2013).

Even though Covid-19 is relatively very recent in terms of research interest, AD has already found its way to being employed in proposed ML/DL Covid-19 detection solutions, albeit in a seemingly limited amount. A DL model consisting of three components, specifically a backbone component based on an ImageNet dataset trained CNN, a classification component, and an AD component was suggested as a solution to detect Covid-19 using X-ray images. Due to limited availability of Covid-19 images and requirement for more data in DL, a mix of pneumonia confirmed images are also included. The backbone extracts the necessary features, provided as input in classification and AD components, both of which then generate classification and anomaly scores. Statistically significant scores are assigned to Covid-19 X-ray images to optimize the model to identify Covid-19. This solution too was suggested as an alternative in circumstances where availability of RT-PCR tests is limited (Zhang et al., 2020). A wearable device was proposed as an option to detect Covid-19 early on by measuring physiological features such as body temperature, physical activity, breathing and cough patterns. Both K-Means and Isolation Forest techniques were utilized as AD based symptom identification. The outliers would point to patients with deviating patterns that require a closer look (Ali et al., 2021).

Most of the ML/DL methods used in the detection of Covid-19 discussed so far did not consider the impact of AD. Those that did applied AD on quantitative data that was usually image-based data like X-ray, CT scans and MRI. To the best of our knowledge, impact of AD in a symptomatic structured data, particularly a Covid-19 dataset that is categorical in nature is yet to be explored.

3. Aim & Objectives

This research is aimed at understanding the impact of Anomaly Detection on the prediction of Covid-19 using symptomatic data. The goal is to investigate the effect that high quality data,

that is data without outliers or anomalies, has on building effective models using machine learning techniques to predict Covid-19 at an early stage.

The objectives of the research are stated as follows:

- To suggest an Anomaly Detection technique that is suitable for application on the Covid-19 dataset that is categorical in nature.
- To evaluate the impact of the Anomaly Detection technique on predictive power of different machine learning models.
- To improve the prediction of Covid-19 at an early stage using the symptomatic data.

4. Significance of the Study

Treatment of Covid-19 is at best experimental and vaccine rollouts are still at a very early stage. As a result, early detection and containment of Covid-19 spread remains one of the most crucial strategies available to stave off its worst impacts on healthcare at an individual as well as infrastructure level. Machine Learning based detection of Covid-19 at an early stage plays an important role in effective prognosis and decision making for healthcare workers. ML models improved by Anomaly Detection could only serve to improve this process further.

5. Scope of the Study

The dataset includes symptomatic data from a few provinces in China. The data as a result is not representative of the Covid-19 symptoms in other parts of the world where different virus mutations, social circumstances and differences in healthcare systems could result in variations in symptomatic data. The dataset is predominantly categorical in nature and therefore the Anomaly Detection method chosen may not work for quantitative data. The study is intended to assist and complement the diagnostic capabilities of the healthcare system and its frontline workers only for Covid-19. The process can inspire or be replicated but cannot be deployed as is for detection and mitigation of other diseases.

5.1 Deliverables

This section will describe the deliverables that are necessary to achieve the stated objectives of this study.

6. Explore various AD techniques for categorical data.
7. Feasibility and implementation test of the AD technique on our dataset.
8. Exploratory Data Analysis to profile the Covid-19 dataset.
9. Compare differences in prediction power of models prior to and post removal of anomalies.
10. Train predictive models that have higher predictive scores on symptomatic Covid-19 data

6. Research Methodology

6.1 Introduction

The methodology section presents details on how we propose to evaluate the impact of AD on symptomatic data for Covid-19. At a high level, we start with data preparation to handle any missing values or data imbalance. This will be followed by exploratory data analysis to further understand the nature and quirks of the available data. An appropriate AD method for categorical data is then applied to label instances as normal or as outliers. Classification models of our choice are then evaluated pre and post removal of the labelled outliers to understand the impact of AD on predictive power. A diagram demonstrating the process is shown in Figure 1.

