

**DEBRE BERHAN UNIVERSITY
COLLEGE OF COMPUTING
DEPARTMENT OF INFORMATION
TECHNOLOGY**



**Developing Stroke Diagnosis and
Treatment Knowledge-based System By
Integrating Expert Knowledge and
Machine Learning**

by
TADESSE KASSU

January 2024

**DEBRE BERHAN UNIVERSITY
COLLEGE OF COMPUTING
DEPARTMENT OF INFORMATION
TECHNOLOGY**

**Developing Stroke Diagnosis and
Treatment Knowledge-based System By
Integrating Expert Knowledge and
Machine Learning**

A thesis submitted to Department of Information
Technology in partial fulfilment of the
requirements for the degree of Master of
Science in Information Technology

by
TADESSE KASSU



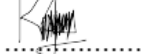
January 2024
DEBRE BERHAN, ETHIOPIA

**DEBRE BERHAN UNIVERSITY
COLLEGE OF COMPUTING
DEPARTMENT OF INFORMATION
TECHNOLOGY**

**Developing Stroke Diagnosis and
Treatment Knowledge-based System By
Integrating Expert Knowledge and
Machine Learning**

by
TADESSE KASSU

Name and Signature of Members of the Examining Board

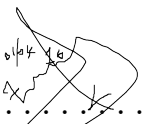
No	Name	Title	Signature	Date
1.	Dr. Sofonias Yitagesu	Advisor		17/01/2024
2	Tibebe Beshah (PhD)	External Examiner		22/01/2024
3	Dr. Kindie Biredagn	Internal Examiner		Jan-25-2024

January 2024

Declaration

I, Author, declare that this thesis "Developing Stroke Diagnosis and Treatment Knowledge-based System By Integrating Expert Knowledge and Machine Learning" is my own work, that it has not been submitted before for any degree or assessment at any other university, and that all the sources I have used or quoted have been indicated and acknowledged by means of complete references.

Tadesse Kassu

Signature:.....

Date: **26/01/2024**

Confirmed by: Dr. Sofonias Yitagesu

Signature:.....

Date: 17/01/2024

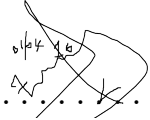
Acknowledgements

First and foremost, I praise to my almighty **GOD** (HEAVENLY FATHER) for his amazing mercy, care and my **HOLY VIRGIN MARY** for giving me strength and hope to complete this thesis work successfully.

I am so glad to express my warm gratitude, heartfelt appreciation to my Advisor Dr. Sofonias Yitagesu for his amazing humble welcoming, timely response, for his sustainable and appreciable guidance, tireless advising, for sharing his knowledge, skill, experience and fine-tuning up to the successful completion of this thesis and thank you for showing me how to become a good person additional to your knowledge and skill.

I wish to express my deepest gratitude to Debre Birhan Referral Hospital workers, Dr. Aklilu Kibru and Sister Asegedeche Tekelu helping to understand in the domain area to extract knowledge about stroke. I would also like to thank Debre Berhan University for financial support and overall facilitation of the research from the beginning until the end. The last but not the least, I would like to thank you all of my friends for your invaluable support and encouragement.

Tadesse Kassu

Signature:.....

Date: **26/01/2024**

Contents

Declaration of Authorship	i
Acknowledgements	ii
List of Figures	vii
List of Tables	ix
Abbreviations	xi
Abstract	xiii
1 INTRODUCTION	1
1.1 BACKGROUND AND MOTIVATION	1
1.2 STATEMENT OF PROBLEM	4
1.3 OBJECTIVE OF THE STUDY	6
1.3.1 General Objective	6
1.3.2 Specific Objectives	6
1.4 SCOPE AND LIMITATION OF THE STUDY	6
1.5 METHODOLOGY OF THE STUDY	7
1.6 SIGNIFICANCE OF THE STUDY	8
1.7 DOCUMENT ORGANIZATION	9
2 LITERATURE REVIEW	10
2.1 OVERVIEW OF STROKE	10
2.1.1 Stroke Diagnosis and Treatment	11
2.2 OVERVIEW OF KNOWLEDGE BASED SYSTEM	11
2.2.1 Types of Knowledge	12
2.2.2 Architecture of knowledge-based system	12
2.2.3 Knowledge Based Reasoning Methods	14
2.2.4 Evaluation of Knowledge-Based System	15
2.3 OVERVIEW OF MACHINE LEARNING	16
2.3.1 Supervised Learning:	16
2.3.2 Unsupervised Learning:	16
2.3.3 Semi-Supervised Learning:	17
2.3.4 Reinforcement Learning:	17
2.4 MACHINE LEARNING APPLICATIONS	17
2.5 MACHINE LEARNING MODELS	18
2.5.1 Decision Tree algorithm	19
2.5.2 Random Forest algorithm	19
2.5.3 Support Vector Machine algorithm	21
2.6 ENCODING AND EMBEDDING IN MACHINE LEARNING	21

2.7	INTEGRATION OF DOMAIN KNOWLEDGE AND MACHINE LEARNING	22
2.8	EVALUATION OF KNOWLEDGE BASE AND MACHINE LEARNING MODEL	23
2.9	RELATED WORKS	24
3	METHODS AND ALGORITHM	30
3.1	PROPOSED RESEARCH METHODOLOGY	30
3.2	DESIGN SCIENCE RESEARCH PROCESS MODEL	30
3.2.1	Problem Identification and motivation	31
3.2.2	Define the Objective for a Solution	31
3.2.3	Design and Development	32
3.2.3.1	Study Population and Sampling Method	32
3.2.3.2	Knowledge Acquisition/Data Collection Methods	32
3.3	Proposed System Framework	34
3.3.1	Data Collection	35
3.3.2	Data Preprocessing	35
3.3.2.1	Detecting Outliers	35
3.3.2.2	Detecting of Noisy Value	35
3.3.2.3	Handling Missing Values	36
3.3.2.4	Encoding and Embedding	37
3.3.2.5	Synthetic Minority Over-sampling Technique	40
3.3.2.6	Tools and Libraries	45
3.3.3	Demonstration	46
3.3.4	Evaluation of Classifier Models	46
3.3.4.1	Accuracy	47
3.3.4.2	True Positive Rate (TPR) and False Positive Rate (FPR)	48
3.3.4.3	Precision and Recall	48
3.3.5	Communication	48
4	DOMAIN UNDERSTANDING AND DATA PREPARATION	50
4.1	PROBLEM DOMAIN UNDERSTANDING	50
4.2	UNDERSTANDING OF THE DATA	54
4.2.1	The raw data descriptions	54
4.3	DATA PREPROCESSING	56
4.3.1	Data visualization	56
4.3.2	Handling missing values	60
4.3.3	Handling imbalance dataset	62
4.3.4	Encoding and Embedding	63
4.3.5	Data Normalization	64
4.3.6	Dataset Splitting	64
5	EXPERIMENTATION	66
5.1	EXPERIMENTAL SETUP	66

5.2	CONSTRUCTING PREDICTIVE MODEL	67
5.2.1	Scenario 1: Experiments with all attributes	67
5.2.1.1	Experiment 1 Decision tree classifier with all attributes under 10-fold	68
5.2.1.2	Experiment 2 Random forest classifier with all attributes under 10-fold	69
5.2.1.3	Experiment 3 Support vector machine classifier with all attributes under 10-fold	70
5.2.2	Scenario 2: Experiments with selected attributes	71
5.2.2.1	Experiment 4 Decision tree classifier with selected attributes under 10-fold	71
5.2.2.2	Experiment 5 Random forest classifier with selected attributes under 10-fold	71
5.2.2.3	Experiment 6 Support vector machine classifier with selected attributes under 10-fold	73
5.2.2.4	Experiment 7 Decision tree classifier with selected attributes	74
5.2.2.5	Experiment 8 Random forest classifier with selected attributes	74
5.2.2.6	Experiment 9 Support vector machine classifier with selected attributes	74
5.3	MODEL COMPARISON AND SELECTION	75
5.3.1	Model Comparison	75
5.3.2	Model Selection	77
5.3.3	Error rate (Mis-classification) of the selected model	77
5.4	RULE EXTRACTION	78
5.4.1	Rule Extraction from Random Forest Classifier with Selected Attributes Under 10-fold Cross Validation Test Option Methods	78
5.5	TESTING MODEL PERFORMANCE	80
6	DOMAIN EXPERT KNOWLEDGE EXTRACTION	82
6.1	KNOWLEDGE ACQUISITION METHOD	82
6.1.1	Expert Interviews:	82
6.1.2	Medical Records Analysis:	82
6.1.3	Clinical Guidelines and Research:	83
6.1.4	Collaboration with Stroke Centers:	83
6.1.5	Observation and Shadowing:	83
6.1.6	Case Studies:	83
6.1.7	Workshops and Conferences:	83
6.2	EXPERT KNOWLEDGE MODELING	83
6.3	EXPERT KNOWLEDGE REPRESENTATION	85
6.3.1	Rules extracted from expert knowledge	85
7	KNOWLEDGE-BASED SYSTEM	87

