APPLICATION OF SEVERAL MACHINE LEARNING ALGORITHMS FOR MULTIPLE STAGE INFERENCE DATA

by

KHALID A AMEN

A dissertation submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN SYSTEMS ENGINEERING

2024

Oakland University
Rochester, Michigan

Doctoral Advisory Committee:

Mohamed A. Zohdy, Ph.D., Chair
Julian Rrushi, Ph.D.
Gary McDonald, Ph.D.
Mohammed Mahmoud, Ph.D.

الحمد و الشكر لله لما وفقني في انجاز هذا العمل و له المنة اولا و اخرا

I dedicate the final product of this dissertation to my family who has supported and inspired me throughout this journey.

ACKNOWLEDGMENTS

This dissertation and the research behind it would not have been possible without the grace, the bounty, and the blessing of almighty Allah (God) first and foremost and the exceptional mentoring, guidance, support, and encouragement of my Professors, Mohamed Zohdy, Julian Rrushi, Gary McDonald, and Mohammed Mahmoud. This dissertation bears the first step towards a new beginning for a new stage to come. My extended thanks and appreciation to the department of Electrical and Computer Engineering and Computer Science and Engineering for help and assistance to form my committee and provided guidance along the journey of my research. Finally, I would like to thank my wife for standing with me, support, encouragement, and inspiration to begin and continue this journey, and it wouldn't have been possible without her.

Khalid A Amen

ABSTRACT

APPLICATION OF MACHINE LEARNING FOR MULTIPLE STAGE INFERENCE DATA

by

KHALID A AMEN

Adviser: Mohamed Zohdy, Ph.D.

Historically, machine learning techniques have been dependent on utilizing data from two distinct phases to predict and identify particular occurrences. The outcomes of these studies may exhibit either validity or inaccuracy, represented by binary values of one or zero. An alternative term for this is a prognostication of one of two potential results. Several issues are present in this approach, which have the potential to yield inaccurate outcomes. The issues encompassed in this context consist of data imbalance, overfitting, and error propagation. This study aims to employ and use a multiple stage outcome approach to enhance accuracy and optimize the performance of outcomes. In this step of our research, we will be implementing the Multiclass Classification One-vs.-All methodology to analyze the data collected from various stages of the experiment's conclusion. In the subsequent phase, it is necessary to engage in the utilization or investigation of a diverse range of potential supervised models, which are trained through the application of machine learning algorithms. Subsequently, the determination of the model that exhibits a superior level of accuracy will be made by designating it as the

victor. In our study, we employ and evaluate five distinct Machine Learning algorithms, namely Support Vector Machines (SVM), Logistic Regression (LR), Random Forest (RF), Gradient Tree Boosting (GTB), and Extremely Randomized Trees (ERF). These algorithms are used within our Machine Learning framework to analyze multi-stage data and ascertain the technique that exhibits the highest accuracy in predicting outcome stages. This multi-stage conclusion would effectively narrow down the problem or difficulties at hand, reduce the potential for errors, and enhance the ability to accurately predict and diagnose medical diseases or cyber security threats. A Python-based model was developed to execute the proposed methodology. The utilized notion employs a binary format, which has been substantiated by empirical evidence and offers two potential outcomes. Upon the completion of our research, it was determined that the Logistic Regression and Support Vector Machine algorithms exhibited better performance compared to the other algorithms when a multiple stage outcome was employed. The results were assessed in terms of accuracy, precision, recall, and the F measure.

TABLE OF CONTENTS

TABLE OF CONTENTS - Continued

TABLE OF CONTENTS - Continued

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

ML	Machine Learning
CNN	Convolutional Neural Network
LR	Logistic Regression
SVM	Support Vector Machine
GTB	Gradient Tree Boosting
GBM	Gradient Tree Boosting
RF	Random Forest
ERF	Extra Random Forest
AI	Artificial Intelligence
OvA	One-vs-All
OvR	One-vs-Rest
OvO	One-vs-One
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
C	Classifier
K	Class
k	label
X	Samples
y	outcome
CM	Confusion Matrix
TP	True Positive

TN	True Negative
FP	False Positive
FN	False Negative
P	Output
b0	Intercept
b1	Coefficient
DT	Decision Tree
EF	Ejection Fraction
rEF	reduced Ejection Fraction
UCI	University of California Irvin
URL	Uniform Resource Locator
RSA	Rivest-Shamir-Adleman
SLD	Second Level Domain
TLD	Top-Level Domain
DNS	Domain Name Service
HTTPS	Hypertext Transfer Protocol Secure
SFH	Server Form Handler
JS	JavaScript
IoT	The Internet of Things

CHAPTER ONE

INTRODUCTION

Machine learning, a specialized domain within the realm of artificial intelligence (AI), is dedicated to the advancement of algorithms and models that enable computers to perform tasks without explicit programming. In contrast, the computer acquires knowledge by the assimilation of factual information provided to it. Throughout the procedure, information is entered into an algorithm, which afterwards employs statistical methodologies to facilitate the machine's progressive enhancement of its performance [1].

1.1 Significance of Machine Learning

Machine learning is the term used to describe the process of instructing a computer system to generate accurate predictions by utilizing provided data. This is achieved by the utilization of diverse methodologies and neural network models within computer systems, facilitating a steady enhancement of their overall performance [1]. Machine Learning algorithms autonomously generate a mathematical model by leveraging sample data, also known as "training data." This model is thereafter employed to render assessments without undergoing explicit training to arrive at such inferences [1, 2], [14]. These predictions encompass various tasks such as classifying a fruit in an image as either a banana or an apple, detecting pedestrians crossing the road in front of an autonomous vehicle, disambiguating the meaning of the word "book" in a sentence as either a physical copy or a hotel reservation, discerning the legitimacy of an email as spam or not, and accurately transcribing speech to generate captions for a YouTube video [3]. Machine learning is an area of artificial intelligence (AI) and computer science that

focuses on the application of data and algorithms to imitate the method in which humans learn, with the goal of steadily improving the accuracy of the simulation [4] [5].

The field of data science, which is experiencing fast growth, encompasses a crucial subfield referred to as machine learning. In the realm of data mining endeavors, statistical methodologies are employed to instruct algorithms in generating classifications or predictions, so facilitating the identification of pivotal insights [1, 4]. The insights obtained from these findings have the potential to significantly impact important growth indicators, hence shaping subsequent actions made inside applications and companies. The anticipated rise in the demand for data scientists is attributed to the ongoing expansion and growth of big data. The data scientists will be tasked with aiding in the identification of the most relevant business inquiries and, subsequently, the selection of the data necessary to address those inquiries [2].

How Machine Learning Works

The learning system of an algorithm for machine learning at the University of California, Berkeley is categorized into three main sections [6][7][8].

1. **A Decision Process**: In the majority of instances, the utilization of Machine Learning algorithms is driven by the objective of generating predictions or classifications. The algorithm will construct an estimation of a pattern within the data, utilizing input data that may or may not possess labels.
2. **An Error Function**: The assessment of the model's prediction is achieved by employing an error function. In instances where pre-existing instances are available, an error function can be employed to facilitate a comparison and ascertain the level of accuracy exhibited by the model.

3. **A Model Optimization Process**: If the model can be improved to achieve a better match with the data points in the training set, the weights will be adjusted in order to minimize the discrepancy between the observed example and the estimated output generated by the model. The algorithm will iterate through the process of evaluation and optimization, automatically updating weights until a preset level of accuracy is achieved.

Machine Learning Methods

- Supervised Machine Learning

  Supervised learning, often known as supervised machine learning, is defined by the use of labeled datasets to train algorithms that reliably categorize data or predict outcomes [38]. Supervised learning, alternatively referred to as supervised machine learning, is a widely recognized term in the field. The model will iteratively update its weights until it has achieved a satisfactory fit to the input data. In the context of cross-validation, this particular step is implemented to ensure that the model does not encounter issues of overfitting or underfitting. Supervised learning enables businesses to address a diverse range of real-world challenges on a large scale, such as the effective categorization of spam emails into a separate folder from other incoming communications. Supervised learning encompasses a variety of methodologies, including neural networks, naive Bayes, linear regression, logistic regression, random forest, support vector machine (SVM), and additional techniques [9][10][14].

- Unsupervised Machine Learning

Unsupervised learning, commonly referred to as unsupervised machine learning, is an approach to learning that leverages machine learning algorithms to assess and group datasets that lack labeled information. These algorithms find previously unknown patterns or data groupings without requiring any assistance from a human researcher. Due to its capacity to identify similarities and differences in data, it emerges as the optimal solution for exploratory data analysis, cross-selling strategies, customer segmentation, image identification, and pattern recognition. Principal component analysis (PCA) and singular value decomposition (SVD) are two basic approaches that are used for dimensionality reduction. Additionally, it is employed to reduce the number of features in a model by means of dimensionality reduction. Unsupervised learning can employ several methods, such as neural networks, k-means clustering, and probabilistic clustering methodologies, among others [9][10][14].

- Semi-supervised learning

  Learning in a semi-supervised setting offers a favorable equilibrium between learning in a supervised context, where labeled data is readily available, and learning in an unsupervised context, where no labeled data is provided. During the training phase, a smaller dataset that has been labeled is utilized for the purposes of guiding classification and feature extraction from a larger, unlabeled dataset. The issue of insufficient labeled data, whether due to limited resources or cost constraints, can be effectively addressed by employing semi-supervised learning as an alternative to traditional supervised learning [9][10][11][14].

1.2 Why Multiple Stage

The utilization of machine learning in processes or outcomes that include multiple stage outcome has the potential to enhance efficiency. This can lead to improved performance, enhanced decision-making capabilities, and increased system resilience [12][13][14]. Several reasons lead to the requirement of multiple processing or outcome steps in machine learning:

1. **Cleaner Data Processing**: In the context of data preparation, it is imperative to partition the procedure into several stages to facilitate the systematic cleaning, conversion, and organization of the data. At each hierarchical level, the focus of attention may transition to a distinct activity, such as the process of encoding, normalization, or imputation of missing information.
2. **Modularity**: A modular pipeline can be constructed by organizing the data into stages and facilitating its movement along the pipeline, allowing each phase to operate as an autonomous unit. This simplifies the process of debugging, changing, and validating each level of the data pipeline.
3. **Scalability**: Simultaneously processing all elements might result in significant resource use, particularly when handling large datasets. Processing can be executed either in batches or in a distributed manner due to the presence of many phases, hence facilitating scalability.
4. **Feature Engineering**: The process has several sequential stages, facilitating the iterative implementation of feature engineering. The extraction and analysis of the initial features can be conducted, and subsequent procedures can be employed to enhance or broaden the scope of these features.

5. **Efficient Resource Use**: There is a potential for enhanced computing resource deployment by adopting a phased approach to data processing, wherein each stage is tailored to maximize performance.
6. **Flexibility**: When data processing is divided into several phases, the implementation of new procedures becomes significantly more straightforward. For example, in the event that a novel data source becomes available or if a preprocessing step necessitates revision, the relevant stage can be adjusted without the need for a whole reconfiguration of the pipeline.
7. **Reduction of Complexity**: The decomposition of a complex data processing task into smaller, sequential operations becomes feasible through the incorporation of numerous phases of data handling. The act of simplifying has the capacity to facilitate the decrease in errors and enhance the overall efficiency of the operation.
8. **Improved Model Performance**: Systematic and comprehensive data preparation has the capacity to yield data of superior quality, thereby enhancing the performance of machine learning models.
9. **Audit and Validation**: When using a methodology comprising many steps, it is possible to incorporate audit checkpoints following each stage to verify the expected execution of data processing or transformation.
10. **Version Control**: The inclusion of many phases in the data pipeline has several advantages, one of which is the enhancement of version control capabilities. The process of tracking and documenting modifications made at a particular stage can be accomplished more easily.

1.3 Performance Assessment of Multiple Stage Data Predication

Assessing the performance of a single model is frequently less complex compared to evaluating the performance of a machine learning prediction that involves multiple phases. In contrast, it is imperative to evaluate the effectiveness of multi-stage forecasts in machine learning or other forecasting methodologies to ascertain the efficacy of each stage in contributing to the ultimate prediction and to prevent the introduction of undesired complexities or inaccuracies. There exist a variety of approaches for assessing the performance of employees [15][16][17]:

1. **Define Clear Objectives**: Acquire knowledge and set the objectives that ought to be achieved at every phase of the prediction procedure. Is it a preliminary estimation that will undergo further refinement in subsequent phases? Does the act of preparing the data involve getting it ready for the subsequent stage in the process?
2. **Individual Stage Evaluation**: The performance of each prediction stage should be assessed individually using suitable metrics, such as the F1 score for classification tasks. This analysis aims to evaluate the extent to which each stage successfully accomplishes its specific objectives, as well as the level of accuracy or error rate associated with each stage.
3. **End-to-End Performance**: Consider the complete pipeline, which includes multiple phases of prediction, and assess its overall performance. This will offer a detailed viewpoint on the process of forecasting.

4. **Error Propagation**: This analysis focuses on the impact of errors or ambiguities in a particular stage on subsequent stages and their outcomes. Determine whether the severity of faults increases as they advance through the phases.
5. **Efficiency and Latency**: If there is a need for real-time prediction, it is advisable to perform an analysis on the duration it takes for an input to traverse through all the stages and generate a prediction.
6. **Consistency and Robustness**: Ensure that the predictions are consistently correct across all of the processes. Are there several varieties, would you argue? Evaluate the system's ability to handle atypical scenarios and unanticipated inputs.
7. **Comparison with Single-Stage Prediction**: The efficacy of a multi-stage prediction can be meaningfully evaluated in relation to the effectiveness of a single-stage forecast. This aids in evaluating if the implementation of a multistage plan provides significant benefits compared to a single stage method.
8. **Stakeholder Feedback**: The collection of input from stakeholders can yield valuable insights into practical performance and potential areas for enhancement, particularly in contexts where the accuracy of forecasts directly impacts users or business decision-making. This phenomenon holds particular significance in contexts where the prognostications have a direct influence on end-users.