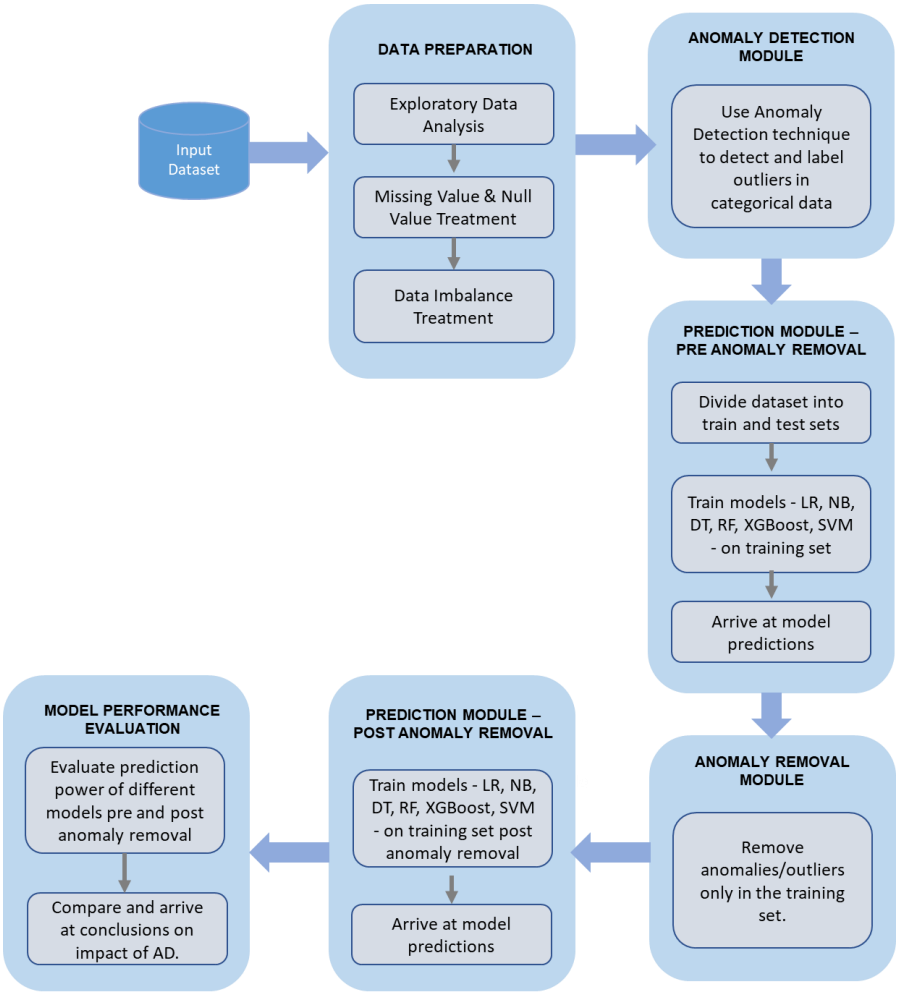

Figure 1: Framework of Anomaly detection based model development

6.2 Data Description

The dataset has been made available by Big Data High-accuracy Center, Beijing University on their Github page as well as on Kaggle. The anonymized dataset consisting of 6,512 patients from seven different provinces of China and was collected by the university from official channels of the Chinese national government's websites. Attributes include basic information (gender, age), symptoms (muscle soreness , cough ,fever, runny nose), symptoms that resulted in hospital admission (diarrhoea , pneumonia), results from diagnostics (radiographic imaging ,lung infection) and other information considered relevant (isolation treatment status, travel history). A target variable is also present which confirms if the patient has Covid-19 (1,572 cases) or not (4,970 cases). The dataset is categorical in nature, with exception to the age attribute. Table 1 below provides further necessary details on the dataset.