7.1	MAPPING KNOWLEDGE TO KNOWLEDGE-BASED SYSTEM	87
7.1.1	Structure of Random Forest and Prolog Rule	88
7.2	KNOWLEDGE BASE CONSTRUCTION	90
7.2.1	Knowledge Base	90
7.2.2	Inference Engine	92
7.2.2.1	Prioritized Inferring	94
7.2.3	User Interface	95
7.2.4	Explanation Facility	96
7.3	EVALUATION OF SYSTEM	97
7.3.1	System performance testing using test cases	98
7.3.2	User Acceptance Testing	100
7.4	DISCUSSION AND RESULTS COMPARISON	103
8	CONCLUSION AND FUTURE WORK	106
8.1	CONCLUSION	106
8.2	FUTURE WORK	108
	Bibliography	110
	Appendix I	125
	Appendix II	127
	Appendix III	128
.1	Information about kaggle stroke prediction dataset	128
.2	Sample Prolog code with the syntax of python	128
.3	Sample HTML code for GUI	130

List of Figures

2.1	Architecture for Knowledge-Based System	13
2.2	Rule Based Reasoning System Components and Processes	15
2.3	Basic Parts of Decision Tree classifier	20
2.4	Diagram of random forest classifier	20
2.5	Diagram of support vector machine classifier	21
3.1	Design Science Research Process Model	31
3.2	Framework for Proposed System	34
3.3	confusion matrix	47
4.1	Machine learning process model	51
4.2	numerical variables distribution	57
4.3	categorical variables distribution	57
4.4	proportion of target variables(stroke)	58
4.5	Correlations of variables	59
4.6	Risk level by age	59
4.7	Risk level by BMI	60
4.8	Risk level by avg_glucose_level	60
4.9	'other' value drooped from gender column	60
4.10	Before and after the Skew of variables	61
4.11	Missed value handling	62
4.12	Before and after oversampling	63
4.13	Sample Encoding and Embedding Results	64
5.1	Confusion Matrix of Decision Tree classifier with all attribute under 10-fold	68
5.2	Confusion Matrix of Random forest classifier with all attribute under 10-fold	69
5.3	Support vector machine classifier Confusion Matrix with all attribute under 10-fold	70
5.4	Selected Independent features with correlation to target feature.	72
5.5	Decision tree classifier with selected attributes under 10-fold	72
5.6	Random forest classifier with selected attributes under 10-fold	73
5.7	Support vector machine classifier with selected attributes under 10-fold	73
5.8	Decision tree classifier with selected attributes	74
5.9	Random forest classifier with selected attributes	75
5.10	Support vector machine classifier with selected attributes	75
5.11	Experiment result comparison	77
5.12	Random forest classifier performance with all performance matrix	78
5.13	Model Performance Testing	81

6.1	Decision Tree for stroke diagnosis and treatment acquire from domain expert	84
7.1	User interface of proposed system	96
7.2	Flask Application Linking the Model and Web Page.	97
7.3	Stroke additional Diagnosis facilities page	97
7.4	Stroke treatment facilities page	98
7.5	Performance evaluation of proposer system with test case	99
7.6	Performance of developed system	100
7.7	User acceptance criteria with their corresponding answer	102

List of Tables

2.1	Approaches category of reviewed work.	27
2.2	Data type of reviewed work.	28
3.1	Expert’s profile on Stroke diagnosis and treatment.	33
4.1	Selected attributes from the business domain.	54
4.2	Stroke dataset description.	56
5.1	Experimental parameter with separate value	67
5.2	performance result of Decision Tree classifier with all attribute under 10-fold.	68
5.3	performance result of Random forest classifier with all attribute under 10-fold.	69
5.4	performance result of Support Vector Machine classifier with all attribute under 10-fold.	70
5.5	Performance Comparison of experimental results in accuracy	76
7.1	Attributes, comparison operators, and values in knowledge mapping	88
7.2	Performance Comparison of previous studies in accuracy	105

List of Algorithms

1	Inter Quantile Range(IQR) outliers detecting algorithm	36
2	Procedure: Calculate Noise Factor(string FUNC, dataset X)	37
3	Imputing algorithm for missing value handle	38
4	One-Hot Encoding	38
5	word embedding with Continuous Bag-of-Words(CBOW)	40
6	SMOTE algorithm	41
7	Pseudocode of Decision Tree Algorithm	42
8	Pseudo code for the random forest algorithm	43
9	Pseudocode for the SVM algorithm	44
10	pseudocode for Prioritized Inference algorithm	95

Abbreviations

AI	Artificial Intelligence
AVI	Artariovenous Malformation
BMI	Body Mass Index
CDC	Center for Disease Control and Prevention
CDS	Clinical Decision Support
CLI	Command Line Interface
CPU	Central Processing Unit
CSV	Comma Separated Values
CT	Computerized Tomography
DALY	Disability Adjusted Life year
DBRH	Debre Birhan Referral Hospital
DNN	Deep Neural Network
DS	Design Science
DSRM	Design Science Research Methodology
DT	Decision Tree
DTC	Decision Tree Classifier
ECG/EKG	Electrocardiogram
EEG	Electroencephalogram
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GBD	Global Burden of Disease
GUI	Graphics User Interface
GP	General Practitioner
HO	Health Officer
HTML	HyperText Markup Language
KBS	Knowledge Base System
KNHDS	Korean National Hospital Discharge Survey

ML	M achine L earning
MRI	M agnetic R esonance I maging
NSAID	N on- S teroidal A nti- I nflammatory D rugs
ONCHIT	O ffice of the N ational C oordinator for H ealth I nformation T echnology
PCA	P rincipal C omponent A nalysis
RAM	R andom A ccess M emory
RBR	R ule B ased R easoning
RF	R andom F orest
RFC	R andom F orest C lassifier
ROC	R eceiver O perating C haracteristic
SDAT	S troke D iagnosis A nd T reatment
SMOTE	S ynthetic M inority O versampling T echnique
SVM	S upport V ector M achines
SVMC	S upport V ector M achines C lassifier
SWI-PROLOG	S teering W heel I nterface P rogramming in L ogic
TIA	T ransient I schemic A ttack
TN	T rue N egative
TP	T rue P ositive
TPR	T rue P ositive R ate
US	U nited S tate
WHO	W orld H ealth O rganization

Abstract

Stroke is a medical condition where the blood arteries in the brain rupture, causing brain damage. This interruption of blood flow can lead to the development of symptoms. According to the World Health Organization (WHO), stroke is the leading cause of death and disability worldwide. Recognizing the warning signs of a stroke early on can help lessen the severity of the condition.

To the best of the author's knowledge, no research has been conducted on creating a knowledge-based system that combines expert knowledge and machine learning prediction to assist healthcare workers in managing stroke without the need for experts. This study aims to develop such a system, referred to as a knowledge-based system (KBS), for stroke diagnosis and treatment by integrating machine learning prediction with expert knowledge.

Data was gathered from Debre Birhan Referral Hospital (DBRH) using in-depth interviews with experts selected through purposive sampling techniques and from a public dataset obtained from the Keggale website. The dataset comprised 5,110 instances, 12 attributes, and 2 class labels. To balance the class labels, a Synthetic Minority Over-sampling Technique (SMOTE) was utilized, increasing the number of instances from 5,110 to 9,720 for experimentation.

This research utilized the Design Science Research Methodology. Expert knowledge was extracted and represented using production rules, which were then modeled using a decision tree. To identify the most suitable machine learning classifier models, 9 experiments were conducted with a decision tree, random forest, and support vector machine classifiers, employing 10-fold cross-validation and the percentage split test option in two scenarios: one using all attributes and another using selected attributes. Finally, the rules of the random forest classifier with the selected attributes achieved the best performance, with an accuracy of 99%, and were integrated with expert knowledge to develop the knowledge-based system.

The collaboration of two programming languages, Python and Prolog, was employed to develop the knowledge-based system. The rule base was constructed using the Prolog programming language and SWI-Prolog 7.6.4, while HTML and

a sublime text editor were used to create the system's graphical user interface (GUI). The developed knowledge-based system was evaluated by preparing test cases and user acceptance testing, achieving a performance rating of 95% and a user acceptance score of 88%. Therefore, the knowledge-based system successfully fulfills its intended purpose without needing experts.

Keywords: Stroke, Machine learning techniques, Domain Expert Knowledge, Knowledge-based system.

Chapter 1

INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

Stroke is a common non-communicable disease affecting public health in both developed and developing countries [1]. It's a cerebrovascular disease affecting blood vessels, causing brain damage. Immediate medical attention is crucial to prevent permanent damage or death. The Global Burden of Diseases, Injuries, and Risk Factors Study (GBD) 2019 [2] revealed 12.2 million (95%) incidents and 101 million prevalent strokes, 143 million disability-adjusted life years (DALYs) due to Stroke, and 6.55 million deaths worldwide, making Stroke the second-leading cause of death (11.6% of total deaths) after ischemic heart disease (16.2%) [3].

Stroke burden has increased by 33.5 million globally due to population growth, aging, and modifiable risk factors such as hypertension, diabetes, dietary risks, impaired glucose intolerance, obesity, smoking, air pollution, alcohol use, hyperlipidemia, and physical inactivity with an annual mortality rate of 5.5 million [3, 4]. Fewer women (2.6 million) die from Stroke, especially in lower-income countries, with 86.0% of deaths and 89.0% of DALYs.