CHAPTER TWO

RESEARCH OBJECTIVE

The main objective of this dissertation is to implement a Machine Learning method for the analysis of structured data across multiple stage outcome data. Due to our methodology, we possess the capability to predict multiple stage outcome data using Multiclass Classification One-vs-All approach. To ascertain the most precise machine learning method for forecasting outcome stages, we proceed by implementing and employing five distinct machine learning algorithms (Support Vector Machines, Logistic Regression, Random Forest, Gradient Boosting Trees, and Extremely Randomized Trees) within our proprietary machine learning framework for multiple stage outcome data. Multiple stage outcome of result can significantly enhance the accuracy of predicting and detecting medical diseases or cyber security risks, while simultaneously reducing the margin of error and facilitating the identification and resolution of specific problems or difficulties. When endeavouring to forecast the probability of a particular outcome, the result generated by an algorithm that has undergone training using a dataset and afterwards applied to novel data is referred to as the prediction. The result obtained from a computational model is commonly referred to as the "prediction." The subsequent sections encompass the domains of focus and challenges:

- What is the recommended approach for using our Machine Learning methodology on data sets that involve multiple stage outcome?
- Which algorithm classifier is the most effective for our implementation?

- This inquiry pertains to the effective use of a multi-stage data collection methodology for the purposes of diagnosing and predicting medical diseases, as well as identifying cyber security dangers.

CHAPTER THREE

MOTIVATION

Regrettably, the healthcare industry and Cyber Security sector accumulate substantial quantities of data; yet, the process of extracting previously undiscovered information for the purpose of making precise forecasts is not currently being employed. The failure to profit on the discovery of previously unknown patterns and relationships is a common occurrence. Advanced machine learning techniques can be utilized to offer a resolution to this issue. Machine Learning algorithms have the capability to predict not only the occurrence of an event, but also its potential harm, the presence of malware, the exploitation of vulnerabilities, and other related factors. This is particularly relevant in the domains of health illnesses and cyber security concerns. We possess the capability to extend our predictive capabilities beyond that level to enhance precision. The outcome of a conventional Machine Learning prediction can be categorized as either verifiably accurate or incorrect. In the event if a comprehensive inventory of global mortality factors were to be assembled, it would be plausible to deduce that cardiovascular disease, specifically heart illness or heart failure, emerges as the predominant cause of death. The identification of heart disease or heart failure in a patient poses a challenging task for medical practitioners, necessitating extensive diagnostic tests and years of experience to ascertain the presence of risk factors for these conditions. The inclusion or exclusion of a prognosis regarding heart disease or heart failure does not significantly enhance the diagnostic procedure. In the domain of predicting heart illness or heart failure, there has

been a notable improvement in the precision of forecasting. The current emphasis in heart disease or heart failure research is primarily on the predictive capacity of identifying these conditions in patients through certain indicators or parameters. This prognostication will provide insight into the presence or absence of heart illness or heart failure in the patient. However, it does not offer information regarding the specific stage of disease failure the patient is presently undergoing or the future stages they may potentially reach. The narrowing of the prediction can facilitate the precise diagnosis of the disease, the administration of suitable medicine, the mitigation of disease progression, and the prevention of adverse outcomes. This holds significance within the framework of data breaches and vulnerabilities in the realm of cyber security. If the occurrence of a cyberattack or the presence of malware can be anticipated by the analysis of specific data or variables, it may be feasible to impede, decelerate, or maybe circumvent the cyberattack or infection. In the subsequent chapters, we will systematically examine each research study to demonstrate the comprehensive implementation of multiple stages pertaining to medical illnesses and cyber security concerns. This task will be undertaken in anticipation of the subsequent chapters.

CHAPTER FOUR

THE DESIGN OF THE ALGORITHM

Introduction

In contemporary economies, a significant proportion of companies generate a considerable volume of data, and Machine Learning (ML) serves as a tool to effectively use this data, so enabling enterprises to realize enormous value.

Individuals engaged in data science endeavors must possess a comprehensive comprehension of the fundamental stages essential for accomplishing efficient machine learning, as each project exhibits distinct characteristics and requirements. Only when certain conditions are met will it become possible to effectively identify and develop comprehensive machine learning solutions that can potentially impact business operations.

In the present study, we guide the reader through an adapted iteration of our Machine Learning system. This framework provides a clear and effective structure for any machine learning project that one may be engaged in. The use of a clear and systematic procedure will facilitate the settlement of intricate problems.

The primary aim of the 7 Stages framework is to systematically organize the various activities associated with machine learning by decomposing them into their constituent elements. Ultimately, the framework serves as a versatile technique that can be universally applied to any project, irrespective of the industry or type of company being undertaken.

The 7 Stages of Machine Learning are:

1. Data Collection
2. Data Preparation
3. Choosing ML Model
4. Training the Model
5. Model Evaluation
6. Prediction

The framework of machine learning can be divided into multiple stages, as indicated by various scholarly sources. The framework consisting of seven stages will be utilized as the foundation for our design. If the ML Model were to undergo alteration, the implementation of the modification would occur during stage 3. This particular machine learning modeling technique is commonly employed for the analysis of binary outcomes. The proposed adjustment entails the construction or implementation of many stages that rely on binary outcome data. In order to put our algorithm into reality, we will use the Python programming language [14][18][19].

4.1 Data Collection:

The initial phase in the life cycle of machine learning is commonly referred to as the data acquisition stage. The objective of this stage is to identify and gather all concerns that are linked to the data [14][18].

The identification of diverse data sources, such as files, databases, the internet, and mobile devices, is necessary in this step due to the large range of available data. In terms of the life cycle, this is one of the most essential stages. The efficacy of the output will be directly correlated with both the quantity and the quality of the data that was

gathered. Increasing the amount of data collected will enhance predictive accuracy, enabling a more faithful representation of reality.

This step encompasses all of the tasks enumerated below:

- Identify various data sources.
- Collect data.
- Integrate the data obtained from different sources.

By executing the aforementioned procedures, we can acquire a consolidated assemblage of information, commonly denoted as a dataset. It will be employed in the later stages.

4.2 Data Preparation:

After the conclusion of the data collection process, the subsequent step involves the organization and cleansing of the acquired information. The initial phase referred to as "data preparation" involves the organization and formatting of data in a suitable manner for utilization in Machine Learning training [14][18].

During this step, we are going to begin by getting all of the data together, and then we are going to randomize the ordering of the data. The performance of a machine learning model is significantly influenced by both the characteristics and the quality of the data that is incorporated into the model. During the stage referred to as "Data Preparation," researchers engage in the examination, pre-processing, conditioning, and transformation of the data that will subsequently be utilized for modeling and analysis. Prior to proceeding to the subsequent phase, it is imperative to obtain a comprehensive comprehension of the data, to familiarize oneself with the data, and to acquire knowledge pertaining to the data. The subsequent instances illustrate several procedures undertaken during this particular stage [14][18][19]:

- **Data Exploration**: It is used to understand the nature of data that we have to work with. We need to understand the characteristics, format, and quality of data. A better understanding of data leads to an effective outcome. In this, we find Correlations, general trends, and outliers.
- **Data Wrangling**: Data wrangling is the process of cleaning and converting raw data into a useable format. It is the process of cleaning the data, selecting the variable to use, and transforming the data in a proper format to make it more suitable for analysis in the next step. It is one of the most important steps of the complete process. Cleaning of data is required to address the quality issues. It is not necessary that the data we have collected is always of our use as some of the data may not be useful. In real-world applications, collected data may have various issues, including missing values, duplicate data, invalid data, noise. So, we use various filtering techniques to clean the data. It is mandatory to detect and remove the above issues because it can negatively affect the quality of the outcome.
- **Data Visualization**: Data Visualization is used to perform Exploratory Data Analysis (EDA). When one is dealing with large volumes of data, building graphs is the best way to explore and communicate findings. Visualization is an incredibly helpful tool to identify patterns and trends in data, which leads to clearer understanding and reveals important insights. Data Visualization also helps for faster decision making through graphical illustration. Some of common ways of visualization are area chart, bar chart, Heat Map, Histogram, and Network Diagram.

- **Scaling Data**: To accomplish the multiple stage output data, it is important to scale the data, so the Machine Learning algorithms do not overfit to the wrong features [29] [30]. We use a multiclass classification one-vs-all approach to handle multiple stage out data.

The Multiclass Classification approach is a technique used to categorize entities, wherein the Target variable consists of multiple classes. Each new sample or data point is assigned to only one of these classes. The methodology involves partitioning a dataset containing many classes into distinct sets of binary problems. Subsequently, a binary classifier is trained to handle each individual binary classification model, followed by making predictions utilizing the model that exhibits the highest degree of confidence.

The strategies known as OvO (One-vs-One) and OvA (One-vs-All) are widely recognized approaches for addressing multiclass classification problems by employing algorithms that function as binary classifiers. These procedures are occasionally referred to as One-vs-Rest (OvR) techniques. Both of these strategies decompose the multiclass problem into many binary classification problems, while employing distinct approaches to achieve this.

We decided to go with Multiclass Classification One-vs-All (OvA) approach for the following reasons:

- **Simplicity**: The OvA technique is characterized by a straightforward conceptual framework. N distinct classifiers are trained to classify N distinct classes, with each classifier assigned the task of distinguishing one class from the remaining

classes. The inherent simplicity of this concept typically facilitates its practical application.

- **Scalability with Number of Classes**: In the context of OvA (One-vs-All) classification, it can be observed that the number of classifiers needed exhibits a proportionate relationship with the total number of categories. In contrast to the One-vs-One (OvO) strategy, which necessitates the use of N (N - 1) / 2 classifiers for N classes, this particular approach is often characterized by enhanced scalability.
- **Full Dataset Utilization**: Each classifier in the One-vs-All (OvA) approach is trained using the entire dataset specific to that classifier. This can lead to more precise generalization, especially when the dataset under consideration is not significantly large.
- **Speed and Computation**: Particularly when the quantity of classes expands, the process of training the N classifiers essential for the One-vs-All (OvA) approach can exhibit greater computing efficiency compared to training the N (N – 1) / 2 classifiers mandated by the One-vs-One (OvO) approach. The OvA technique of prediction is considered to be more efficient due to its evaluation of only N classifiers, in contrast to the OvO approach which necessitates the evaluation of N (N – 1) / 2 classifiers.

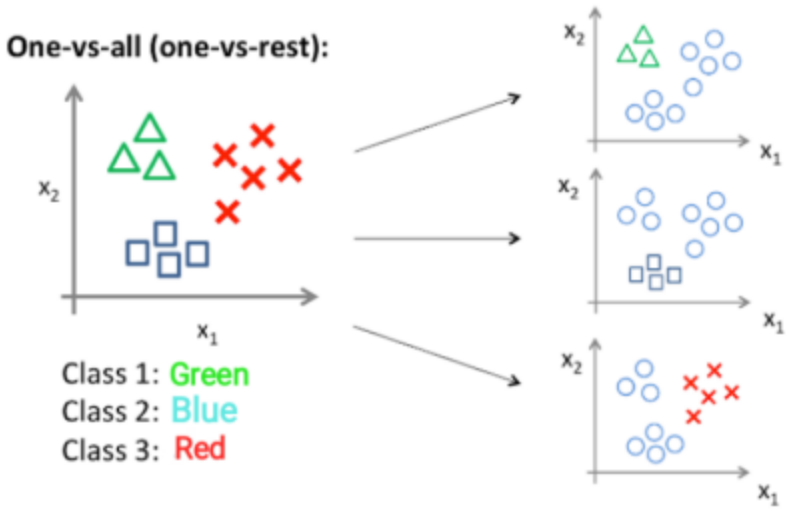

Figure 1: Multiclass classification

When doing multiclass classification, we are required to develop N-binary classifier models for each dataset containing N-class of instances as shown in Figure 1. Both the number of class labels that are included in the dataset as well as the number of binary classifiers that are constructed need to be equal. In the context of a multi-class classification problem including datasets labeled with distinct three colors such as red, green, and blue, binary classification can be subdivided into three subsequent categories:

- The first binary classifier is red against green and blue.
- The second binary classifier is blue against green/red.
- The third binary classifier is green versus blue and red.

As it can be seen in Table 1, there are three class labels Green, Blue, and Red shown in the dataset. Now we need to generate a training dataset for each class:

- Class 1: Green
- Class 2: Blue
- Class 3: Red

Features			Classes
x1	x2	x3	G
x4	x5	x6	B
x7	x8	x9	R
x10	x11	x12	G
x13	x14	x15	B
x16	x17	x18	R

Table 1: Sample of Dataset with multiclass

The scaling would be as follows:

1. From the original dataset, we generate a training dataset1 for class green as shown in Table 2. In the training dataset 1, we update column green to +1 whenever there is a G in the original dataset and -1 for Red and Blue. This is called green vs all.

Features			Classes		Features			Green
x1	x2	x3	G		x1	x2	x3	+1
x4	x5	x6	B		x4	x5	x6	-1
x7	x8	x9	R		x7	x8	x9	-1
x10	x11	x12	G		x10	x11	x12	+1
x13	x14	x15	B		x13	x14	x15	-1
x16	x17	x18	R		x16	x17	x18	-1

Table 2: Main Dataset and Training Dataset / Class Green

2. From the original dataset, we generate a training dataset 2 for class Blue as shown in Table 3. In the training dataset 2, we update column Blue to +1 whenever there is a B in the original dataset and -1 for Red and Green. This is called blue vs all.

Features			Classes		Features			Blue
x1	x2	x3	G		x1	x2	x3	-1
x4	x5	x6	B		x4	x5	x6	+1
x7	x8	x9	R		x7	x8	x9	-1
x10	x11	x12	G		x10	x11	x12	-1
x13	x14	x15	B		x13	x14	x15	+1
x16	x17	x18	R		x16	x17	x18	-1

Table 3: Main Dataset and Training Dataset / Class Blue

3. From the orignal dataset, we generate a training dataset 3 for class Red as shown in Table 4. In the training dataset 3, we update colmun Red to +1 whenever there is a R in the original dataset and -1 for Blue and Green. This is called red vs all.

Features			Classes		Features			Red
x1	x2	x3	G		x1	x2	x3	-1
x4	x5	x6	B		x4	x5	x6	-1
x7	x8	x9	R		x7	x8	x9	+1
x10	x11	x12	G		x10	x11	x12	-1
x13	x14	x15	B		x13	x14	x15	-1
x16	x17	x18	R		x16	x17	x18	+1

Table 4: Main Dataset and Training Dataset / Class Red

Input: Training data with features X and labels Y, where Y can have values from 1 to K (K classes).