Table 1: Data Description

Attribute	Data Type	Description
Fever	Boolean	Symptomatic with body temperature is greater than 38 degrees.
Pneumonia	Boolean	Symptomatic for Pneumonia resulting in hospital admission
Runny Nose	Boolean	Symptomatic for Runny Nose
Lung Infection	Boolean	CT Scan or Radiography images indicate lung infection
Cough	Boolean	Symptomatic for dry cough or cough with sputum
Diarrhoea	Boolean	Symptomatic for Diarrhoea resulting in hospital admission
Muscle Soreness	Boolean	Symptomatic for limb pain or soreness of muscle
Isolation	Boolean	Status of isolation treatment
Travel History	Boolean	Travelled to one or more places within China or abroad
Age	Integer	Age within range of 1-96
Gender	String	Indicates whether patient is Male or Female.
SARS-CoV-2 Positive	Boolean	Confirms whether the patient is Covid-19 positive or not

6.3 Data Preparation Module

Datasets generally consist of instances that can be attributed to noise as well as a high number of features, missing values, null values, and inconsistencies likely due to the manner in which the data was sourced. Exploratory Data Analysis (EDA) would be required to understand the intricacies of the data.

6.3.1 Exploratory Data Analysis

EDA is necessary to further inspect and understand the data. This stage would include:

7. Data inspection to get an overview of the dataset.
8. Univariate and/or Bivariate analysis of the available features.
9. Where necessary, derive variables and/or modify existing ones for better interpretation.
10. Identify variables that are skewed for further action.
11. Use of data visualization packages to assist in the above steps.

6.3.2 Missing Value Treatment

Our dataset does not have any missing values and therefore will not require any specific treatment.

6.3.3 Null Value Treatment

Our dataset does not have any null values and therefore will not require any specific treatment.

6.3.4 Data Imbalance Treatment

The number of Covid-19 positive cases in our dataset is 1,572 while that negative is 4,970. That is about a ratio of 1:3 with the positive cases being the minority class. Imbalanced data can be challenging as machine learning models and methods of evaluation assume that the dataset has a balanced distribution (Brownlee, 2020). Treatment can be carried out using some of the common methods available such as Synthetic Minority Class Oversampling Technique (SMOTE) or Weight of Class. The former creates synthetic datapoints to artificially balance out the minority class while the latter penalizes the model when it predicts the outcome of a minority class incorrectly.

6.4 Anomaly Detection Module

The AD technique for categorical data is yet to be decided but a preliminary review of the available techniques informs us that there are several choices, each of which define what an anomaly is in a unique manner. Given that one of the objectives of the research is to suggest an appropriate AD method, we will not be able to get into the specifics at this juncture. What is expected is that the technique is likely to fall into one of the categories discussed earlier in the Related Works. Frequency-Based, Density-Based, Distance-Based and Clustering-Based AD methods are some of the know types available for categorical data. The ROAD technique, discussed earlier in the Clustering Based AD methods is one of the candidates under consideration. Principle Component Analysis (PCA) is another option with a history of being used for AD but normally for quantitative data (Flowick, 2018). The feasibility to apply PCA as part of an AD technique in our categorical dataset is still being explored. In this module, the outliers will be labelled so that they can be removed at a later point.

6.5 Prediction Module

In this module, we train several ML classification models on the Covid-19 dataset to predict Covid-19 at an early stage. The dataset will be divided into a train and test set post which the models will be trained on the former and their performance tested on the latter. The models are listed and described below.

6.5.1 Logistic Regression (LR)

Logistic Regression (LR) is a ML model based on a technique from the field of statistics. It is a popular method for classification problems that are binary in nature. In this method, each instance is modelled for its probability of falling into a certain class. A statistics-based function, namely the Sigmoid function, is used to arrive at the probability, ranging from 0 to 1 (Cramer, 2003).

6.5.2 Naïve Bayes (NB)

Naïve Bayes (NB) is a probabilistic machine learning technique and is popular for classification tasks. The workings of the algorithm is based on the Bayes' probability theorem. There are a few versions of this algorithm depending on the type of data available. In our case, we will use the Bernoulli Naïve Bayes algorithm which is suitable for a dataset that is predominantly Boolean in nature (Sharma, 2021).