The American Heart Association and the National Institutes of Health [5] release annual reports on heart disease, Stroke, and cardiovascular risk factors. In 2020, heart disease and Stroke caused more deaths annually than cancer and chronic lower respiratory disease combined. The number of deaths due to cardiovascular disease in 2019 represented 33% of all global deaths, with ischaemic heart disease (9.1 million deaths) and Stroke (6.6 million deaths) totaling 85% of all cardiovascular disease deaths worldwide. While the number of deaths due to cardiovascular disease over the last 30 years has increased globally in large part due to an aging and growing population, the age-standardized death rate has declined by one-third

from 354.5 deaths per 100,000 people in 1990 to 239.9 deaths per 100,000 people in 2019 [6].

In 2021, the Clinical Decision Support System (CDS) [7] reported that strokes are responsible for 1 out of 6 deaths related to cardiovascular disease. In the United States [8], strokes occur approximately every 40 seconds. Annually, over 795,000 individuals have a stroke, with 610,000 being new cases and 185,000 being individuals who have previously had a stroke [8].

Strokes are medical emergencies that fall into five main categories: Ischemic Stroke, Transient Ischemic Attack (TIA), Hemorrhagic Stroke, Brain Stem Stroke, and Cryptogenic Stroke. Ischemic Stroke occurs when a blood vessel blocks the blood supply to the brain and is the most common type of stroke. TIAs are temporary blockages in blood flow to the brain, while Hemorrhagic Stroke happens when bleeding in the brain damages nearby cells. Brain Stem Stroke affects both sides of the body and can result in paralysis. The last type, cryptogenic stroke, refers to strokes where the cause cannot be determined [8, 9].

Strokes are becoming a global concern due to population growth, aging, and increased risk factors, especially in low and middle-income countries. Major risk factors for strokes include hypertension, diabetes, heart conditions, smoking, age, gender, race, family history, brain aneurysms, arteriovenous malformations (AVMs), alcohol and illegal drug use, certain medical conditions, vacuities, bleeding disorders, lack of physical activity, obesity, stress, depression, unhealthy cholesterol levels, diet, and the use of non-steroidal anti-inflammatory drugs (NSAIDs) [3].

Feigin et al. [10] suggest that over 90% of strokes are due to modifiable risk factors, and controlling these could prevent over three-quarters of the global stroke burden. Factors influencing health risk reduction include threat severity, cost-effective interventions, societal values, culture, and preferences. Identifying and managing risk factors is crucial for preventing diseases or injuries. Stroke prevention is achieved through a healthy lifestyle, controlling body mass index (BMI), and maintaining heart and kidney health. Predicting and treating stroke is crucial to prevent long-term damage or death. Diagnosis strategies rely on doctors'

experience, and long-term diagnoses can increase the doctor's weakness.

Machine learning [11], a sub-field of artificial intelligence, can be used in healthcare to detect and predict strokes using domain experts' knowledge. This technique helps medical professionals manage clinical data and improve patient outcomes by providing medical insights that were previously unavailable. Applications of machine learning in healthcare include disease prediction, visualization of biomedical data, improved diagnosis, more accurate health records, and AI-assisted surgery [12]. Its goal is to validate doctors' decisions through predictive algorithms.

Knowledge-based systems (KBS) [13] utilize different knowledge sources to address intricate problems, such as Stroke, with the help of artificial intelligence. These systems assist in human decision-making and learning. In the medical field, healthcare organizations are increasingly depending on Knowledge-based systems and automation to enhance operational efficiency, minimize costs, and uphold quality. The applications of Knowledge-based systems have a direct influence on the quality of healthcare services [14], making them an essential tool for researchers and practitioners.

A Clinical Decision Support system is a solution used nowadays to tackle the complexity involved in medical diagnosis and treatments. It can significantly improve the quality, safety, efficiency, and effectiveness of healthcare. The Office of the National Coordinator for Health Information Technology (ONCHIT) supports the development, adoption, implementation, and evaluation of CDS to enhance healthcare decision-making [15].

One promising approach is to combine computer technology and artificial intelligence in healthcare services to offer competitive services that patients value. Knowledge-based systems and Machine Learning serve as a sub-field of Artificial Intelligence that focuses on effective decision-making. They achieve this by imitating the behavior of human experts within a specific and well-defined area of knowledge through various techniques and algorithms [16].

Medical emergencies like Stroke impose a significant global burden. The integration of computer technology and artificial intelligence in healthcare services plays a direct role in the quality of healthcare provided. This motivates researchers to explore the development of a knowledge-based system for Stroke diagnosis and treatment. Such a system would merge expert knowledge with a machine learning predictive model.

In this thesis, we have developed a knowledge-based system for Stroke diagnosis and treatment. Our approach involves integrating domain expert knowledge with a machine learning predictive model. By combining these two elements, we aim to create a more comprehensive and accurate system that can assist healthcare professionals in making informed decisions when dealing with Stroke cases.

1.2 STATEMENT OF PROBLEM

The World Health Organization [17] defines a stroke as a brain accident causing rapid clinical signs of cerebral function disturbance, lasting 24 hours or longer, or leading to death, involving cerebral infarction, intracerebral hemorrhage, and subarachnoid hemorrhage.

Global Stroke Statistics in 2019 [2] shows variation in stroke incidence, case-fatality, and mortality rates among countries with high burdens in sub-Saharan Africa. Hospital-based Stroke unit reporting is limited, and long-term mortality data is scarce. The one-month pooled Stroke case-fatality rate was 24.1% and 33.2% at one year, with high heterogeneity at three and five years. High Stroke case fatalities over one month are attributed to weak healthcare systems and vascular risk factors [18].

Research on Stroke has focused on assessing its severity [19, 20], socioeconomic status, incidence, prevalence, mortality, and worldwide burden [21]. It has also explored expert-system-based medical stroke prevention [22], integrating data mining results with knowledge-based systems for diagnosis and treatment recommendations [23], data mining techniques for Stroke length of stay prediction [24], and machine learning for stroke diagnosis and outcome prediction [25, 26].

Researchers have identified significant gaps in the integration of computer technology and artificial intelligence. Samuel et al. [23] developed a prototype integrating data mining results with a knowledge-based system for stroke diagnosis and treatment recommendation, using rule-based reasoning and Naive Bayes classifier. However, the main limitation is the challenging knowledge-acquisition process in the medical domain.

Pardamean et al. [22] developed an expert-system-based medical Stroke prevention system using an inference engine to match facts with domains knowledge to determine Stroke risk levels. However, the system focuses on inferring to determine risk levels for prevention. Adoukonou et al. [18] suggest smart systems determine Stroke length using data mining techniques and model-based systems but lack prioritization based on expert and data knowledge.

Mainali et al. [25] indicates that machine learning is effective for rapid clinical decision-making but requires clinical expert oversight to address specific aspects not accounted for in automated algorithms.

This study aims to use machine learning techniques with domain expert knowledge for Stroke disease diagnosis and treatment. The proposed method learns directly from data without a predetermined model, using logic as the knowledge base. The research aims to answer specific research questions.

- What are the determinant factors used to determine the Stroke?
- Which machine learning algorithm best designs the predictive Stroke diagnosis and treatment model?
- How do we represent the acquired knowledge from the machine learning algorithm and domain expert for developing the knowledge-based systems?
- How could a machine learning predictive model integrated with expert knowledge for developing a knowledge-based system for diagnosis and treatment of Stroke?

1.3 OBJECTIVE OF THE STUDY

1.3.1 General Objective

The overall objective of this study is to construct a knowledge-based system for Stroke diagnosis and treatment by integrating knowledge acquired from domain experts and machine learning to increase the effectiveness and efficiency of the system.

1.3.2 Specific Objectives

In this study, the following specific objectives will be achieved.

- To identify the determinant factors that determine Stroke diagnosis and treatment.
- To design a machine learning algorithm that predicts Stroke for diagnosis and treatment.
- To compare the models generated by machine learning algorithms.
- To represent the knowledge acquired from the predictive model and domain expert suitable for developing the knowledge-based systems.
- To integrate a machine learning predictive model with expert knowledge for developing a knowledge-based system for diagnosis and treatment of Stroke.
- To evaluate the proposed system in predicting, diagnosis and treatment of Stroke.

1.4 SCOPE AND LIMITATION OF THE STUDY

The scope of the proposed system covers designing an intelligent integrated model for Stroke diagnosis and treatment by considering factors that highly determined the stroke. The machine learning part will be limited on computational learning machine and back-propagation. The knowledge of the knowledge-based system acquired from the domain expert's is from interview and document analysis. This study is not identifying the type of drugs that the patient will take and it have not use local labeled dataset for model learning and validation in machine learning.

1.5 METHODOLOGY OF THE STUDY

The research will follow the design science research methodology (DSRM) to perform the study. As stated by Peferrs et al. [27], DSRM follows six elements. Each of these is described below.

A. Problem identification and motivation. The starting of any research is defining of the exact research problem to be researched to get appropriate solution [28]. It can be explained in terms of problem statement and this already done in the previous section of this chapter. Since the problem definition will be used to develop an effective artifact solution.

B. Objectives of a solution. The objectives should be inferred rationally from the problem specification [27]. The objective of the solution in this research will provide suitable prediction for the case of Stroke.

C. Design and development. This element of the DSRM will help to create the artifact solution. The solution can be constructs, models, methods, or instantiates. This activity includes determining the artifact's desired functionality and its architecture and then creating the actual artifact [27]. The proposed framework that shows the design and development part will be presented in the third chapter of this research work.