Output: A model that predicts class labels for given features.

1. Load the training data: X, Y
2. Preprocess the data:
    - Normalize feature values.
    - Treat the target class as the positive class and the other classes combined as the negative class.
    - Split the data into training and test sets (80% training and 20% test).
3. Choose a classification algorithm (GTB, RF, SVM, ERF, LR).

4. Train the chosen classifier on the training data:
    - For each iteration or epoch:
        - Use the training data to update the model's parameters.
        - Check the model's performance on the validation data.
5. Once training is complete, evaluate the model's performance on the validation set.
6. If performance is satisfactory, proceed; otherwise, iterate over the model's hyperparameters or try a different algorithm.
7. For new data points, use the trained model to predict class labels.

Inputs:

- $C,\ a\ binary\ training\ classifier$
- $Samples\ X$
- $Labels\ y\ where\ y_i\ \in \{1 \dots K\}$

Outputs:

- $A\ collection\ of\ classifiers\ f_k\ for\ k\ \in \{1 \dots K\}$

Procedure:

- $for\ each\ k\ in \{1 \dots K\}$
- $build\ a\ new\ lable\ vector\ v\ where\ v_i = y_i\ if\ y_i = k\ and\ v_i = 0\ otherwise$
- $Apply\ C\ to\ X, v\ to\ obtain\ f_k$
- $X\_train,\ X\_test,\ y\_train, y\_test \leftarrow train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)$
- $SVM(X\_train, y\_train, X\_test, y\_test)$

  $evaluationParameters(X\_train, y\_train, X\_test, y\_test)$
- $LR(X\_train, y\_train, X\_test, y\_test)$

$$evaluationParameters(X\_train, y\_train, X\_test, y\_test)$$

- $RF(X\_train, y\_train, X\_test, y\_test)$

$$evaluationParameters(X\_train, y\_train, X\_test, y\_test)$$

- $GTB(X\_train, y\_train, X\_test, y\_test)$

$$evaluationParameters(X\_train, y\_train, X\_test, y\_test)$$

- $ERF(X\_train, y\_train, X\_test, y\_test)$

$$evaluationParameters(X\_train, y\_train, X\_test, y\_test)$$

Evaluation parameters:

- $Define\ evaluationParameters(X\_train, y\_train, X\_test, y\_test)$:

  $X\_train \leftarrow fit\_transform(X\_train)$

  $Classifier \leftarrow sklearn()$

  $y\_pred \leftarrow classifier.predict(X\_test)$

  $cm\_test \leftarrow confusion\_matrix(y\_pred, y\_test)$

  $y\_pred\_train \leftarrow classifier.predict(X\_train)$

  $cm\_train \leftarrow confusion\_matrix(y\_pred\_train, y\_train)$

  $training\_accuracy \leftarrow (cm\_train[0][0] + cm\_train[1][1])/len(y\_train)$

  $test\_accuracy \leftarrow (cm\_test[0][0] + cm\_test[1][1])/len(y\_test)$

  $training\_percision \leftarrow cm\_train[0][0]/(cm\_train[0][0] + cm\_train[1][0])$

  $test\_percision \leftarrow cm\_test[0][0]/(cm\_test[0][0] + cm\_test[1][0])$

  $training\_recall \leftarrow cm\_train[0][0]/(cm\_train[0][0] + cm\_train[0][1])$

  $test\_recall \leftarrow cm\_test[0][0]/(cm\_test[0][0] + cm\_test[0][1])$

  $training\_f\_measure \leftarrow (2 * training\_percision * training\_recall)/$
  $(training\_percision + training\_recall))$

  $test\_f\_measure \leftarrow (2 * test\_percision * test\_recall)/(test\_percision +$
  $test\_recall))$

$$return\ (training\_accuracy, test\_accuracy,$$
$$training\_percision, test\_percision, training\_recall,$$
$$test\_recall, training\_f\_measure, training\_f\_measure)$$

The method of making judgments involves applying various classifiers to an unobserved sample x and predicting the label k based on the classifier that yields the highest confidence score:

$$y^{\wedge} = argmax\ f_k(x)\ where\ k\ \in \{1 \dots K\}, y^{\wedge}\ new\ dataset$$

4.3 Choosing Model:

The process of choosing the suitable machine learning model for a specific task encompasses both artistic and scientific elements. Several factors influence the decision-making process, including the characteristics of the data, the research question being addressed, the computational resources at hand, and the desired level of interpretability for the model.

The aim of this step is to build a Machine Learning model to analyze the data using various analytical techniques and review the outcome. It starts with the determination of the type of problems, where we select the Machine Learning techniques such as Classification**,** Regression**,** Cluster analysis**,** Association, feature characteristics, etc. Machine Learning algorithms can be divided into supervised and unsupervised learning:

- **Supervised Learning Algorithms** are employed where the training data has output variables corresponding to the input variables. The algorithm analyses the input data and learns a function to map the relationship between the input and output variables. Supervised learning can further be classified into Regression, Classification, Forecasting, and Anomaly Detection.

- **Unsupervised Learning** algorithms are used when the training data does not have a response variable. Such algorithms try to find the intrinsic pattern and hidden structures in the data. Clustering and Dimension Reduction algorithms are types of unsupervised learning algorithms.

Based on the above classification and all of our datasets are numeric, text format, and labeled. We would use supervised learning algorithms for our model design. We chose the most 5 popular algorithms, Gradient Tree Boosting, Random Forest, Support Vector Machine (SVM), Extra Random Forest, and Logistic Regression to predict multiple stage data dataset problem.

4.4 Methodology:

The proposed methodology using five classification techniques; Gradient Tree Boosting, Random Forest, Support Vector Machine (SVM), Extra Random Forest, and Logistic Regression to predict multiple stage data dataset problem as the proposed methodology shown in Figure 2. These classifiers are used to improve prediction. The performance of these classifiers is evaluated on the bases of accuracy, precision recall, and F measure [14][18][19].

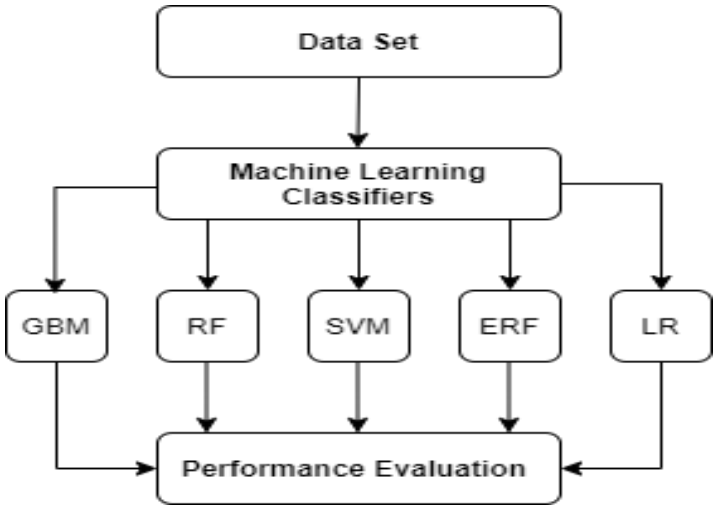

Figure 2: Methodology

4.5 Evaluation Parameters

Some evaluation parameters in data mining are accuracy, precision, recall, and F measure. Where TP True Positive, TN- True Negative, FP- False Positive and FN- False Negative [14][18][19].

- Accuracy is defined as the number of accurately classified instances divided by the total number of instances in the dataset as in equation 1.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

- Precision is the average probability of relevant retrieval as described in equation 2.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

- The recall is defined as the average probability of complete retrieval as defined in equation 3.

$$\text{Recall } = \frac{TP}{TP + FN} \tag{3}$$

- F- Measure is the calculated by using both precision and recall as shown in equation 4.

$$\text{F Measure } = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{4}$$

Where all evaluation parameters accuracy, precision, recall, and F measure are calculated from dataset when splitting the dataset into training data and test data. The Pseudocode for the evaluation parameters are as follow [14][18]:

Algorithm 1: Evaluation Parameters Pseudocode

$Define\ evaluationParameters(X\_train, y\_train, X\_test, y\_test)$:

$X\_train \leftarrow fit\_transform(X\_train)$

$Classifier \leftarrow sklearn()$

$y\_pred \leftarrow classifier.predict(X\_test)$

$cm\_test \leftarrow confusion\_matrix(y\_pred, y\_test)$

$y\_pred\_train \leftarrow classifier.predict(X\_train)$

$cm\_train \leftarrow confusion\_matrix(y\_pred\_train, y\_train)$

$training\_accuracy \leftarrow (cm\_train[0][0] + cm\_train[1][1])/len(y\_train)$

$test\_accuracy \leftarrow (cm\_test[0][0] + cm\_test[1][1])/len(y\_test)$

$training\_percision \leftarrow cm\_train[0][0]/(cm\_train[0][0] + cm\_train[1][0])$

$test\_percision \leftarrow cm\_test[0][0]/(cm\_test[0][0] + cm\_test[1][0])$

$training\_recall \leftarrow cm\_train[0][0]/(cm\_train[0][0] + cm\_train[0][1])$

$$test\_recall \leftarrow cm\_test[0][0]/(cm\_test[0][0] + cm\_test[0][1])$$

$$training\_f\_measure \leftarrow (2 * training\_percision * training\_recall)/ (training\_percision + training\_recall))$$

$$test\_f\_measure \leftarrow (2 * test\_percision * test\_recall)/(test\_percision + test\_recall))$$

$$return\ (training\_accuracy, test\_accuracy, training\_percision, test\_percision, training\_recall, test\_recall, training\_f\_measure, training\_f\_measure)$$

4.6 Training:

Now the next step is to train the model, in this step we train our model to improve its performance for a better outcome of the problem. We use datasets to train the model using various Machine Learning algorithms. Training a model is required so that it can understand the various patterns, rules, and features. We will use 5 Machine Learning algorithms to see which performs better.

The training data delivered from the previous Data stages are utilized within the training stage. The implementation of model training involves passing the refined aggregated training data through the implemented model to create a model that can perform its dedicated task well [14].

The training of the implemented model involves iteratively passing mini batches of the training data through the model for a specified number of epochs. During the early stages of training, model performance and accuracy can be very unimpressive. Still, as the model conducts predictions and a comparison of predicted values is made to the desired/target value, backpropagation takes place within the neural networks, the model begins to improve and gets better at the task it's designed and implemented to do.

For training, there are three ways to train our model:

- Use model. fit() to model for a fixed number of epochs.
- Use model_train_on_batch() to train with a single batch only and once.
- To create a custom training loop

Since our model is used for a fixed number of epochs, we will use model. fit() to train our model [14][18].

Algorithm 2: Model Training

$$X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)$$

$$Model.fit(X\_train, y\_train)$$

4.7 Model Evaluation:

Once training is complete, it's time to see which model or classifier is better than others in terms of accuracy, precision, recall, and F measure. We plot the progress on accuracy, precision, recall, and F measure using some common ways of visualization such as Area Chart, Bar Chart, Heat Map, Histogram, or Network Diagram. Once training is complete, it's time to see if the model is any good, using Evaluation. This is where that dataset that we set aside earlier comes into play. Evaluation allows us to test our model against data that has never been used for training. This metric allows us to see how the model might perform against data that it has not yet seen. This is meant to be representative of how the model might perform in the real world.

A good rule of thumb we use for a training-evaluation is split somewhere on the order of 80/20 or 70/30. Much of this depends on the size of the original source dataset. If we have a lot of data, perhaps we don't need as big of a fraction for the evaluation dataset.

Prediction:

Once our Machine Learning model has been trained on a given dataset, then the final step is to use this model to make predictions or inference. In this step, we check for the accuracy, precision, recall, and F measure of our model by providing a test dataset to it. Testing the model determines the performance of the model as per the requirement of project or problem. We can finally use our model to predict the multiple stage outcome on specific data [14][18][19].

Algorithm 3: Model Prediction

$model.predict(x\_test)$

4.8 Machine Learning Classifiers

In our prediction, five classification methods are implemented and used in our algorithm. These classifiers are compared to find out which of the multiple stages best predicts the outcome. In the next section, we briefly describe these classification techniques/ classifiers.

1. **Logistics Regression (LR)**: is the predictive analysis to conduct on discrete (binary) values based on a specified set of independent variables. LR describes the data and clarifies the relationship between one (binary) dependent variable and independent variables. It predicts the event occurrence probability by fitting the data into a logit function. Therefore, it is also called logit regression [14][18].

Input values x is linearly combined using coefficient values b, to calculate an output value p. The output values as predictable lies between 0 and 1. Input data associated with coefficient b (constant value) learned from training data. Where p is the output, b0 is an intercept term and b1 is the coefficient of input value x as shown in equation 5.

$$P = \frac{e^{(b0+b1+x)}}{1 + e^{(b0+b1+x)}} \quad (5)$$

2. **Support Vectors Machine (SVM)**: is a classification and regression algorithm. In SVM, every data item is plotted in n-dimensional space, several dimensions are equivalent to the number of features or attributes. Where n represents the number of attributes. The value of each attribute being the value of certain coordinates. Once plotting all the data items then performed classification by drawing a line or by finding the optimal hyperplane that separates two classes completely. For example, if we have two features of an individual like hair and height length. First, we plot these two features in two-dimensional space where every point has two coordinates (these co-ordinates are also known as Support Vectors) [14][18].
3. **Random Forests (RF)**: are ensemble learning techniques for regression, regression, classification, and for other tasks. That operate by making a multitude of Decision Tree (DT) at training stint and outputting that is the mean prediction (regression) or mode of classes (classification) of the distinct trees.

Every tree in the forest contributes to a classification. To classify new case based on its attributes. We identify the tree “votes” for that class, so the forest indicates the classification of the case that is taking the most votes [14][18].

Every tree is planted and grown as follows:

- If the number of objects N in training set, the sample of N objects is taken randomly with replacement. This sample acts as a training set for growing tree.
- If there is an input variable N, number n < N is stated that at each node, randomly selection of n variable out of input variable N. So, the best splitting on n is used to split the node. The value of m (node splitter) is constant during growing the forest.
- Each tree is growing up to the largest magnitude possible so there is no trimming.

4. **Gradient Tree Boosting**: is a Machine Learning technique for regression and classification problems, which produces a predication model in the form of an ensemble of weak predication models. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function [14][18].