6.5.3 Decision Tree (DT)

Decision Tree (DT) models are utilized for both regression and classification. The model is based on a tree representation where the leaf node is associated with features and branches are associated with values. DT is based on the idea that a tree can be built using the data available and at every leaf, a unique output is represented. A statistical concept called information gain (IG) is used to measure how well a feature is able to assist in classifying the data. A feature present at a node with high IG can effectively split the dataset thereby assisting in improving accuracy of classification.

6.5.4 Random Forest (RF)

The Random Forest (RF) (Breiman, 2001) is a classification technique based on the DT model. It is an ensemble of decision trees, each of which are trained on a distinct subset of the original dataset. Each of these trees make their own prediction on the classification of samples in the test set. The final classification is decided based on prediction from multiple sets of trees through a mechanism that is based on majority voting called Bagging. This results in the overall model being more reliable as the predictions made by an ensemble of trees will likely be more consistent than that made by individual DTs.

6.5.5 Extreme Gradient Boosting or XGBoost

XGBoost is another DT ensemble-based technique, like RF (Chen and Guestrin, 2016). The difference is that it is based on a mechanism called boosting where DT models are built sequentially with each model learning from the mistakes of the previous one. Models which perform well have more influence on the final outcome or are 'boosted'. Additionally, the technique also uses Gradient Descent to reduce errors in the sequentially created models.

6.5.6 Support Vector Machine

Support Vector Machine (SVM) is a very popular ML model, primarily due to the fact that it is a high performance technique with relatively little tuning required (Cortes et al., 1995). SVM classifies data by plotting each instance in k-dimensions where k is the total attributes in our dataset. Each datapoint is plotted as a coordinate based on the values of its corresponding attributes. The SVM then identifies the hyperplane that can separate the two classes, thus enabling the classification of the datapoints.

6.6 Anomaly Removal Module

Post initial training of the above listed models on the original dataset, in this module we will remove the labelled outliers only for the training set. It is expected that since outliers do not conform to the typical aspects of the dataset, they are likely to impact the prediction performance of the models. Given this reason, we will once again train the models on the dataset post removal of labelled outliers.

6.7 Model Performance Evaluation

To quantify the performance of our models in prediction of Covid-19, we will use well known metrics in the ML field. The performance evaluation will be based on the model's prediction on the test dataset.

6.7.1 Precision

Precision helps us understand how many of the positive classes predicted by the model are in fact positive. This metric is helpful when the cost of a false positive is high. The formula is given below:

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)}$$

6.7.2 Recall

Recall is important when the cost of a false negative is high. It tells us how many of the actual positives the model was able to predict correctly. In domains such as healthcare, this is a crucial metric since the cost of not taking action due to a false negative could be high. Below is the formula:

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)}$$

6.7.3 Accuracy

This metric informs us of the model's general performance and if it is being trained correctly overall. Accuracy tells us the overall ratio of the correctly predicted to the total observations. However, it will not be able to get into any specifics of the model's performance like how Precision & Recall does. The formula is as follows:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

6.7.4 F1 Score

F1 score is a balance between Precision and Recall. It takes the weighted average between the two metrics. The score ranges between 0 and 1 with higher being better. While not as intuitive as accuracy, it is more useful in cases where the class distribution is uneven. It is a more useful metric than accuracy when the cost of fall positives and false negatives are uneven. The formula is described below:

$$F1\ Score = \frac{2 * (Recall * Precision)}{(Recall + Precision)}$$

7. Required Resources

For implementation of the different techniques described so far, on the software end we will use Python v3 on Windows 10. Python v3 access and implementation will be carried out

using the latest versions of Jupyter Notebooks and Anaconda3. Certain python packages available via Anaconda3 will also be required, namely scikit-learn, numpy, pandas, matplotlib, seaborn, statsmodels, pydotplus and graphivz.

On the hardware end, we will be use a Intel(R) Core(TM) i5-8250U processor with 8 GB RAM.

8. Research Plan

The below Gantt Chart details the different activities/milestones of our study, those completed as well as those planned until the completion of the thesis. Along with activity description, the chart also indicates the planned vs actual start week, the planned vs actual start duration and the current completion status in percentages.