D. Demonstration. This involves experimentation analysis and result section based on the proposed framework [27]. The experimental analysis and result part will be presented in the fifth chapter of this document.

E. Evaluation. The evaluation observes and measures how well the artifact supports a solution to the problem [27]. This activity involves comparing the objectives of a solution to actual observed results from use of the artifact in the demonstration. To evaluate the performance of the proposed machine learning technique confusion matrix will be used. This confusion matrix performance metrics like recall, precision and accuracy will be calculated. The experimental analysis and result part that will be presented in the fifth chapter show the demonstration

and evaluation elements of the DSRM elements. Besides this it will show the performance comparison of each proposed machine learning techniques.

F. Communication. It is the final element the DSRM. Communicating the problem and its importance, the artifact, its utility and novelty, the rigor of its design, and its effectiveness to the stakeholders like medical researchers, Information science researchers in the stream of data analysis, other relevant audiences, such as medical experts, will validate the each and every step to be followed. Communication is not only from the experts, but also what has been done by the researches in the past with the related works. Such communication will be covered in the Literature Review Chapter. Besides this, to make this research work available, it will be presented for department internal and external examiner's and also will be published.

1.6 SIGNIFICANCE OF THE STUDY

Knowledge-Based Systems and Machine Learning's are artificial intelligent tools working in a specific domain to provide advice and consultation in decision making [29]. The proper utilization of artificial intelligent tools increase productivity and enhances problem-solving capacity. The general benefit of this study is providing advice on diagnosis and treatment of Stroke for health workers in health institutions where there are no enough specialists. Hence, developing and implementing integrated computer based system helps to make effective decision in diagnosis and treatment of Stroke and this gives benefit directly or indirectly as listed below :- The direct beneficiaries of this research output are those health workers like interns, General practitioners (GP), Health officers (HO), Residents, etc. helps to Stroke diagnosis and treatment in hospitals and health centers.

- To improve timely diagnostic services and the accessibility of services for enhancing health workers capacity in fields of Stroke management.
- To cure or prolong the life of Stroke patients and ensure the best possible quality of life.
- This study can give hands on experience for the researcher for understanding studies in the future.

In general, this study is supposed to have many advantages due to the effectiveness, non-subjectivity and efficiency for Stroke diagnosis and treatment and help to reduce morbidity and mortality with the case of Stroke, reduce economic burden and create awareness and help as second opinion for specialists.

1.7 DOCUMENT ORGANIZATION

This thesis composed of eight chapters. As presented here the first chapter consists of Introduction section that holds background of the study, statement of the problem, objective of the study and proposed research methodology.

The second chapter will present the Literature review part. This chapter will describe about the global burden statistics, the prevalence, severity, mortality and risk factors of Stroke and about integrate computer technology and artificial intelligence into health services to provide competitive healthcare services.

In addition to this; it will give highlight of the machine learning techniques and show the algorithms of the selected computing technique. At last, this chapter will present the related research work done by others.

The proposed research design section will be presented in the third chapter. This chapter will start by showing the proposed research methodology and system framework; then explain each component of the framework. The business and data understanding and data preprocessing will be presented in the fourth chapter. The fifth chapter will present the experimental analysis and result of the proposed computing technique for the learning from the dataset. The sixth chapter will indicate knowledge extraction for comprehensive knowledge-based systems involves extracting knowledge from domain experts. The integration of knowledge and all about knowledge-based system will explored in the seventh chapter. Finally Conclusion, and future work of the research work will be presented in the eighth chapter.

Chapter 2

LITERATURE REVIEW

2.1 OVERVIEW OF STROKE

A Stroke occurs when a blood vessel that carries oxygen and nutrients to the brain is either blocked by a clot or bursts (or ruptures) [30]. When that happens, part of the brain cannot get the blood (and oxygen) it needs, so it and brain cells die. The Stroke can happen to anyone at any time and it is a medical emergency that requires urgent medical attention. Early detection and appropriate management are required to prevent further damage to the affected area of the brain and other complications in other parts of the body. World Health Organization (WHO) [31] estimates that fifteen million people worldwide suffer from strokes each year, with one person dying every four to five minutes in the affected population. Stroke is the sixth leading cause of mortality in the United States according to the Centers for Disease Control and Prevention (CDC).

Anyone can have a Stroke at any age. But your chance of having a Stroke increases if you have certain risk factors. Some risk factors for Stroke can be changed or managed, while others can't. Risk factors for Stroke that can be changed, treated, or medically managed are: High blood pressure, Heart disease, Diabetes, Smoking, Birth control pills, History of TIAs (transient ischemic attacks), High red blood cell count, High blood cholesterol and lipids, Lack of exercise, Obesity, Excessive alcohol use, Illegal drugs, Cardiac structural abnormalities, etc. but Risk factors for Stroke that can't be changed are: Older age, Race, Gender, History of prior Stroke, Heredity or genetics, etc. even Other risk factors include are: Where you live, Temperature, season, and climate, Social and economic factors, etc. Generally 80% of strokes are preventable because of most of Stroke risk factories are changed, treated, or medically managed [32].

As Stroke is an emergency situation; Stroke symptoms may happen suddenly and each person's symptoms may vary. Symptoms of Stroke may include [33]: Weakness or numbness of the face, arm, or leg, usually on one side of the body, Having trouble speaking or understanding, Problems with vision, such as dimness or loss of vision in one or both eyes, Dizziness or problems with balance or coordination, Problems with movement or walking, Fainting (loss of consciousness) or seizure, Severe headaches with no known cause, especially if they happen suddenly, etc.

2.1.1 Stroke Diagnosis and Treatment

According to Kuriakose et al. [34] the first step in assessing a Stroke patient is to determine whether the patient is experiencing an ischemic or hemorrhagic Stroke because treatment depends on the type of Stroke; so that the correct treatment can begin. Strokes are usually diagnosed by doing physical tests and studying images of the brain produced during a scan. A Computed tomography (CT) scan or magnetic resonance imaging (MRI) of the head is typically the first test performed.

To help Stroke patient: determine the type, location and cause of a Stroke and to rule out other disorders physicians may use: Blood tests, Electrocardiogram (ECG, EKG), Carotid ultrasound or Doppler ultrasound and cerebral angiography. Immediate treatment can save lives and reduce disability. Treatment is focus on restoring blood flow for an ischemic Stroke and on controlling bleeding and reducing pressure on the brain in a hemorrhagic Stroke. This is done by using different medication drugs as a type of Stroke.

2.2 OVERVIEW OF KNOWLEDGE BASED SYSTEM

A knowledge-based system (KBS) is a system that uses artificial intelligence (AI) to solve problems. It consists of a repository of expert knowledge with utilities designed to facilitate the knowledge retrieval in response to specific queries, along with learning and justification. Knowledge-Based System focus on using knowledge-based techniques to support human decision making, learning, and action [35].

Knowledge-based systems are computer programs designed to solve problems, generate new information (such as a diagnosis), or provide advice, using a knowledge base and an inference mechanism. Most systems include a user interface and some explanation capability. Knowledge-based systems are characterized as focusing on the accumulation, representation and use of knowledge specific to a particular task [29, 35].

2.2.1 Types of Knowledge

In business and knowledge management, two types of knowledge are usually defined, namely explicit and tacit knowledge [36].

Tacit knowledge is knowledge in the human brain. This kind of knowledge is difficult to transfer to another person using writing it down or verbalizing [37].

Explicit knowledge is knowledge that can be readily articulated, codified, accessed and verbalized. It can be easily transmitted to others. Most forms of explicit knowledge can be stored in certain media [38].

2.2.2 Architecture of knowledge-based system

Architecture is the blueprint of the system that describes the structure of an object and guideline of the system. System architecture is the conceptual model that describes the structure, behavior and view of the system [39]. The architecture of the knowledge-based system includes a knowledge base, inference engine, user interface and explanation facility. Figure 2.1 shows the general architecture of the knowledge-based system adopted from [40].

Knowledge base: contains the knowledge necessary for understanding, formulating and solving problems. It is a warehouse of the domain-specific knowledge captured from the human expert via the knowledge acquisition module. To represent the knowledge production rules, frames, logic, semantic net, etc. are used [41]. The knowledge base stores all relevant information, data, rules, cases, and relationships used by the expert system. A knowledge base can combine the knowledge of multiple human experts [42].