   It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize errors. How are the targets calculated? The target outcome for each case in the data depends on how much changing that case's prediction impacts the overall prediction error [14][18].

5. **Extra Random Forest (ERF)**: is very similar to Random Forests (RF) but there are two main differences:
    - ERF does not resample observations when building a tree. They do not perform bagging.
    - ERF does not use the best split. Like RF, ERF selects a random subset of predictors for each split. Instead of the best split for predictors, ERF makes a small number of randomly chosen splits-points for each of the selected predictors. ERF then selects the best split from this small number of choices.

    In ERF, the features and splits are selected at random. Since splits are chosen at random for each feature, it is less computationally expensive than RF [14][18].

CHAPTER FIVE

APPLICATION OF MULTIPLE STAGE DATA ALGORITHM - HEART DISEASE PREDICTION

There are many applications to use for our algorithm to predict the outcome of multiple stage data. In this chapter, we will apply our algorithm on one of the popular medical disease, it is the heart disease. We will use our algorithm to apply it to five stage outcome data. Then, will apply the most popular Machine Learning Random Forest, Support Vector Machine, Logistics Regression, Decision Tree, and Nave Bayes into our multiple stage data algorithm and compare the result to see which one performs better than the other. The main objective of this significant research work is to identify the best classification algorithm suitable for providing maximum accuracy when classification of five stages.

In this chapter, we will be talking about predicting of the five stage outcome data of heart disease ranging from not presented, good indication of showing, started developing, high risk and advanced. We investigate different potential supervised models that are trained by Machine Learning algorithms and find out which of these models has better accuracy. Heart Failure outcome could be classified as follows:

- Stage A: No disease is present.
- Stage B: Stage B is considered pre-heart failure. It means you are at high risk of developing heart failure because you have a family history of heart failure, or you have one of more of these medical conditions [20][21][22]:
    - Hypertension.

- Diabetes.
- Coronary artery disease.
- Metabolic syndrome.
- History of alcohol abuse.
- History of rheumatic fever.
- Family history of cardiomyopathy.
- History of taking drugs that can damage the heart muscle, such as some cancer drugs.

- Stage C: Stage C is considered a pre-heart failure. It means you have been diagnosed with systolic left ventricular dysfunction but have never had symptoms of heart failure. Most people with Stage B heart failure have an echocardiogram (echo) that shows an ejection fraction (EF) of 40% or less. This category includes people who have heart failure and reduced EF (HF rEF) due to any cause [21][22].
- Stage D: Patients with Stage C heart failure have been diagnosed with heart failure and have (currently) or had (previously) signs and symptoms of the condition.
- Stage E: Patients with Stage E HF-rEF have advanced symptoms that do not get better with treatment. This is the final stage of heart failure [21][23].

5.1 Heart Disease

Heart disease describes a range of conditions that affect the heart. Diseases under the heart disease umbrella include blood vessel diseases such as coronary artery disease, heart rhythm problems, and congenital heart defects, among others [24]. The term "heart disease" is often used interchangeably with the term "cardiovascular disease" [25]. Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Other heart conditions, such as those that affect the heart's muscle, valves or rhythm, are also considered forms of heart disease.

Heart disease causes roughly 735,000 heart attacks each year in the U.S. killing more than 630,000 Americans. According to the American Heart Association, over 7 million have suffered a heart attack in their lifetime [24][26]. There are several risk factors for heart disease; some are controllable, others are not. Uncontrollable risk factors for heart disease include male, older age, family history of heart disease, being postmenopausal, and race. About half of Americans (47%) have at least one out of three key risk factors for heart disease: high blood pressure, high cholesterol and smoking [20][27].

Genetic factors likely play some role in high blood pressure, heart disease, and other related conditions. However, it is also likely that people with a family history of heart disease share common environments and other factors that may increase their risk. Most people with a significant family history of heart disease have one or more other risk factors. Just as you cannot control your age, sex and race, you cannot control your family history; so, it’s even more important to treat and control any other modifiable risk factors you have.

The risk for heart disease can increase even more when heredity is combined with unhealthy lifestyle choices, such as smoking cigarettes and eating an unhealthy diet [21][28].

Diagnosis for various forms of heart disease can be detected with numerous medical tests, however, predicting heart disease without such tests is very difficult. Machine Learning can help process medical big data and provide hidden knowledge which otherwise would not be possible with the naked eye [29][30].

5.2 Background and State-of-the-Art

A study conducted by Randal S. Olson provides insightful best practice advice for solving bioinformatics problems with Machine Learning, “Data-driven Advice for Applying Machine Learning to Bioinformatics Problems”. He analyzed 13 state-of-the-art commonly used Machine Learning algorithms on a set of 165 publicly available classification problems in order to provide data-driven algorithm recommendations to current researchers. We use these algorithms to perform our prediction on medical diseases and Cyber Security and see which one to perform the best prediction of five stage outcome [14][18].

From his findings, he was able to provide a recommendation of five algorithms with hyperparameters that maximize classifier performance across the tested problems, as well as general guidelines for applying machine learning to supervised classification problems. The recommendations are as follows:

Algorithm	Parameters
GradientBoostingClassifer	Loss= deviance Learning_rate = 0.1, n_estimators = 500 max_depth = 3, max__features = log2
RandomForestClassifier	n_estimators = 500, max__features = 0.25, criterion = entropy
SVC	C=0.01, gamma = 0.1, degree = 3, coef0 = 10.0
ExtraTreesClassifier	n_estimators = 1000, max__features = log2, criterion = entropy
LogisticRegression	C = 1.5, Penalty = L1, Fit_intercept = true

Table 5: Five Machine Learning Algorithms

The database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by Machine Learning researchers to this date. The " Diagnosis of heart disease" field in the figure refers to the presence of heart disease in the patient. It is integer valued from zero (no presence) to 1. Experiments with the Cleveland database have concentrated on attempting to distinguish presence (values 0, 1, 2, 3, 4) [15] [23].

1. Age
2. Sex
3. Chest pain type
4. Resting blood pressure

5. Serum cholesterol
6. Fasting blood sugar
7. Resting electrocardiographic
8. Maximum heart rate achieved.
9. Exercise induced angina.
10. Depression is induced by exercise relative to rest.
11. The slope of the peak exercise
12. Number of major vessels
13. Normal, fixed defect, reversable defect
14. Diagnosis of heart disease (the predicted attribute)

5.3 Methodology:

The proposed methodology uses five classification techniques; Gradient Tree Boosting, Random Forest, Support Vector Machine (SVM), Extra Random Forest, and Logistic Regression to predict heart disease as the proposed methodology shown in Fig 4. These classifiers are used to improve prediction. We applied the classifiers in Fig 5 to heart disease data that comes from the Cleveland dataset to predict in which of five stages a patient has heart problems. The performance of these classifiers is evaluated on the bases of accuracy, precision recall, and F measure [14][18].

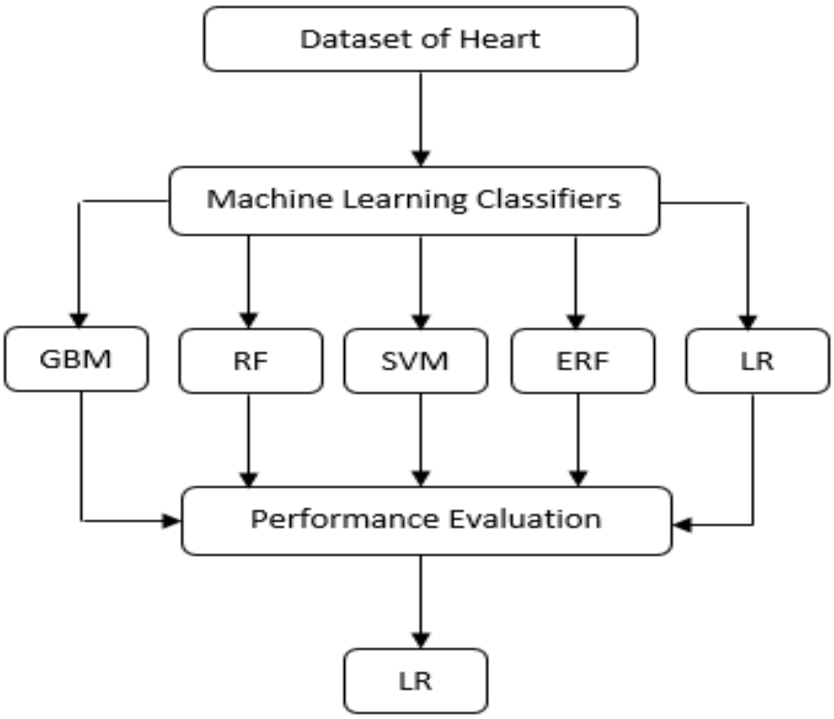

Figure 3: Proposed Heart Disease Prediction Methodology

The dataset of heart is taken from UCI repository, the classifier taking it as input for disease prediction. These classifiers are implemented in Python language. Python is a powerful interpreter language and a reliable platform for research. The accuracy of prediction increased by comparing the results of these five classifiers using evaluation parameters. The experimental result describes which classifier is best between them.

Evaluation Parameters

Some evaluation parameters in data mining are accuracy, precision, recall, and F measure. Where TP True Positive, TN- True Negative, FP- False Positive and FN- False Negative [14][18].

- Accuracy is defined as the number of accurately classified instances divided by the total number of instances in the dataset as in equation 6.

$$\text{Accuracy } = \frac{TP + TN}{TP + FP + TN + FN} \quad (6)$$

- Precision is the average probability of relevant retrieval as described in equation 7.

$$\text{Precision } = \frac{TP}{TP + FP} \quad (7)$$

- The recall is defined as the average probability of complete retrieval as defined in equation 8.

$$\text{Recall } = \frac{TP}{TP + FN} \quad (8)$$

- F- Measure is the calculated by using both precision and recall as shown in equation 9.

$$\text{F Measure } = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (9)$$

Where all evaluation parameters accuracy, precision, recall, and F measure are calculated from dataset when splitting the dataset into training data and test data. The Pseudocodes for the evaluation parameters are as follow [14][18]:

Algorithm 4: Evaluation Parameters Pseudocode

$Define\ evaluationParameters(X\_train, y\_train, X\_test, y\_test)$:

$X\_train \leftarrow fit\_transform(X\_train)$

$Classifier \leftarrow sklearn()$

$$y\_pred \leftarrow classifier.predict(X\_test)$$

$$cm\_test \leftarrow confusion\_matrix(y\_pred, y\_test)$$

$$y\_pred\_train \leftarrow classifier.predict(X\_train)$$

$$cm\_train \leftarrow confusion\_matrix(y\_pred\_train, y\_train)$$

$$training\_accuracy \leftarrow (cm\_train[0][0] + cm\_train[1][1])/len(y\_train)$$

$$test\_accuracy \leftarrow (cm\_test[0][0] + cm\_test[1][1])/len(y\_test)$$

$$training\_percision \leftarrow cm\_train[0][0]/(cm\_train[0][0] + cm\_train[1][0])$$

$$test\_percision \leftarrow cm\_test[0][0]/(cm\_test[0][0] + cm\_test[1][0])$$

$$training\_recall \leftarrow cm\_train[0][0]/(cm\_train[0][0] + cm\_train[0][1])$$

$$test\_recall \leftarrow cm\_test[0][0]/(cm\_test[0][0] + cm\_test[0][1])$$

$$training\_f\_measure \leftarrow (2 * training\_percision * training\_recall)/ (training\_percision + training\_recall))$$

$$test\_f\_measure \leftarrow (2 * test\_percision * test\_recall)/(test\_percision + test\_recall))$$

$$return\ (training\_accuracy, test\_accuracy, training\_percision, test\_percision, training\_recall, test\_recall, training\_f\_measure, training\_f\_measure)$$

Dataset

To perform the research, a heart disease dataset is used. This heart disease dataset contains 14 attributes and 303 instances. This dataset is taken from UCL repository. It's an online repository that contains 412 diverse datasets. UCI provides data for ML to perform analysis in a different direction. The UCI database is known for its extensiveness in data, its completeness, and accuracy.

5.4 Machine Learning Classifiers

In this research, five classification methods are implemented in python using the pandas and keras libraries. These models are used to improve prediction. These classifiers

are compared to find out which of the five stages best predicts the chance of heart disease in patients. In the next section, we briefly describe these classification techniques/ classifiers.

1. **Logistics Regression (LR)**: is the predictive analysis to conduct on discrete (binary) values based on a specified set of independent variables. LR describes the data and clarifies the relationship between one (binary) dependent variable and independent variables. It predicts the event occurrence probability by fitting the data into a logit function. Therefore, it is also called logit regression.

   Input values x is linearly combined using coefficient values b, to calculate an output value p. The output values as predictable lies between 0 and 1. Input data associated with coefficient b (constant value) learned from training data. Where p is the output, b0 is an intercept term and b1 is the coefficient of input value x as shown in equation 10.

$$P = \frac{e^{(b0+b1+x)}}{1 + e^{(b0+b1+x)}} \tag{10}$$

2. **Support Vectors Machine (SVM)**: is a classification and regression algorithm. In SVM, every data item is plotted in n-dimensional space, several dimensions are equivalent to the number of features or attributes. Where n represents the number of attributes. The value of each attribute being the value of certain coordinates. Once plotting all the data items then performed classification by drawing a line or by finding the optimal hyperplane that separates two classes completely. For

example, if we have two features of an individual like hair and height length. First, we plot these two features in two-dimensional space where every point has two coordinates (these co-ordinates are also known as Support Vectors) [14][18].

3. **Random Forests (RF)**: are ensemble learning techniques for regression, regression, classification, and for other tasks. That operate by making a multitude of Decision Tree (DT) at training stint and outputting that is the mean prediction (regression) or mode of classes (classification) of the distinct trees. Every tree in the forest contributes to a classification. To classify new case based on its attributes. We identify the tree "votes" for that class, so the forest indicates the classification of the case that is taking the most votes [14][18].

   Every tree is planted and grown as follows:

   - If the number of objects N in training set, the sample of N objects is taken randomly with replacement. This sample act as a training set for growing tree.
   - If there is an input variable N, number n < N is stated that at each node, randomly selection of n variable out of input variable N. So, the best splitting on n is used to split the node. The value of m (node splitter) is constant during growing the forest.
   - Each tree is growing up to the largest magnitude possible so there is no trimming.

4. **Gradient Tree Boosting**: is a Machine Learning technique for regression and classification problems, which produces a predication model in the form of an ensemble of weak predication models. It builds the model in a stage-wise fashion

like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize errors. How are the targets calculated? The target outcome for each case in the data depends on how much changing that case's prediction impacts the overall prediction error [14][18].

5. Extra Random Forest (ERF): is very similar to Random Forests (RF) but there are two main differences:
    - ERF does not resample observations when building a tree. They do not perform bagging.
    - ERF does not use the best split. Like RF, ERF selects a random subset of predictors for each split. Instead of the best split for predictors, ERF makes a small number of randomly chosen splits-points for each of the selected predictors. ERF then selects the best split from this small number of choices.

    In ERF, the features and splits are selected at random. Since splits are chosen at random for each feature, it is less computationally expensive than RF [14][18].

5.5 Scaling Data

The model that we construct in chapter four is what we will utilize to put into practice the five stage outcome data forecast for a patient who will be diagnosed with heart disease. The scaling process would involve creating separate training datasets for each class, denoted as dataset1 for class A, dataset2 for class B, dataset3 for class C,

dataset4 for class D, and dataset5 for class E. This approach is illustrated in the table provided.

Heart Disease Features			Classes
x1	x2	x3	0
x4	x5	x6	1
x7	x8	x9	2
x10	x11	x12	3
x13	x14	x15	4

Table 6: Main Dataset of Heart Disease

Heart Disease Features			Stage A	Stage B	Stage C	Stage D	Stage E
x1	x2	x3	+1	-1	-1	-1	-1
x4	x5	x6	-1	+1	-1	-1	-1
x7	x8	x9	-1	-1	+1	-1	-1
x10	x11	x12	-1	-1	-1	+1	-1
x13	x14	x15	-1	-1	-1	-1	+1

Table 7: Training Dataset / Class: A, B, C, D, E

5.6 Experiment Result

The experiment is conducted for the prediction of heart disease stages by applying various Machine Learning classifiers. From the experiment results, we identify that Logistic Regression performs better as compared to the other four ML classifiers in the prediction of these diseases. In this experiment, we use multiple stage outcome data of

heart disease prediction to forecast the stage at which a person is determined to have heart disease. The Pseudocodes for the experiment are as follow:

Algorithm 5: Heart Disease Model Pseudocode

Inputs:

- $C,\ a\ binary\ training\ classifier$
- $Samples\ X$
- $Labels\ y\ where\ y_i\ \in \{\ 1 \ldots\ K\ \}$

Outputs:

- $A\ collection\ of\ classifiers\ f_k\ for\ k\ \in \{\ 1 \ldots\ K\ \}$

Procedure:

- $for\ each\ k\ in \{\ 1 \ldots\ K\ \}$
- $build\ a\ new\ lable\ vector\ v\ where\ v_i = \ y_i\ if\ y_i = k\ and\ v_i = 0$
- $Apply\ C\ to\ X, v\ to\ obtain\ f_k$
- $X\_train,\ X\_test,\ y\_train, y\_test\ \leftarrow\ train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)$
- $SVM(X\_train, y\_train, X\_test, y\_test)$

  $evaluationParameters(X\_train, y\_train, X\_test, y\_test)$
- $LR(X\_train, y\_train, X\_test, y\_test)$

  $evaluationParameters(X\_train, y\_train, X\_test, y\_test)$
- $RF(X\_train, y\_train, X\_test, y\_test)$

  $evaluationParameters(X\_train, y\_train, X\_test, y\_test)$
- $GTB(X\_train, y\_train, X\_test, y\_test)$

$$evaluationParameters(X\_train, y\_train, X\_test, y\_test)$$

- $ERF(X\_train, y\_train, X\_test, y\_test)$

$$evaluationParameters(X\_train, y\_train, X\_test, y\_test)$$

The below Figures show the performance of various evaluation parameters in the prediction of heart disease. The experimental results show the comparison of LR, ERF, GTB, SVM and RF classifiers and evaluate the performance on the bases of accuracy, precision, recall and F measure. In all classifiers, LR performs the best with an accuracy of 82%, followed by SVM with an accuracy of 80% [14][18].

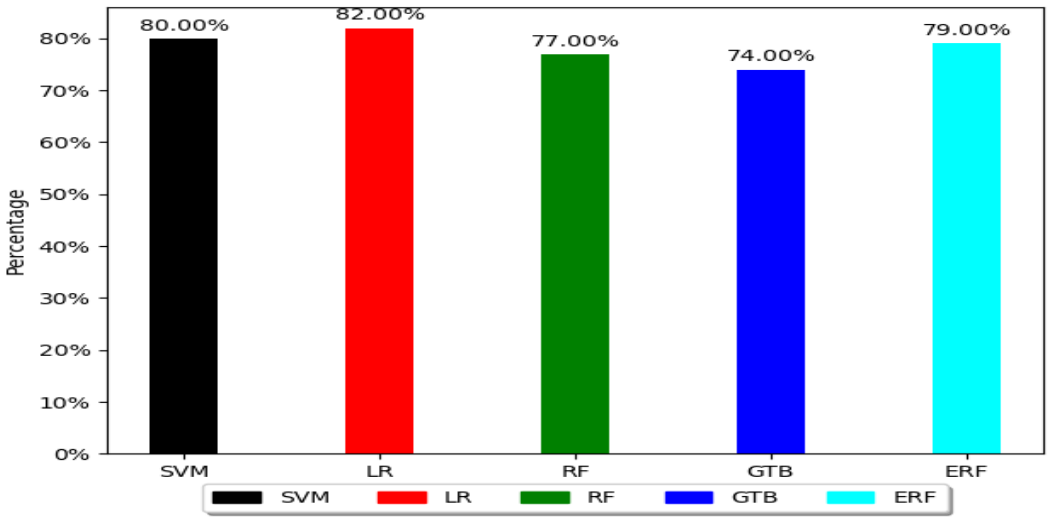

Figure 4: Heart Disease Accuracy

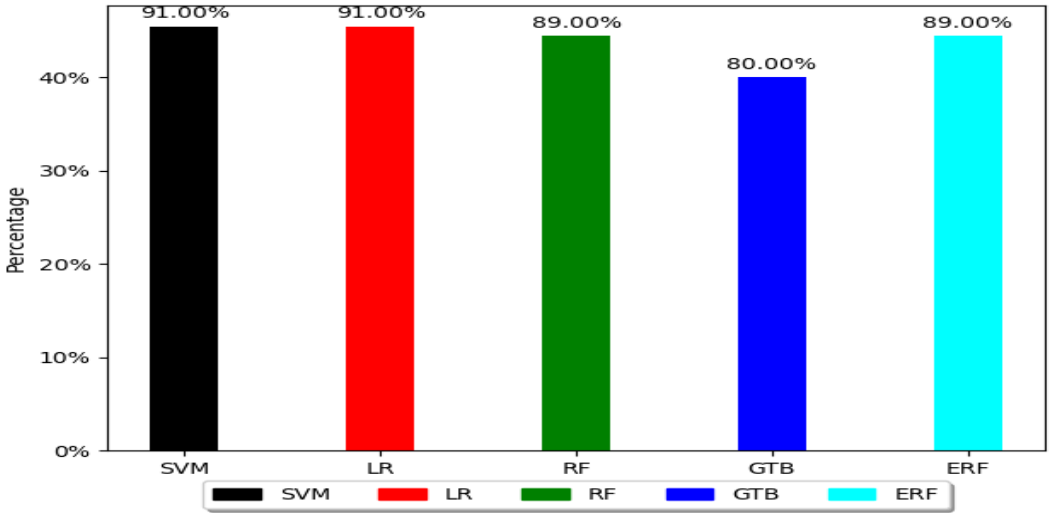

Figure 5: Heart Disease Precision

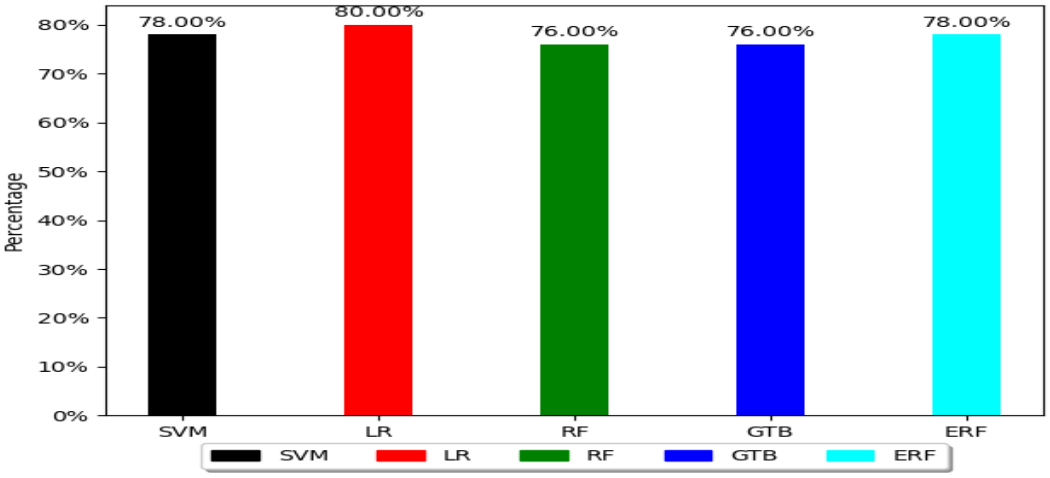

Figure 6: Heart Disease Recall Performance

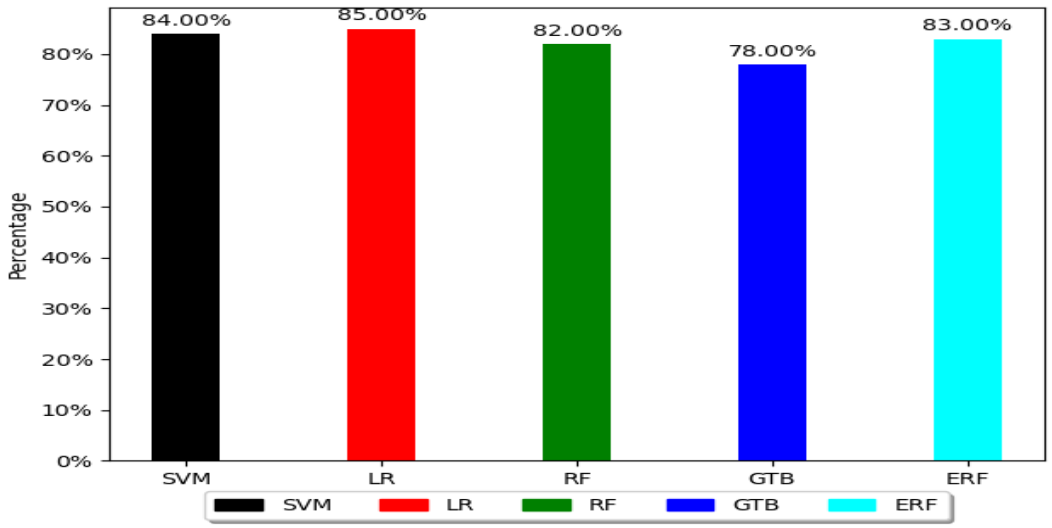

Figure 7: Heart Disease F Measure Performance

Algorithm	Accuracy	Precision	Recall	F Measure
SVM	80%	91%	78%	84%
LR	82%	91%	80%	85%
RF	77%	89%	76%	82%
GTB	74%	80%	76%	78%
ERF	79%	89%	78%	83%

Table 8: ML Algorithms Comparison

CHAPTER SIX

APPLICATION OF MULTIPLE STAGE DATA ALGORITHM - URL PHISHING PREDICTION

As talked in chapter four about applying out multiple stage outcome data on different application. In this chapter, we will apply our algorithm to Cyber Security, specifically URL Phishing attacks. We will use our algorithm to implement three stage outcome data using Multiclass classification. Then, will apply the most popular Machine Learning Random Forest, Support Vector Machine, Logistics Regression, Decision Tree, and Nave Bayes into our multiple stage data algorithm and compare the result to see which one performs better than the other. The main objective of this significant research work is to identify the best classification algorithm suitable for providing maximum accuracy when classification of five stages.

Phishing is a fraudulent process and a form of cybercrime, where an attacker tries to obtain sensitive information for malicious use, such as usernames, passwords, or banking details [30][32]. A phisher uses social engineering and technical deception to fetch private information from the web user. Previous Machine Learning approaches have been used to detect whether URLs is valid or invalid. The purpose of this work is to detect or predict the three stages of Phishing URL starting from valid, suspicious, and invalid URL. We investigate different potential supervised models that are trained by Machine Learning algorithms and find out which of these models has better accuracy. In this work, we describe and investigate five Machine Learning algorithms (SVM, LR, RF, GTB, ERF)

with hyperparameters that maximize classifier performance to show which one is the best to detect the stage at which a URL is determined to be valid, suspicious, or invalid. We found that the LR algorithm performs better compared to the other four algorithms. The experiment results show that LR performs the best with an accuracy of 82%, followed by SVM with an accuracy of 80% when all five classifiers are compared and evaluated for performance based on accuracy, precision, recall, and F measure. This detection can facilitate every step of identifying suspicious URLs, reducing the margin of false positive and true negative errors and contributing to educating users of the serious risk of Phishing websites.

6.1 Machine Learning

Machine Learning is the process of teaching a computer system how to make accurate predictions when provided data. It uses algorithms and neural network models to assist computer systems in progressively improving their performance. Machine Learning algorithms automatically build a mathematical model using sample data – also known as "training data" – to make decisions without being specifically programmed to make those decisions [14][18].

Those predictions could be answering whether a piece of fruit in a photo is a banana or an apple, spotting people crossing the road in front of a self-driving car, whether the use of the word *book* in a sentence relates to a paperback or a hotel reservation, if an email is spam, or recognizing speech accurately enough to generate captions for a YouTube video [1][2][14][18].