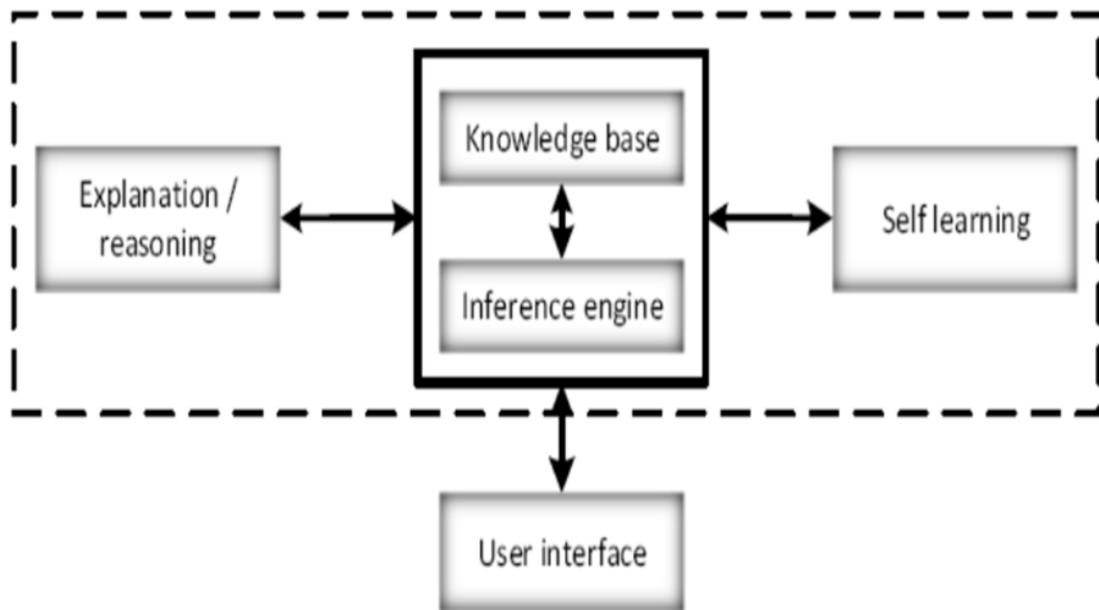


FIGURE 2.1: Architecture for Knowledge-Based System

Inference Engine: is a brain of expert systems. It uses the control structure (rule interpreter) and provides a methodology for reasoning. It acts as an interpreter which analyzes and processes the rules. It is used to perform the task of matching antecedents from the responses given by the users and firing rules. The major task of the inference engine is to trace its way through a forest of rules to conclude [43]. The purpose of the inference engine is to seek information and relationships from the knowledge base and to provide answers, detection, and suggestions in the way a human expert would. The inference engine must find the right facts, interpretations, rules and assemble them correctly [43].

User Interface: - is the interaction point between the user and the system. The user interface can be graphical user interface (GUI) or command line interface (CLI).

Explanation facility: - it provides information to user for the questions asked by the system. This facility is helpful to have clarity while answering the questions asked by the system [41]. Here user would like to ask the basic questions why and how and serves as a tutor in sharing the systems knowledge with the user.

2.2.3 Knowledge Based Reasoning Methods

There are a number of knowledge based reasoning methods. The well-known reasoning approaches are ontology based reasoning, semantic network, neural network, fuzzy logic, case based reasoning and rule based reasoning [41]. For the purpose of this research rule based reasoning approach is used discuss below.

Rule-Based Reasoning Method: Rule based reasoning (RBR) is a system whose knowledge representation contains a set of rules and facts [44]. It is the most common form of knowledge representation technique. Most rule-based representation today uses a variant of the general production system model in production which are represented as "IF...THEN" rules, where the "IF" part is the condition to be satisfied while the "THEN" part is the action to be executed [41]. The term rules represent what to do or not to do while certain conditions are satisfied. Similarly, domain knowledge is represented by a set of rules [45]. The general form of rules based system can be illustrated as follows:

‘IF’ First premise, AND Second premise, AND ... THEN Conclusion

This is semantically the same as a Prolog rule:

Conclusion \longrightarrow First premise, Second premise

The knowledge base contains the domain knowledge pertinent to the problem and the solution in general found by incrementally exploring the rule and process formed by the rules in the knowledge base. Figure 2.2 shows rule base reasoning component and process adopted from [46].

There are two main inference methods in rule based reasoning mechanism. These are backward chaining and forward chaining [47].

Forward chaining: it is the inference engines first predetermine the criterion and the next steps are to add the criterion one at a time, until the entire chain has been trained. With data driven control, facts in the system are represented in a working memory which is continually updated. The system first check to find all

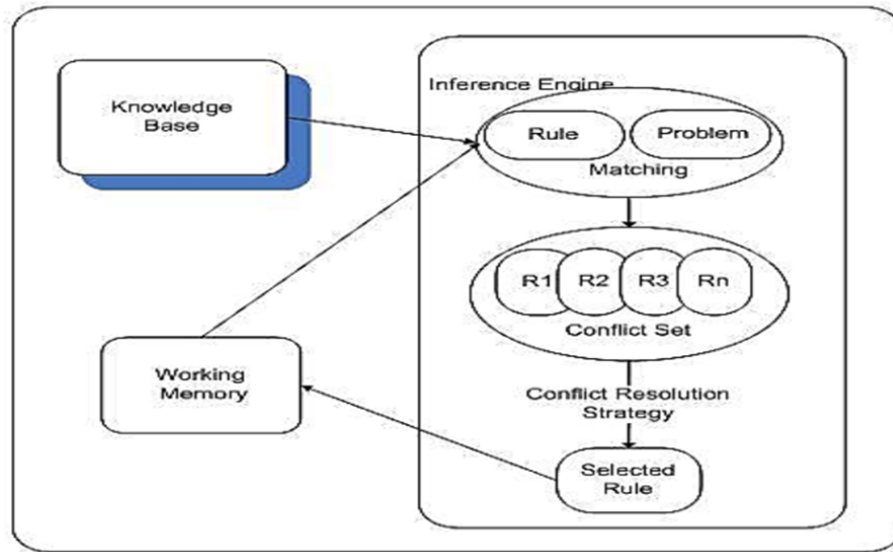


FIGURE 2.2: Rule Based Reasoning System Components and Processes

the rules whose condition holds true. Both data driven and goal driven chaining method follows the same procedures. However, the difference lies on the inference process [48].

Backward chaining: It is similar with forward chaining the difference is it receives the problem description as a set of conclusions instead of conditions and tries to find the premises that cause the conclusion. Given a goal state and then the system try to prove if the goal matches with the initial facts. When a match is found goal is succeeded. But, if it doesn't then the inference engine start to check the next rules whose conclusions (previously referred to as actions) match with the given fact. Goal driven control is commonly known as top-down or backward chaining [48, 49].

2.2.4 Evaluation of Knowledge-Based System

Evaluation is an iterative process of systematic assessment of knowledge based system. The evaluation process carried out at different stage of system development life cycle. The performance of the system was assessed or measured through quantitative and qualitative techniques to achieve the expected objective [50]. In this study both classifier models obtained from machine learning and developed KBS is evaluated in appropriate measurements.

2.3 OVERVIEW OF MACHINE LEARNING

Machine learning is a particular application of Artificial Intelligence (AI) that provides machines with the ability to automatically learn and improve from experience without being explicitly programmed [51]. The first machine learning idea was raised by Alan Turing in the 1950s [52]. The focus of machine learning is learning, that is, acquiring skills or knowledge from experience. Most commonly, this means synthesizing useful concepts from historical data. As such, there are many different types of learning that may encounter as a practitioner in the field of machine learning: from whole fields of study to specific techniques. Machine Learning is mainly divided into four categories: Supervised learning, unsupervised learning, Semi-supervised learning, and Reinforcement learning [53].

2.3.1 Supervised Learning:

Describes a class of problem that involves using a model to learn a mapping between input examples and the target variable [54]. Applications in which the training data comprises examples of the input vectors along with their corresponding target vectors are known as supervised learning problems.

There are two main types of supervised learning problems: they are classification that involves predicting a class label [55] and regression that involves predicting a numerical value [56]. Both classification and regression problems; may have one or more input variables and input variables may be any data type, such as numerical or categorical.

2.3.2 Unsupervised Learning:

Describes a class of problems that involves using a model to describe or extract relationships in data. Compared to supervised learning, unsupervised learning operates upon only the input data without outputs or target variables [57]. As such, unsupervised learning does not have a teacher correcting the model, as in the case of supervised learning. There are many types of unsupervised learning, although there are two main problems that are often encountered by a practitioner:

they are clustering that involves finding groups in the data and density estimation that involves summarizing the distribution of data.

2.3.3 Semi-Supervised Learning:

It is supervised learning where the training data contains very few labeled examples and a large number of unlabeled examples. The goal of a semi-supervised learning model is to make effective use of all of the available data, not just the labeled data like in supervised learning [58].

2.3.4 Reinforcement Learning:

It is learning what to do and how to map situations to actions so as to maximize a numerical reward signal. The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them [59].

The evolution of machine learning has continuously changed with the growth of technology starting from 1950s . Machine learning is used to make the systems learn from data by identifying patterns and making decisions with the minimal human intervention [60]. By using efficient algorithms and fast analysis machine learning algorithms produce accurate results. Machine learning can be used in various fields like finance, health, government, retail, transportation, etc. In health-care and life science, machine learning is used for disease identification, disease diagnosis and treatment, risk prediction and risk management [61].

2.4 MACHINE LEARNING APPLICATIONS

Machine learning is used to create algorithms based on historical data and relationships between data. Machine learning applications are used to design and develop a model of prediction issues, semantics analysis, natural language processing, information retrieval, image processing, etc. [62].

As input more data into the machine helps the algorithms to teach the computer. We cannot apply the machine learning model directly to real-world data. To teach the computer machine learning algorithms use training data and predict unknown data using machine learning algorithms. The various types of machine learning

algorithms that are used for various purposes like data mining, predictive analytic, image processing etc. has also presented in the comprehensive review [62].

2.5 MACHINE LEARNING MODELS

Machine Learning starts within artificial intelligence which is a rapidly developing multidisciplinary research area that aims to solve increasingly complex problems. In today's highly integrated world and promised to become a robust and powerful means for obtaining solutions to previously unsolved problems. Various machine learning algorithms include classification analysis, regression analysis, data clustering, association rule learning, feature engineering for dimensional reduction, and deep learning methods [63]. A general structure of a machine learning-based predictive model is trained from historical data and the outcome is generated from the new test data.