With Machine Learning, cybersecurity systems can analyze patterns and learn from them to help prevent similar attacks and respond to changing behavior. It can help cybersecurity teams be more proactive in preventing threats and responding to active attacks in real time. It can reduce the amount of time spent on routine tasks and enable organizations to use their resources more strategically. Also, Machine Learning can make cybersecurity simpler, more proactive, less expensive, and far more effective. In order to conduct this detection, a Jupyter notebook was constructed in Python using the Phishing Websites Dataset that is put by the University California, Irvine.

6.2 Phishing

Phishing is a form of fraudulent attack where the attacker tries to gain sensitive information by posing as a reputable source. In a typical phishing attack, a victim opens a compromised link that poses as a credible website. The victim is then asked to enter their credentials, but since it is a “fake” website, the sensitive information is routed to the hacker and the victim gets hacked [31][33][34].

Phishing is popular since it is a low effort, high reward attack. Most modern web browsers, antivirus software and email clients are pretty good at detecting phishing websites at the source, helping to prevent attacks The phishing web pages generally have alike page layouts, blocks and fonts to mimic legitimate web pages in an endeavor to influence web users to obtain personal details such as username and password. [33][34].

Phishing is a quickly growing type of fraud and is considered as one of the foremost dangerous threats within the web which causes folks to mislay guarantee in on-line transactions. It is relatively a current web crime as compared with virus hacking and remains an ominous threat to client and business round the world [35].

According to the RSA's online fraud report, the year 2013 has been confirmed to be a record year where many phishing attacks have been launched globally. Additional1y, RSA estimates that over USD $5.9 billion was lost by global organizations due to phishing attacks in the same period. The Internet Security Threat Report 2014 reports that cybercrimes are prevailing and damaging threats from cybercriminals still emerge over businesses and customers. According to RSA monthly fraud report January 2014, the big data analytics and broader intelligence will lead to faster detection resulting in lower financial losses. Data mining techniques are used to extract helpful information by analyzing the past information then predicting the future incidents [34][35].

Phishing attacks' analogy is derived from "fishing" for victims, this type of attack has attracted a great deal of attention from researchers in recent years. It is also a promising and attractive technique for attackers who open some fraudulent websites, which have exactly similar design of the popular and legal sites on the Internet. Although these pages have similar graphical user interfaces, they must have different Uniform Resource Locators (URLs) from the original page. Mainly, a careful and experienced user can easily detect these malicious web pages by looking at the URLs. However, due to the speed of life, most of the time, end users do not investigate the whole address of their active web page, which is generally forwarded by other web pages, social networking tools or by simply an email message as shown in figure 1. By using this type of fraudulent URLs, a phisher tries to capture some sensitive and personal information of the victim like financial data, personal information, username, password, etc. [35][36].

In the case of entering this type of fraudulent site, which is believed to be the original website, computer users can easily give their sensitive information without any doubt. Because the entered web page seems exactly same with the original web page [35]. In many studies about the user experiences of phishing attacks, computer users fall for phishing due to the five main reasons [35]:

- Users don't have detailed knowledge about URLs.
- Users don't know which web pages can be trusted.
- Users don't see the whole address of the web page, due to the redirection or hidden URLs.
- Users don't have much time for consulting the URL, or accidentally enter some web pages.
- Users cannot distinguish phishing web pages from the legitimate ones.

Phishing attacks exploit the vulnerabilities of the human users; therefore, some additional support systems are needed for the protection of the systems/users. The protection mechanisms are classified into two main groups: by increasing the awareness of the users and by using some additional programs. These programs or software detections can use Machine Learning to detect Phishing. Due to the vulnerability of the end user, an attacker can even target some experienced users by using new techniques and before giving the sensitive information, he is believed that this page is legitimate. Therefore, software-based phishing detection systems are preferred as decision support systems for the user [35][36][37].

In this work, we are focused on the multiple stage detection of phishing web pages by investigating the URL of the web page with different five Machine Learning algorithms and different feature sets. In the execution of a learning algorithm, not only the dataset but also the extraction of the features from this dataset are crucial. Therefore, we will use the University of California, Irvine dataset, five Machine Learning algorithms, and Python to build the phishing detector. Our work will [35]:

- Identify the criteria that can recognize fake URLs.
- Build five Machine Learning algorithms that can iterate through the criteria.
- Train our model to recognize fake vs real URLs.
- Evaluate our model to see how it performs.

Three outcome values, valid, suspicious, and invalid URL will be predicated based on the parameters in the dataset to detect the phishing URL.

6.3 URL's and Attacker's Techniques

Attackers use different types of techniques for not to be detected either by security mechanisms or system admins. In this section, some of these techniques will be detailed. To understand the approach of attackers, firstly, the components of URLs should be known. The basic structure of a URL is shown in the figure below [35].

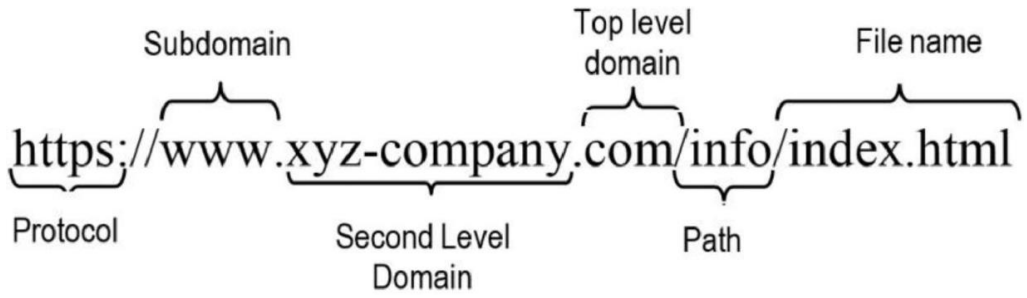

Figure 8: URL structure

In the standard form, a URL starts with its protocol name used to access the web page. After that, the subdomain, and the Second Level Domain (SLD) name, which commonly refers to the organization name in the server hosting, is located and finally the Top-Level Domain (TLD) name, which shows the domains in the DNS root zone of the Internet takes place. The previous parts compose the domain name (host name) of the web page; however, the inner address is represented by the path of the page in the server and with the name of the page in the HTML form [35][36].

Although SLD name generally shows the type of activity or company name, an attacker can easily find or buy it for phishing. The name of SLD can only be set once, at the beginning. However, an unlimited number of URLs can be generated by an attacker by extending the SLD by path and file names, because the inner address design directly depends on attackers [35].

The unique (and critical) part of a URL is the composition of SLD and TLD, which is named as domain name. Therefore, cybersecurity companies make a great effort to identify the fraudulent domains by name, which are used for phishing attacks. If a domain name is identified as phishing, the IP address can be easily blocked to prevent from accessing the web pages located in it [35].

To increase the performance of the attack and steal more sensitive information, an attacker mainly uses some important methods to increase the vulnerability of victims such as the use of random characters, combined word usage, cybersquatting, typosquatting, etc. Therefore, detection mechanisms should take into consideration these attack methods [35].

6.4 Background on Recommendation of Model Algorithms

We use the same five recommended algorithms to implement our three stages prediction of URL Phishing attacks. The algorithms again are [14][18]:

Table 9: Five Machine Learning Algorithms

Algorithm	**Parameters**	**Datasets Covered**
GradientBoostingClassifer	Loss= deviance Learning_rate = 0.1, n_estimators = 500 max_depth = 3, max__features = log2	
RandomForestClassifier	n_estimators = 500, max__features = 0.25, criterion = entropy	19
SVC	C=0.01, gamma = 0.1, degree = 3, coef0 = 10.0	16
ExtraTreesClassifier	n_estimators = 1000, max__features = log2, criterion = entropy	12
LogisticRegression	C = 1.5, Penalty = L1, Fit_intercept = true	8

The database is provided by the University of California. Irvine. The feature sets of the dataset are divided four main categories [14][18][35]:

1. **Address Bar-Based Features** – these are features extracted from the URL itself, like URL length >54 characters, or whether it contains an IP address, uses an URL shortening service like TinyURL or Bitly, or employs redirection. Addition features may also include:
    - Adding a prefix or suffix separated by (-) to the domain.
    - Having sub-domain and multi-sub-domains
    - Existence of HTTPS
    - Domain registration age
    - Favicon loading from a different domain.
    - Using a non-standard port
2. **Abnormal Features** – these may include:
    - Loading images loaded in the body from a different URL.
    - Minimal use of meta tags
    - The use of a Server Form Handler (SFH)
    - Submitting information to email
    - An abnormal URL
3. **HTML and JavaScript-Based Features** - these can include things like:
    - Website forwarding
    - Status bar customization typically using JavaScript to display a fake URL.
    - Disabling the ability to right-click so users can't view page source code.
    - Using pop-up windows

- iFrame redirection

4. **Domain-Based Features** – these can include:
    - Unusually young domains
    - Suspicious DNS record
    - Low volume of website traffic
    - PageRank, where 95% of phishing webpages have no PageRank.
    - Whether the site has been indexed by Google

6.5 Approach

A fraudulent domain or phishing domain is an URL scheme that looks suspicious for a variety of reasons. Most commonly, the URL [35]:

- Is misspelled.
- Points to the wrong top-level domain
- A combination of a valid and a fraudulent URL
- Is incredibly long.
- Is just be an IP address.
- Has a low page rank?
- Has a young domain age
- Ranks poorly on the Alexa Top 1 million Sites

All these are characteristics of a phishing URL that can help us distinguish it from a valid URL. These characteristics can be converted into Machine Learning feature sets such

as numbers, labels and Booleans. Our aim is to extend the outcome values of the predicated URL from binary values, yes or no, 1 or 0 to multiple stages or multiple outcome values. This includes -1 for invalid URL, 0 for suspicious, and 1 for valid URL.

The Dataset is split into 80% training and 20% testing. Giving all the criteria that can help us identify phishing URLs, we can use the five Machine Learning algorithms classifiers to help us predict whether an URL is valid, has no info to decide, or invalid URL [14][18].

1. Gradient Tree Boosting (GradientBoostingClassifier).
2. Random Forest (RandomForestClassifier).
3. Support Vector Machine (SVC).
4. Extra Random Forest (ExtraTreesClassifier).
5. Logistic Regression (LogisticRegression).

6.6 Methodology

The proposed methodology uses five classification techniques; Gradient Tree Boosting, Random Forest, Support Vector Machine (SVM), Extra Random Forest, and Logistic Regression to predict phishing URL as the proposed methodology shown in Fig 5. These classifiers are used to improve prediction. We applied the classifiers in Fig 5 to URL Phishing data that comes from University of California, Irvine dataset to predict in which of three stages a URL is valid, suspicious, or invalid. The performance of these classifiers is evaluated on the bases of accuracy, precision recall, and F measure [14][18].

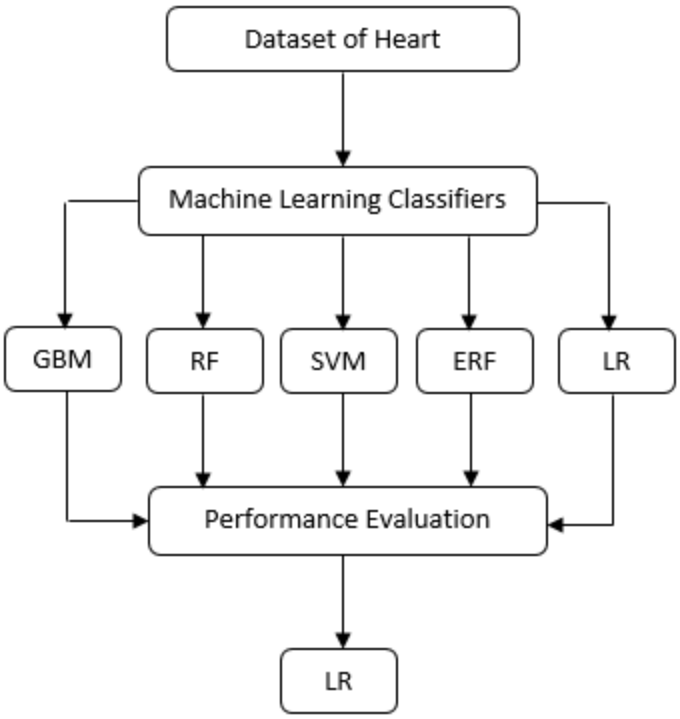

Figure 9: Proposed Methodology

To perform the research, the University of California, Irvine dataset is used. This dataset is taken from UCL repository. It's an online repository that contains 412 diverse datasets. UCI provides data for ML to perform analysis in a different direction. The UCI database is known for its extensiveness in data, its completeness, and accuracy.

Evaluation Parameters

Some evaluation parameters in data mining are accuracy, precision, recall, and F measure. Where TP True Positive, TN- True Negative, FP- False Positive and FN- False Negative [14][18].

- Accuracy is defined as the number of accurately classified instances divided by the total number of instances in the dataset as in equation 11.

$$\text{Accuracy } = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

- Precision is the average probability of relevant retrieval as described in equation 12.

$$\text{Precision } = \frac{TP}{TP + FP} \tag{12}$$

- The recall is defined as the average probability of complete retrieval as defined in equation 13.

$$\text{Recall } = \frac{TP}{TP + FN} \tag{13}$$

- F- Measure is the calculated by using both precision and recall as shown in equation 14.

$$\text{F Measure } = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{14}$$

Where all evaluation parameters accuracy, precision, recall, and F measure are calculated from dataset when splitting the dataset into training data and test data. The Pseudocodes for the evaluation parameters are as follow:

Algorithm 6: Evaluation Parameters Pseudocode

$Define\ evaluationParameters(X\_train, y\_train, X\_test, y\_test)$:

$X\_train \leftarrow fit\_transform(X\_train)$

$Classifier \leftarrow sklearn()$

$y\_pred \leftarrow classifier.predict(X\_test)$

$cm\_test \leftarrow confusion\_matrix(y\_pred, y\_test)$

$y\_pred\_train \leftarrow classifier.predict(X\_train)$

$cm\_train \leftarrow confusion\_matrix(y\_pred\_train, y\_train)$

$training\_accuracy \leftarrow (cm\_train[0][0] + cm\_train[1][1])/len(y\_train)$

$test\_accuracy \leftarrow (cm\_test[0][0] + cm\_test[1][1])/len(y\_test)$

$training\_percision \leftarrow cm\_train[0][0]/(cm\_train[0][0] + cm\_train[1][0])$

$test\_percision \leftarrow cm\_test[0][0]/(cm\_test[0][0] + cm\_test[1][0])$

$training\_recall \leftarrow cm\_train[0][0]/(cm\_train[0][0] + cm\_train[0][1])$

$test\_recall \leftarrow cm\_test[0][0]/(cm\_test[0][0] + cm\_test[0][1])$

$training\_f\_measure \leftarrow (2 * training\_percision * training\_recall)/$
$(training\_percision + training\_recall))$

$test\_f\_measure \leftarrow (2 * test\_percision * test\_recall)/(test\_percision +$
$test\_recall))$

$return\ (training\_accuracy, test\_accuracy, training\_percision,$
$test_{percision}, training_{recall}, test_{recall}, training_{f_{measure}},$
$training\_f\_measure)$

6.7 Scaling the Data

Similar to the approach employed in chapter five for the application of our model in the context of predicting heart disease using multiple stage outcome data, we employ a similar methodology in this chapter to apply our model for the prediction of URL phishing using multiple stage outcome data.

The scaling procedure would involve the creation of multiple training datasets, each dedicated to a certain class. These datasets, denoted as training dataset1 for class Invalid URL, training dataset2 for class Suspicious, and training dataset3 for class Valid

URL, are derived from the original dataset. The allocation of data into these training datasets is illustrated in the table provided.

Heart Disease Features			Classes
x1	x2	x3	0
x4	x5	x6	1
x7	x8	x9	2

Table 10: Phishing URL Main Dataset

URL Phishing Features			Stage Invalid	Stage Suspicious	Stage Valid
x1	x2	x3	+1	-1	-1
x4	x5	x6	-1	+1	-1
x7	x8	x9	-1	-1	+1

Table 11: Training Dataset / Class: Invalid, Suspicious, Valid

6.8 Experiment Result

The experiment is conducted for the prediction of Phishing URL stages by applying various Machine Learning classifiers. From the experiment results, we identify that Logistic Regression performs better as compared to the other four ML classifiers in the prediction of Phishing URLs. In this experiment, we use multiple stages of phishing URL prediction to forecast if the URL is invalid, suspicious, or valid. In previous works, the study used two outcome

predications, either has a URL is valid or invalid; that is represented by (0 ,1) or (true, false). The Pseudocodes for the experiment are as follow [14][18]:

Algorithm 7: Phishing URL Model Pseudocode

Inputs:

- $C,\ a\ binary\ training\ classifier$
- $Samples\ X$
- $Labels\ y\ where\ y_i\ \in \{\ 1 \dots\ K\ \}$

Outputs:

- $A\ collection\ of\ classifiers\ f_k\ for\ k\ \in \{\ 1 \dots\ K\ \}$

Procedure:

- $for\ each\ k\ in \{\ 1 \dots\ K\ \}$
- $build\ a\ new\ lable\ vector\ v\ where\ v_i = \ y_i\ if\ y_i = k\ and\ v_i = 0$
- $Apply\ C\ to\ X, v\ to\ obtain\ f_k$
- $X\_train,\ X\_test,\ y\_train, y\_test\ \leftarrow\ train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)$
- $SVM(X\_train, y\_train, X\_test, y\_test)$

  $evaluationParameters(X\_train, y\_train, X\_test, y\_test)$
- $LR(X\_train, y\_train, X\_test, y\_test)$

  $evaluationParameters(X\_train, y\_train, X\_test, y\_test)$
- $RF(X\_train, y\_train, X\_test, y\_test)$

  $evaluationParameters(X\_train, y\_train, X\_test, y\_test)$
- $GTB(X\_train, y\_train, X\_test, y\_test)$

  $evaluationParameters(X\_train, y\_train, X\_test, y\_test)$
- $ERF(X\_train, y\_train, X\_test, y\_test)$

  $evaluationParameters(X\_train, y\_train, X\_test, y\_test)$

The figures below depict the performance of several evaluation factors in predicting phishing URLs. The experimental results include a comparative analysis of the LR, ERF, GTB, SVM, and RF classifiers, along with an assessment of their performance using metrics such as accuracy, precision, recall, and F measure. When examining the performance of several classifiers on this dataset, it becomes evident that Support Vector Machines (SVM) and Logistic Regression (LR) exhibit an accuracy rate of 82%.

In the given dataset, the Support Vector Machine (SVM) exhibits the highest precision rate, standing at 82%. Following closely are the Random Forest (RF) and Gradient Tree Boosting (GTB) models, both achieving a precision rate of 81%. In the context of this specific dataset, the logistic regression (LR) model exhibits a recall rate of 88%. Finally, it is worth noting that both Support Vector Machines (SVM) and Logistic Regression (LR) exhibit an f-measure of 83% when applied to this particular dataset. In comparison to alternative classifiers, the Support Vector Machine (SVM) and Logistic Regression (LR) have demonstrated greater performance on the given dataset.

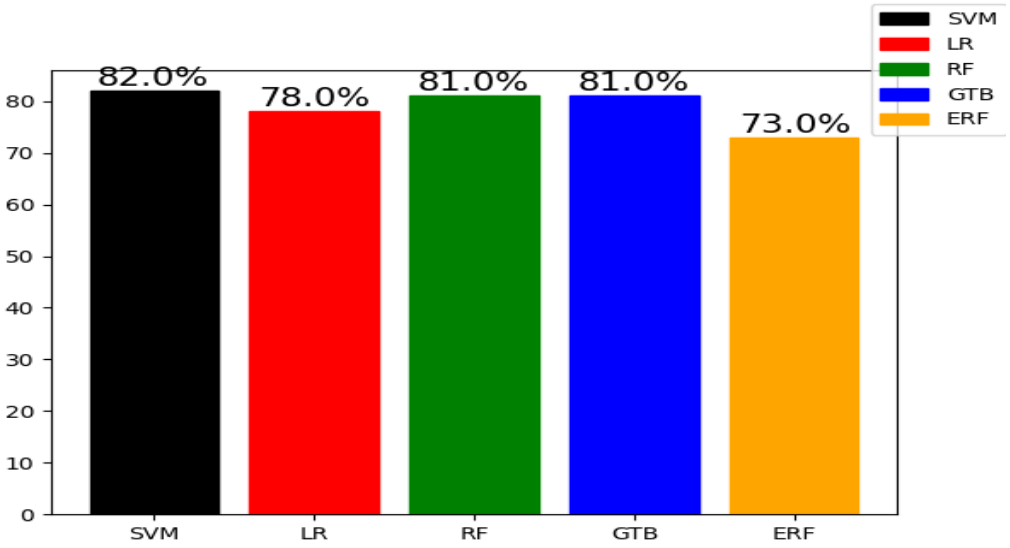

Figure 10: Phishing URL Accuracy Performance

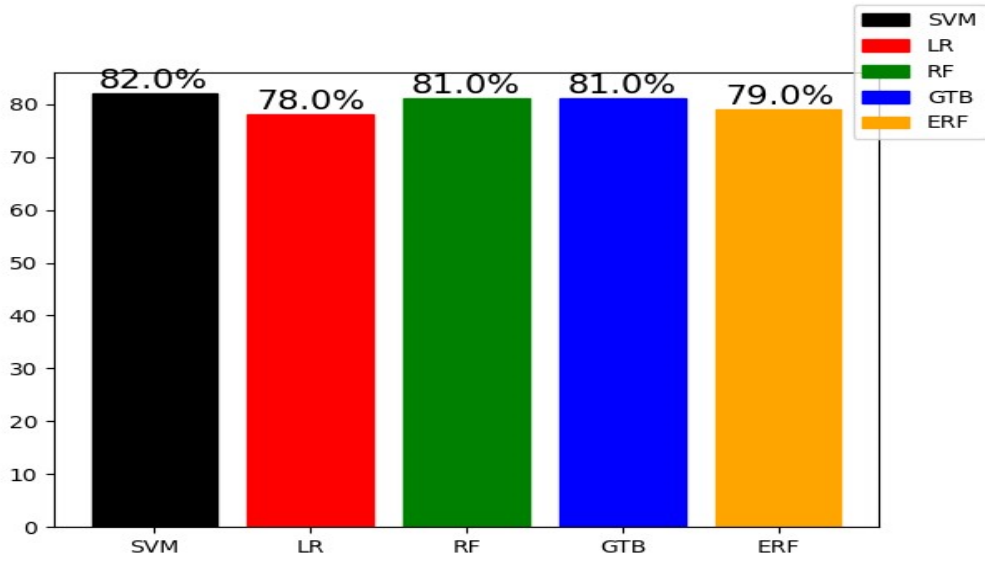

Figure 11: Phishing URL Precision Performance

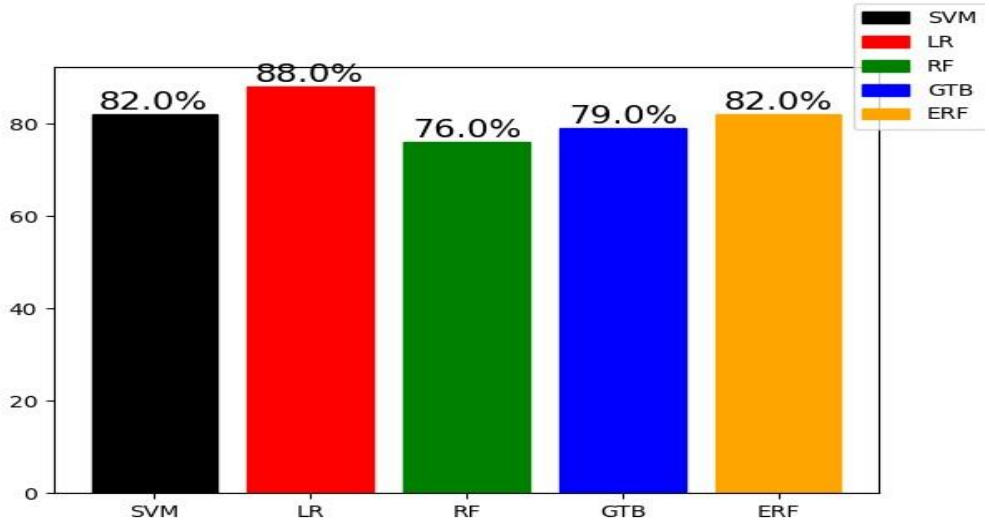

Figure 12: Phishing URL Recall Performance

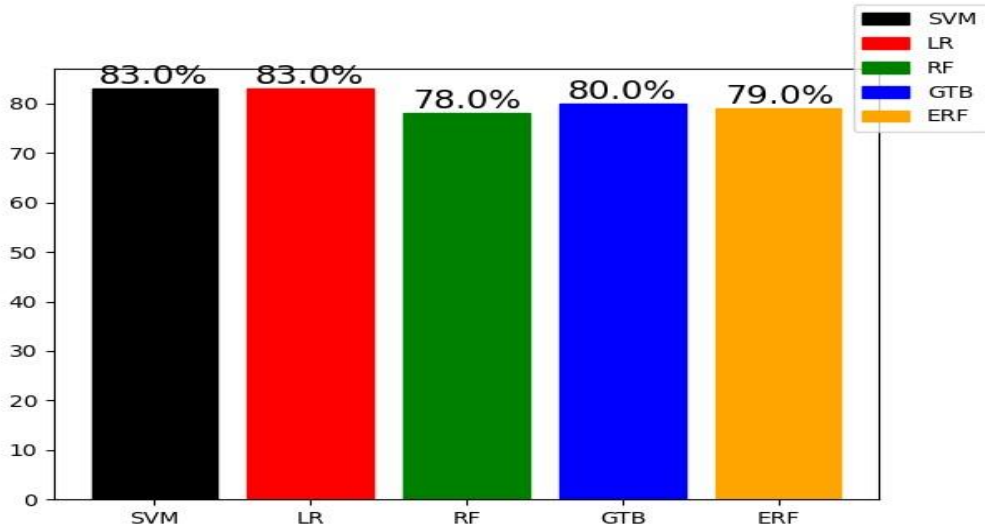

Figure 13: Phishing URL F Measure Performance

Algorithm	Accuracy	Precision	Recall	F Measure
SVM	82%	82%	82%	83%
LR	78%	78%	88%	83%
RF	81%	81%	76%	78%
GTB	81%	81%	79%	80%
ERF	73%	79%	82%	79%

Table 12: ML Algorithms Comparison

CHAPTER SEVEN

APPLICATIONS IN DIGITAL TWIN

A "Digital Twin" is a digital replica or representation of a real-world entity or system [38]. This digital model mirrors its physical counterpart in real-time by continuously collecting data via sensors, IoT devices, and other data sources. Digital twins are used to analyze, simulate, and optimize systems and processes, leading to improved efficiency, reduced costs, and enhanced performance. There are various types of digital twins depending on the level of product magnification [38][39]. The biggest difference between these twins is the area of application. It is common to have different types of digital twins co-exist within a system or process. The core elements and applications of digital twins [39][40]:

1. **Elements**:
    - Physical Items or Systems: This could be anything from machines, buildings, and entire cities to human organs.
    - Sensors: Gather data from the physical item or system.
    - Data & Connectivity: Data from sensors is transmitted, often in real-time, to the digital counterpart.
    - Digital Representation: The data is processed and visualized in a way that mirrors the real-world item or system.

2. **Applications**:
    - Manufacturing: Monitoring machinery in real-time, predicting when maintenance is needed, and optimizing production processes.
    - Healthcare: Creating digital replicas of patients' organs or systems to simulate different treatment approaches.
    - Smart Cities: Optimizing traffic flow, energy consumption, and other urban processes.
    - Agriculture: Monitoring and predicting crop health, optimizing irrigation and pest control.
    - Energy: Real-time monitoring and predictive maintenance for components in power plants or renewable energy sources, such as wind turbines.
    - Real Estate and Construction: Monitoring the structural health of buildings or simulating different construction methods.
    - Automotive & Transportation: Simulating vehicle performance under various conditions or optimizing fleet operations in real-time.
3. **Benefits**:
    - Optimization: By having a real-time digital counterpart, businesses can simulate and test changes, optimizing processes before implementing them in the real world.
    - Predictive Maintenance: Identify issues before they become critical, saving time and money.
    - Flexibility: Test new scenarios or "what-if" situations without affecting the actual system.

- Enhanced Data Analysis: Turn raw data into actionable insights, allowing for better decision-making.