Classification algorithm: is regarded as a supervised learning method in machine learning, referring to a problem of predictive modeling as well, where a class label is predicted [64]. Mathematically, it maps a function (F) from input variables (X) to output variables (Y) as target, label, or categories. Predict the class of given data points, it can be carried out on structured or unstructured data. Common classification problems (tasks):-

- **Binary classification:** It refers to the classification tasks having two class labels [65] such as “true and false”, “yes and no”, “male or female”, etc. In such binary classification tasks, one class could be the normal state, while an abnormal state could be another class. For instance, “Stroke not detected” is the normal state of a task that involves a medical test, and “Stroke detected” could be considered as the abnormal state, spam detection such as “spam” and “not spam” in email service providers, etc. are considered as binary classification.
- **Multi-class classification:** Traditionally, this refers to those classification tasks having more than two class labels [66]. The multi-class classification does not have the principle of normal and abnormal outcomes, unlike binary classification tasks.

- **Multi-label classification:** In machine learning, multi label classification is an important consideration where an example is associated with several classes or labels [67]. It is a generalization of multi-class classification where the classes involved in the problem are hierarchically structured, and each example may simultaneously belong to more than one class at each hierarchical level.

Generally, the most common and popular classification algorithms used widely in this thesis work are Decision Tree, Random Forest, and Support Vector Machine.

2.5.1 Decision Tree algorithm

Decision Tree algorithm [68] belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a decision tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data) .

In Decision Trees [69], for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node. Decision Tree algorithms results are quick and precise. Nodes and the leaves represent the entities. The output is represented in leaves and the data observation is represented in nodes. Decision Tree classifications are simple to read and realize. The following Figure 2.4 shows important terminology related to Decision Trees.

2.5.2 Random Forest algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in machine learning. But however, it is mainly used for classification problems. It is based on the concept of ensemble learning, which is a process

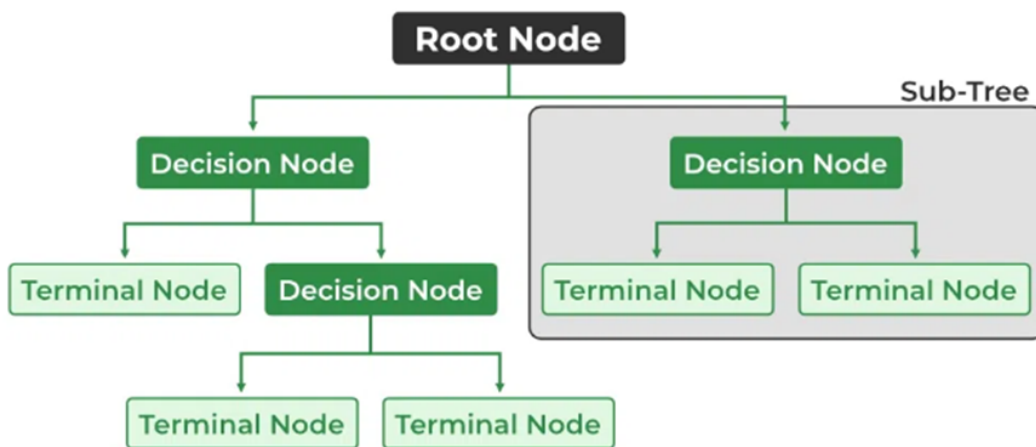


FIGURE 2.3: Basic Parts of Decision Tree classifier

of combining multiple classifiers to solve a complex problem and to improve the performance of the model [70].

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." [71]. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of over fitting. The following Figure 2.5 shows Diagram of random forest classifier.

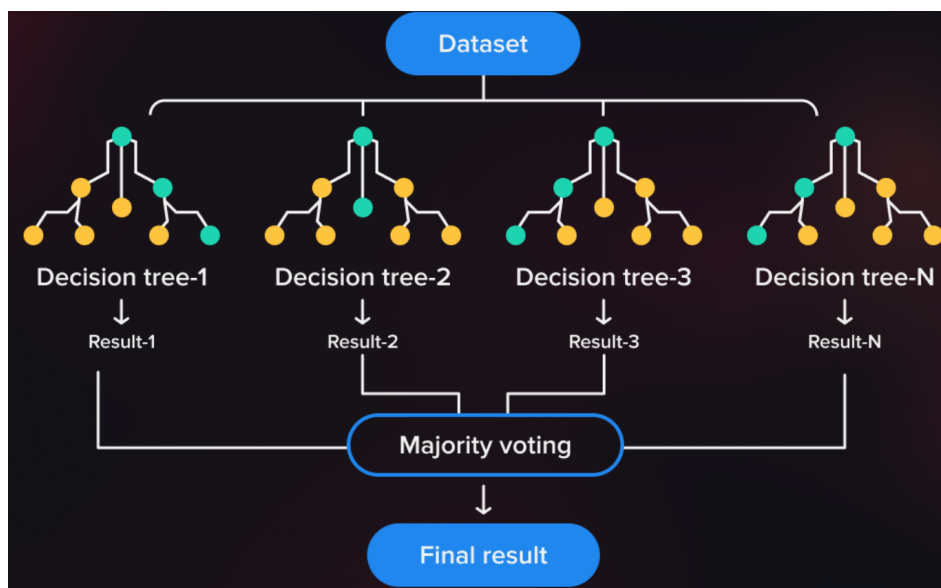


FIGURE 2.4: Diagram of random forest classifier

2.5.3 Support Vector Machine algorithm

Support Vector Machine (SVM) is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyper plane [72].

Support Vector Machine chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below figure 2.6 diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

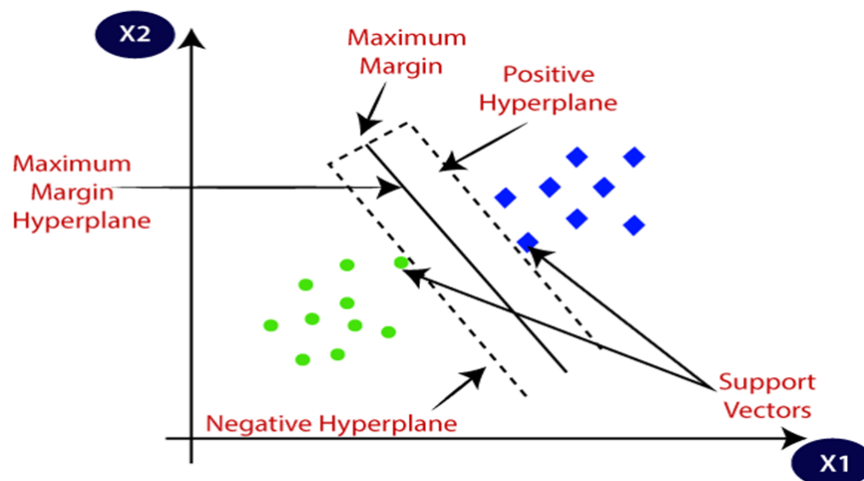


FIGURE 2.5: Diagram of support vector machine classifier

2.6 ENCODING AND EMBEDDING IN MACHINE LEARNING

The success of any machine learning model depends heavily on the quality of the training data that is used to develop it [73]. High-quality training data is often considered to be the most critical factor in achieving accurate and reliable machine learning results. This quality of data can be improved by implemented encoding

and embedding. Encoding and embedding are related concepts in machine learning, but they serve different purposes.

Encoding refers to the process of converting data into a numerical representation that can be fed into a machine learning model. This can involve techniques such as one-hot encoding, binary encoding, or more complex transformations such as Principal Component Analysis (PCA) or t-Distributed Stochastic Neighbor Embedding (t-SNE). The goal of encoding is to transform the data into a format that can be easily processed by a machine learning algorithm.

Embedding, on the other hand, refers to the process of mapping words, phrases, or other high-dimensional data into a lower-dimensional vector space. In this space, similar items are mapped to nearby points, allowing the model to capture their relationships and patterns. Embedding are often used in natural language processing (NLP) tasks such as text classification, sentiment analysis, and machine translation.

In summary, encoding is the process of converting data into a numerical representation, while embedding is the process of mapping high-dimensional data into a lower-dimensional vector space to capture their relationships and patterns.

2.7 INTEGRATION OF DOMAIN KNOWLEDGE AND MACHINE LEARNING

Machine learning has been heavily researched and widely used in many areas. The success is grounded in its powerful capability to learn from a tremendous amount of data. However, it is still far from achieving intelligence comparable to humans. As of today, there have been few reports on artificial intelligence defeating humans in sensory tasks such as image recognition, object detection, or language translation. Some skills are not acquired by machines at all, such as creativity, imagination, and critical thinking. Integrating human knowledge into machine learning can significantly reduce the data required, increase the reliability and robustness of machine learning, and build explainable machine learning systems.

Healthcare is one of potential applications of integrating domain expert knowledge with Machine learning that can be used to analyze medical data and identify

patterns and relationships that can improve patient outcomes. Domain experts can provide valuable insights into the data and help ensure that the machine learning models are accurate and effective.