7.1 Digital Twin and Multiple Stage Data

Multiple stage data in relation to digital twins refers to the lifecycle or stages that data goes through in the digital twin environment. Each stage adds value and context to the raw data, transforming it into actionable insights and enabling better decision-making processes [41]. In essence, the digital twin's multiple stage data lifecycle offers a comprehensive framework for collecting, processing, analyzing, and acting upon data. This lifecycle ensures that decision-makers get the most accurate and actionable insights from their digital twins. The following is the outline of multiple stage data processing within the digital twin [41][42][43]:

1. **Data Collection**:
    - Sensors and IoT Devices: Gather raw data from the physical entity. This can be from embedded sensors, cameras, external monitoring equipment, and other data sources.
    - External Data Sources: Integrate weather data, market prices, user behavior data, etc., depending on the application.
2. **Data Transmission**:
    - Edge Computing: Preliminary data processing may happen at the edge (near the data source) to filter, aggregate, or compress data before transmission.
    - Secure Transmission: Data is securely sent from the physical system to the digital twin system, ensuring data integrity and privacy.

3. **Data Storage**:
    - Time-Series Databases: Store temporal data in an optimized way, allowing for efficient querying.
    - Blob Storage: For unstructured data such as images or logs.
    - Distributed Storage: To ensure scalability, fault tolerance, and high availability.
4. **Data Processing and Cleaning**:
    - Normalization: Scaling data to a standard range.
    - Anomaly Detection: Identifying and handling outliers or erroneous data points.
    - Interpolation: Handling missing data points.
5. **Data Analysis and Simulation**:
    - Descriptive Analysis: Understand the current state of the physical system.
    - Diagnostic Analysis: Find out why certain events occurred.
    - Predictive Analysis: Predict future states or events based on historical data.
    - Simulation: Using the digital twin to run scenarios or "what-if" analyses.
6. **Continuous Learning and Model Updating**:
    - Machine Learning Integration: Over time, Machine Learning models can learn from the data and improve the accuracy of predictions and analyses.
    - Model Refinement: The digital twin's representation may need adjustments as the physical system evolves or as more data becomes available.

7. **Action & Feedback Loop**:
    - Automation: The digital twin can trigger automated actions based on the data analysis. For instance, it might initiate maintenance procedures when wear and tear reach a certain threshold.
    - Human Intervention: Decision-makers are informed and can take strategic actions.
    - Feedback to Physical System: Adjustments are made in the real world based on insights from the digital twin, and the process continues in a loop.

CHAPTER EIGHT

CONCLUSION AND FUTURE WORK

8.1 Conclusion

In this research, we developed a Machine Learning multiple stage model to predict multiple stage outcome data using Multiclass Classification One-vs-All approach. Then, we implemented and applied five Machine Learning algorithms (SVM, LR, RF, GTB, ERF) on our Machine Learning model for multiple stage data to show which one is the best to predict the outcome. In comparison to alternative classifiers, the Support Vector Machine (SVM) and Logistic Regression (LR) have demonstrated greater performance on the given dataset based on accuracy, precision, recall and F measure. Our multiple stage outcome model made a series of decisions. This research demonstrates that significance of multiple stage methods using Multiclass Classification when measuring accuracy, precision, recall and F measure.

8.2 Future Work

There are many possible improvements, extensions, or new directions that could be explored to improve the prediction of this study. Due to time limitations, the following research/ work needs to be performed in the future:

- There is a need for more ML algorithms for comparison.
- Large dataset to be trained.
- Build a system or a framework for automation of multiple stage outcome data conversion.

- More data from different sectors needs to be collected and all the available techniques will be compared for the optimum accuracy.
- Use deep learning to structure algorithms in layers to create Artificial Neural Network (ANN) or Convoluted Neural Network (CNN) that can learn and make intelligent decision.
- Complex Ensembles and Stacking: Further investigate the use of stacked models or complex ensembles that take outputs from one stage and use them as inputs for another.
- Transfer Learning & Pre-training: Explore the utility of pre-training on a large dataset and fine-tuning on the specific multi-stage data. Incorporate knowledge from related tasks or domains to improve performance.
- Incorporate Domain Knowledge: Work closely with domain experts to introduce new features or refine the multi-stage process.
- End-to-end Machine Learning: Investigate end-to-end Machine Learning models that can inherently handle multiple stage processing without manual feature engineering.
- Robustness and Generalization: Improve the robustness of models to adversarial attacks or out-of-distribution samples. Explore techniques like regularization, dropout, and Bayesian neural networks.
- Model Interpretability: Delve into model interpretability tools and techniques to understand how the multiple stage processing affects model decisions.

- Optimization Techniques: Explore advanced optimization methods to improve convergence and model performance, especially if the multiple stage process introduces non-trivial dynamics.
- Active Learning & Semi-supervised Learning: Given that multiple stage processing can sometimes result in smaller refined datasets, techniques that make the most of limited labeled data can be explored.
- Feedback Loops: Introduce mechanisms for the model to feedback information to earlier stages, making the multiple stage process more dynamic.
- Integration with Other Systems: Explore how the multi-stage Machine Learning process can be integrated with other systems, such as databases, IoT devices, or web applications.

REFERENCES

Machine Learning

[1] El Naqa, Issam, and Martin J. Murphy. What is Machine Learning ?. Springer International Publishing, 2015.

[2] Bi, Qifang, et al. "What is Machine Learning ? A primer for the epidemiologist." American journal of epidemiology 188.12 (2019): 2222-2239.

[3] Chaabene, Wassim Ben, Majdi Flah, and Moncef L. Nehdi. "Machine Learning prediction of mechanical properties of concrete: Critical review." Construction and Building Materials 260 (2020): 119889.

[4] Helm, J. Matthew, et al. "Machine Learning and artificial intelligence: definitions, applications, and future directions." Current reviews in musculoskeletal medicine 13 (2020): 69-76.

[5] Joshi, Ameet V. "Machine Learning and artificial intelligence." (2020).

[6] Alzubi, Jafar, Anand Nayyar, and Akshi Kumar. "Machine Learning from theory to algorithms: an overview." Journal of physics: conference series. Vol. 1142. IOP Publishing, 2018.

[7] Mahesh, Batta. "Machine Learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9.1 (2020): 381-386.

[8] Daumé, Hal. A course in Machine Learning . Hal Daumé III, 2017.

[9] Nasteski, Vladimir. "An overview of the supervised Machine Learning methods." Horizons. b 4 (2017): 51-62.

[10] Athey, Susan, and Guido W. Imbens. "Machine Learning methods that economists should know about." Annual Review of Economics 11 (2019): 685-725.

[11] Bi, Qifang, et al. "What is Machine Learning ? A primer for the epidemiologist." American journal of epidemiology 188.12 (2019): 2222-2239.

[12] Injadat, MohammadNoor, et al. "Multi-stage optimized Machine Learning framework for network intrusion detection." IEEE Transactions on Network and Service Management 18.2 (2020): 1803-1816.

[13] Mardani, Abbas, et al. "A multi-stage method to predict carbon dioxide emissions using dimensionality reduction, clustering, and Machine Learning techniques." Journal of Cleaner Production 275 (2020): 122942.

[14] Amen, Khalid, Mohamed Zohdy, and Mohammed Mahmoud. "Machine Learning for multiple stage heart disease prediction." Proceedings of the 7th International Conference on Computer Science, Engineering and Information Technology. 2020.

[15] Ghezelbash, Reza, Abbas Maghsoudi, and Emmanuel John M. Carranza. "Performance evaluation of RBF-and SVM-based Machine Learning algorithms for predictive mineral prospectivity modeling: integration of SA multifractal model and mineralization controls." Earth Science Informatics 12 (2019): 277-293.

[16] Zhu, Mengqi, et al. "Performance Evaluation Indicator (PEI): A new paradigm to evaluate the competence of Machine Learning classifiers in predicting rockmass conditions." Advanced Engineering Informatics 47 (2021): 101232.

[17] Sahoo, Abhaya Kumar, Chittaranjan Pradhan, and Himansu Das. "Performance evaluation of different Machine Learning methods and deep-learning based convolutional neural network for health decision making." Nature inspired computing for data science (2020): 201-212.

[18] Amen, Khalid, Mohamad Zohdy, and Mohammed Mahmoud. "Machine Learning for Multiple Stage Phishing URL Prediction." 2021 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2021.

[19] Ramzan, Farheen, et al. "A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks." Journal of medical systems 44 (2020): 1-16.

Heart Disease

[20] Khemphila, Anchana, and Veera Boonjing. "Heart disease classification using neural network and feature selection." 2011 21st International Conference on Systems Engineering. IEEE, 2011.

[21] Deekshatulu, B. L., and Priti Chandra. "Classification of heart disease using k-nearest neighbor and genetic algorithm." Procedia technology 10 (2013): 85-94.

[22] Masethe, Hlaudi Daniel, and Mosima Anna Masethe. "Prediction of heart disease using classification algorithms." Proceedings of the world Congress on Engineering and computer Science. Vol. 2. No. 1. 2014.

[23] Viskin, Sami, and Bernard Belhassen. "Polymorphic ventricular tachyarrhythmias in the absence of organic heart disease: classification, differential diagnosis, and implications for therapy." Progress in cardiovascular diseases 41.1 (1998): 17-34.

[24] Ulbricht, T. L. V., and D. A. T. Southgate. "Coronary heart disease: seven dietary factors." The lancet 338.8773 (1991): 985-992.

[25] Pozuelo, Leo, et al. "Depression and heart disease: what do we know, and where are we headed." Cleve Clin J Med 76.1 (2009): 59-70.

[26] Humphries, Steve E., et al. "coronary heart disease risk prediction in the era of genome-wide association studies: current status and what the future holds." Circulation 121.20 (2010): 2235-2248.

[27] Mackay, Judith, and George A. Mensah. The atlas of heart disease and stroke. World Health Organization, 2004.

[28] Centers for Disease Control and Prevention (CDC. "Prevalence of coronary heart disease--United States, 2006-2010." MMWR. Morbidity and mortality weekly report 60.40 (2011): 1377-1381.

[29] Maganti, Kameswari, et al. "Valvular heart disease: diagnosis and management." Mayo Clinic Proceedings. Vol. 85. No. 5. Elsevier, 2010.

[30] Shouman, Mai, Tim Turner, and Rob Stocker. "Using data mining techniques in heart disease diagnosis and treatment." 2012 Japan-Egypt Conference on Electronics, Communications and Computers. IEEE, 2012.

Phishing

[31] Alkhalil, Zainab, et al. "Phishing attacks: A recent comprehensive study and a new anatomy." Frontiers in Computer Science 3 (2021): 563060.

[32] Gupta, Brij B., et al. "Fighting against phishing attacks: state of the art and future challenges." Neural Computing and Applications 28 (2017): 3629-3654.

[33] Verkijika, Silas Formunyuy. "“If you know what to do, will you take action to avoid mobile phishing attacks”: Self-efficacy, anticipated regret, and gender." Computers in Human Behavior 101 (2019): 286-296.

[34] Bhavsar, Vaishnavi, Aditya Kadlak, and Shabnam Sharma. "Study on phishing attacks." International Journal of Computer Applications 182.33 (2018): 27-29.

[35] Amen, Khalid, Mohamad Zohdy, and Mohammed Mahmoud. "Machine Learning for Multiple Stage Phishing URL Prediction." 2021 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2021.

[36] Abroshan, Hossein, et al. "Phishing attacks root causes." Risks and Security of Internet and Systems: 12th International Conference, CRiSIS 2017, Dinard, France, September 19-21, 2017, Revised Selected Papers 12. Springer International Publishing, 2018.

[37] Binks, Adam. "The art of phishing: past, present and future." Computer Fraud & Security 2019.4 (2019): 9-11.

Digital Twin

[38] Liu, Mengnan, et al. "Review of digital twin about concepts, technologies, and industrial applications." Journal of Manufacturing Systems 58 (2021): 346-361.

[39] Barricelli, Barbara Rita, Elena Casiraghi, and Daniela Fogli. "A survey on digital twin: Definitions, characteristics, applications, and design implications." IEEE access 7 (2019): 167653-167671.

[40] Zheng, Yu, Sen Yang, and Huanchong Cheng. "An application framework of digital twin and its case study." Journal of Ambient Intelligence and Humanized Computing 10 (2019): 1141-1153.

[41] Tao, Fei, et al. "Digital twin-driven product design, manufacturing and service with big data." The International Journal of Advanced Manufacturing Technology 94 (2018): 3563-3576.

[42] Al-Ali, Abdul-Rahman, et al. "Digital twin conceptual model within the context of internet of things." Future Internet 12.10 (2020): 163.

[43] San, Omer. "The digital twin revolution." Nature Computational Science 1.5 (2021): 307-308.

LIST OF PUBLICATIONS

- Amen, K., Zohdy, M., & Mahmoud, M. (2020). Machine Learning for Multiple Stage Heart Disease Prediction. 205–223. https://doi.org/10.5121/csit.2020.101118, 2nd International Conference on Machine Learning & Applications (CMLA 2020), September 26- 27, 2020, Copenhagen, Denmark
- Amen, K., Zohdy, M., & Mahmoud, M. (2021). Towards Comparing Machine Learning Models to Foresee the Stages for heart disease. 33–44. https://doi.org/10.5121/csit.2021.110304, 2nd International Conference on Artificial Intelligence and Big Data (AIBD 2021) March 20-21, 2021, Vienna, Austria
- Amen, K., Zohdy, M., & Mahmoud, M. (2021), Using Deep Learning to Optimize Software Define Radio in Smart City and Health Care, The 17th International Conference on Data Science, July 26-29, 2021, Las Vegas.
- Amen, K., Zohdy, M., & Mahmoud, M. (2021), Using Machine Learning For Predicting URL Phishing Attacks, 2nd International Conference on Robotics, Computer Vision, Intelligent System, October 27-28, 2021 Setubal, Portugal

LIST OF PROJECTS

- Quaternions For Rotations, developed a Quaternions package that can be used to rotate any point in a 3-D space using Python Language.
- SQL Injection Attacks in Android Mobile App, an attacker uses SQL Injection to exploit SQLite query to gain access to pages or critical data that he/she is not authorized to view or use.
- CloudLab Architectures, used to build network architecture locally or remotely such as VM, Interfaces, Gateways and assign private IP and public IP.