There are several ways to integrate domain expert knowledge with ML:

- **Feature engineering:** Domain experts can provide valuable insights into the data and help identify relevant features that can improve the performance of ML algorithms.
- **Labeling data:** Domain experts can help label the data, which is essential for training ML models. High-quality labeled data can significantly improve the accuracy of ML models.
- **Model selection:** Domain experts can help select the most appropriate ML algorithm for a specific problem, based on their knowledge of the domain and the characteristics of the data.
- **Model interpretation:** Domain experts can interpret the results of ML models and provide insights into the underlying mechanisms and relationships in the data.
- **Hybrid approaches:** Domain experts can work with ML practitioners to develop hybrid approaches that combine the strengths of both domain expertise and ML algorithms

2.8 EVALUATION OF KNOWLEDGE BASE AND MACHINE LEARNING MODEL

Evaluation is an iterative process of systematic assessment of system. The evaluation process carried out at different stage of system development life cycle. The performance of the system was assessed or measured through quantitative and qualitative techniques to achieve the expected objective [74]. In this study both classifier models obtained from machine learning and developed knowledge-based system is evaluated in appropriate measurements (see chapter 3).

2.9 RELATED WORKS

Researchers worldwide have conducted various studies in the past few decades on predicting Stroke disease. These studies involve the use of data mining, knowledge base systems, machine learning, and deep learning algorithms. Here are a few examples related to the discussed problem:

Cheon S et al. [75] conducted a study using 2013–2016 Korean National Hospital Discharge In-depth Injury Survey (KNHDS) data to identify factors that contribute to Stroke mortality. They developed a predictive model using deep learning techniques and analyzed a total of 15,099 Stroke patients with 11 variables. The researchers employed a combination of deep neural networks (DNN) and scaled principal component analysis (PCA) to automatically extract features from the data and determine the risk factors for Stroke. The data was split into training (66%) and testing sets (34%), with 30% of the samples in the training set used for validation. The deep learning models utilized simple feed-forward neural networks with four hidden layers and were trained using a standard back-propagation algorithm. Considering all the model parameters, the researchers found that the optimal Stroke probability threshold was 0.13. The model exhibited a sensitivity (Sn) of 64.32%, specificity (Sp) of 85.56%, positive predictive value (PPV) of 25.7%, accuracy (Acc) of 84.03%, and an area under the receiver operating characteristic curve (AUC) of 83.48%. However this research work has no mechanism to incorporate the engagement of domain expertise. It automatically extracts features that can determine the risk factor of Stroke using only Deep learning techniques.

Zhang S et al. [76] conducted a study on automatically diagnosing and preventing ischemic Stroke. The researchers collaborated with two local Grade III A hospitals. They collected 5,668 brain MRI randomly selected from 2017 to 2019, along with clinical imaging reports from 300 cases. Professional neurologists accurately labeled all the lesion areas. To detect lesions in MRI images automatically, three object detection networks were designed and implemented: Faster R-CNN, YOLOv3, and SSD. The accuracy of lesion detection (mAP) was similar for Faster

R-CNN (VGG-16), Faster R-CNN (ResNet-101), and YOLOv3, at 74.9%. However, SSD performed the best, achieving an accuracy of 89.77%. In this study the researcher use experts only for labeling MRI but their detail knowledge about ischemic Stroke is mandatory for better diagnosing and preventing ischemic Stroke.

Tazin T et al. [31] examined the efficiency of different machine learning algorithms in accurately predicting Stroke based on various physiological variables. The researchers utilized a Stroke prediction dataset from the Kaggle website, which included 5110 patient records and 12 attributes. Their findings reveal that the random forest classification method outperformed other tested approaches, achieving a classification accuracy of 96% when using cross-validation metrics for forecasting brain Stroke. However in this study the researcher focus only to examine the efficiency of different machine learning algorithms for predicting Stroke but in addition to machine learning prediction domain expert knowledge is essential for Stroke prediction as well as best diagnosis and treatment.

Sailasya G and Kumari GLA [77] conducted a study to demonstrate how different machine learning algorithms can accurately predict Stroke based on various physiological attributes. They used a dataset from Kaggle, consisting of 5110 rows and 12 columns. The classification algorithms utilized in this study included Logistic Regression, Decision Tree Classification, Random Forest Classification, K-Nearest Neighbors Classification, Support Vector Machine, and Naïve Bayes Classification. Amongst these algorithms, Naïve Bayes Classification proved to be the most effective, achieving an accuracy rate of 82%. In this study only predicting stroke with machine learning algorithm is the concern but integrating this machine learning prediction with domain expert knowledge provide better health services.

Thammaboosadee S and Kansadub T [78] presented a Stroke risk prediction model that uses three datasets. These datasets include demographic data, medical screening data, and their combined application using three classification algorithms: Naive Bayes, Decision Tree, and Artificial Neural Network (ANN). The model's performance is evaluated based on accuracy, AUC, FPR, FNR, and the ten-fold cross-validation method. The experiment results show that the best

model is the ANN with integrated data, achieving an accuracy of 84%, an FP rate of 12%, an FN rate of 25%, and an AUC of 90%. In this study stroke risk prediction is determined only by considering different types of dataset but domain expert insight also should be incorporated.

The research by **Almadani O and Alshammari R** [79] aims to achieve two main objectives. Firstly, it aims to use data mining techniques to predict the patients who are at risk of experiencing a Stroke. Secondly, it aims to identify the patient with the highest likelihood of developing a Stroke. To accomplish this, three classification algorithms (C4.5, Jrip, and multi-layer perceptron) were employed on Stroke patient data sets collected from National Guard hospitals in three cities within the Kingdom of Saudi Arabia.

The data set used in the research was extracted from King Abdulaziz Medical City (KAMC). It comprised a total of 969 instances, with 69 instances classified as Stroke mimics and 899 instances classified as Stroke patients. The data set contained 1,004 attributes. After applying Principal Component Analysis (PCA) on the Stroke data, a comparison of the data mining algorithms revealed that C4.5 achieved the highest accuracy on the test data set, reaching 95.25%. The researchers are focused only to identify risk of experiencing and likelihood of developing a Stroke with a certain collected dataset but also domain expert knowledge is mandatory for better health system.

Alberto J et al. [80] conducted research using data science and machine learning techniques to develop an accurate Stroke outcome prediction model. They utilized a dataset from Kaggle, consisting of information on 5110 individuals and 10 distinct features. The dataset was divided into 75% for model training and 25% for testing and evaluating the model. Various machine learning algorithms were employed during the training process, ultimately demonstrating that the Random Forest algorithm performed the best. The model achieved an accuracy rate of 92%, an F-score of 92%, and an AUC of 96%. This study also focus on predicting stroke based on dataset by using machine learning technique but engaging domain expert knowledge can help to more effective and efficient the health system.

Chun M et al. [81] compared Cox and ML models to predict Stroke risk in China over different follow-up periods (i.e., within 9 years, 0-3 years, 3-6 years, 6-9 years). They aimed to determine when ML models perform better than traditional Cox-based approaches and to create an ensemble model combining both methods to identify high-risk individuals. The China Kadoorie Biobank (CKB) conducted a prospective cohort study with 512,726 participants aged 35-74. The participants were randomly divided into a training set (85%), a validation set (12.75%), and a test set (2.25%). Using the training set, Cox, random survival forest (RSF), logistic regression (LR), support vector machine (SVM), gradient boosted tree (GBT), and multilayer perceptron (MLP) models were derived to predict Stroke risk. The ensemble model showed slight improvements in identifying high-risk individuals, with Accuracy at 80%, specificity at 81%, and PPV at 24% compared to either Cox or GBT models alone. However in addition to ensemble Cox and ML models to predict Stroke risk integrating domain expert knowledge to machine learning technique helps to have a system effective and efficient performance.

To conclude related work and gaps, several studies have been conducted in case of Stroke as discussed above. Most of the study is conducted in predicting Stroke patient mortality, Stroke lesion detection and analysis, Analyzing the performance of Stroke prediction, Stroke disease detection and prediction , Stroke prediction and Stroke risk prediction. All these previous study focused merely on using data mining classification techniques, machine learning algorithms, and deep learning approaches. As summary the following Table 2.1 shows the category of approaches for reviewed work.

No	Approach	Reviewed work
1	Machine learning	[31, 77, 78, 80, 81]
2	Deep learning	[75, 76]
3	Data mining	[79]

TABLE 2.1: Approaches category of reviewed work.

As clearly seen from the reviewed work in their approach domain expert knowledge are not incorporated. Incorporating domain expert knowledge to Knowledge-based

system is crucial to make more informed decisions and provide more accurate results. Additionally, domain expert knowledge can help the system to identify and address specific needs and challenges within a particular domain, leading to more targeted and effective solutions.

To detecting, predicting and analyzing Stroke the researcher used different type of data with different variables. Most of them used Stroke prediction dataset from the Kaggle website, which included 5110 patient records and 12 attributes and others used from health sectors. But the data has the form of text and image. The following Table 2.2 shows type of data utilized in previous research work.

No	Type of data	Reviewed work	Source
1	Text	[31, 77, 78, 80] [75, 79, 81]	from the Kaggle website
2	Image	[76]	two local Grade III A hospitals (from health sectors)

TABLE 2.2: Data type of reviewed work.

Based on the nature of data the researcher should implement different data preprocessing techniques to ensure that the data is clean, consistent, and well-prepared for modeling. But in the reviewed work their is no more techniques of data preprocessing for Improved data quality, Better model performance, Reduced noise, and Increased interpretability.

Some of related work with the use of domain knowledge to Stroke diagnosis and treatment are basically to using the method of developing ontology with subsequent activities: specifications of requirements, domain knowledge acquisition, conceptualization and formalization of conceptual model followed by evaluation such as:

Marczykowska T et al. [82] The paper presents the preliminary results of the

research aiming at the development of the Diagnostic Stroke Ontology. The described machine processable knowledge representation was designed for the applications supporting stroke diagnosis and management of stroke patients. Presented in the article structure and content of Stroke Diagnostic Ontology (DStrokeOnto ontology) include salient ontology modules, classes in modules, relations between modules, main classes definition and details of ontology metrics and expressively.

Bandrowski A et al. [83] One of the critical components for the overall NIF system, the NIF Standardized Ontologies (NIFSTD), provides an extensive collection of standard neuroscience concepts along with their synonyms and relationships. The knowledge models defined in the NIFSTD ontologies enable an effective concept-based search over heterogeneous types of web-accessible information entities in NIF's production system. NIFSTD covers major domains in neuroscience, including diseases, brain anatomy, cell types, sub-cellular anatomy, small molecules, techniques, and resource descriptors

Generally the main limitations that found from the reviewed paper are lack of domain expert knowledge and more data preprocessing technique to improve the accuracy, interpretability, and generalization of the model. As a result, this study aims to address this gap by developing a Stroke diagnosis and treatment system based on both machine learning prediction models and domain expert knowledge. Also more data preprocessing techniques like parameter tuning and feature selection, and encoding and embedding methods in machine learning prediction are implemented to be more efficient and accurate model. The intention is to support healthcare professionals in making informed and consistent clinical decisions.

Chapter 3

METHODS AND ALGORITHM

3.1 PROPOSED RESEARCH METHODOLOGY

The study uses the Design Science Research Methodology (DSRM) [84], an iterative process that focuses on creating and evaluating artifacts like models, frameworks, and systems to tackle real-world problems, with the aim of developing innovative solutions that advance theory and practice.

Design science focuses on improving problem context through the design and investigation of artifacts, with the goal of creating practical solutions that advance theory and practice. Researchers [85, 86] believe truth and utility are inseparable, with truth informing design and utility informing theory. Research assessment can identify weaknesses in theories or artifacts, requiring refinement and reassessment.

Design Science Research Methodology is a tool used to generate design science knowledge in an area of interest, resulting in design theories, constructs, models, methods, and abstracts [84]. The knowledge base, composed of foundations and methodologies, serves as the raw materials for research. Rigor is achieved by applying existing methodologies. In this study, health workers from Debre Birhan Referral Hospital are involved, and the knowledge base is assessed through experiments and parameter variations.

3.2 DESIGN SCIENCE RESEARCH PROCESS MODEL

Design Science Research Methodology Process Model (DSRMPM) [87] used to study organizational problems. As shown in Figure 3.1. [88], Design Science Research Methodology typically consists of the following stages [88]: It begins with problem identification, defining the problem, setting objectives, designing and developing an artifact, demonstrating functionality, and evaluating performance. It

involves reflection, learning, and documentation, contributing to existing knowledge in the field through research papers, presentations, or technical reports.

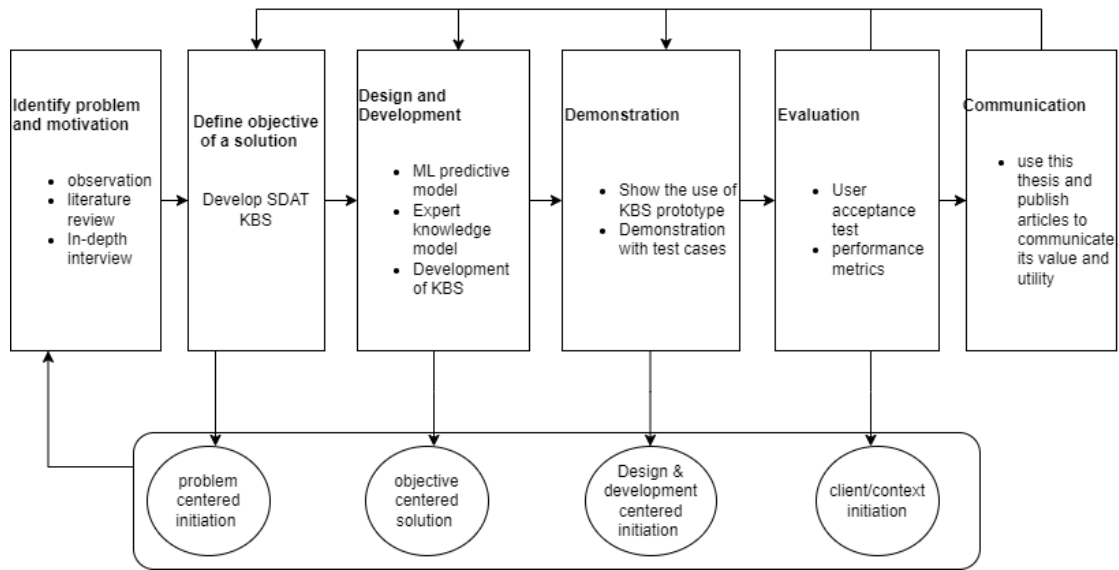


FIGURE 3.1: Design Science Research Process Model

3.2.1 Problem Identification and motivation

Problem Identification involves identifying and defining the problem or opportunity that the research aims to address. It includes understanding the context, stakeholders, and objectives of the research. It involves defining the research problem and justifying its value. Problem identification is achieved through literature review, in-depth interviews, and document analysis. The study aims to address the increasing Stroke morbidity and mortality due to a lack of medical facilities and awareness. Factors motivating this include previous research gaps, increased patient numbers, and delayed diagnosis.

3.2.2 Define the Objective for a Solution

Once the research problem is understood, the objective of the study is crucial for setting the foundation for subsequent activities. The objective of the study is to develop a Knowledge-Based System (KBS) for Stroke diagnosis and treatment, integrating machine learning prediction with expert knowledge to support solutions to the problem specification, thereby assisting health workers in their tasks. This rational inference from the problem specification is essential for successfully implementing the Knowledge-Based System .

3.2.3 Design and Development

The design and development stage involves creating an artifact, such as a model, framework, prototype, or tangible solution, to address a problem and fulfill defined requirements. This step is based on existing knowledge and theoretical concepts. Key activities include finalizing a knowledge-based system, planning and implementing experiments, determining desired functionality and architecture, and creating the actual artifact [88]. The research focuses on data sources, knowledge modeling and representation, framework, and implementation tools for a Stroke diagnosis and treatment Knowledge-based system (SDAT-KBS).

3.2.3.1 Study Population and Sampling Method

The primary data have been collected from DBRH, where Stroke patients obtain medical services. These have been considered the main sources of data because of direct involvement in the implementation of knowledge base development from domain experts to develop predictive models and select the best model for Knowledge-Based System. Secondary data was retrieved from the Kaggle website healthcare dataset [89] and documents (guidelines).

3.2.3.2 Knowledge Acquisition/Data Collection Methods

The study utilized various data collection methods, including interviews, questionnaires, observation, document analysis, and knowledge discovery techniques, to acquire knowledge, including machine learning techniques, literature reviews, and datasets.

In-depth Interview: The researcher conducted in-depth interviews with experts from Debre Birhan referral hospital to gather relevant knowledge on Stroke diagnosis and treatment, aiming to improve existing practices and challenges through suggestions for potential solutions.

This study utilized purposive sampling to gather detailed knowledge about Stroke from practitioners. The interviews were conducted by experts at DBRH, who directly interact with patients, allowing for purposive data collection. The DBRH

was chosen due to its service to Stroke patients, ensuring representative views and the ability to apply findings to the general population.

A total of 3 experts participated in the interview for this study; these experts were interviewed about their background knowledge to understand the domain area very well and to identify determinant risk factors for Stroke diagnosis and treatment. The interview was conducted through face-to-face interaction and called using their phone number frequently if any unclear ideas Stroke in-depth interview. Sample interview questions are provided in the Appendix. Table 3.1 displays the expert's profile on Stroke diagnosis and treatment.

No	Educational level	Specialization	Organization
1	General practitioner	Neurologist	DBRH
2	General practitioner	Oncologist	DBRH
3	General practitioner	—	DBRH

TABLE 3.1: Expert's profile on Stroke diagnosis and treatment.

Literature Review: In this study, the researcher conducted a thorough review of reputable research on Stroke, knowledge base application, and machine learning in the health sector, focusing on Stroke diagnosis and treatment, to identify gaps and formulate the study's problem.

Knowledge Acquired from machine learning: Machine learning and knowledge acquisition are complementary approaches to knowledge acquisition and organization. Machine learning develops autonomous algorithms for data acquisition and knowledge compilation, while knowledge acquisition improves and partially automates knowledge acquisition from human experts. Both fields are moving towards an integrated approach, with machine learning techniques automating knowledge acquisition from experts and knowledge acquisition techniques guiding the learning process [90]. In health sectors, increasing data storage necessitates automatic knowledge extraction techniques, especially when domain experts are busy and manual acquisition is time-consuming.

3.3 Proposed System Framework

The framework involves three phases: dataset knowledge extraction, domain expert knowledge extraction, and a knowledge-based system integrating machine learning and expert knowledge, as illustrated in Figure 3.2.

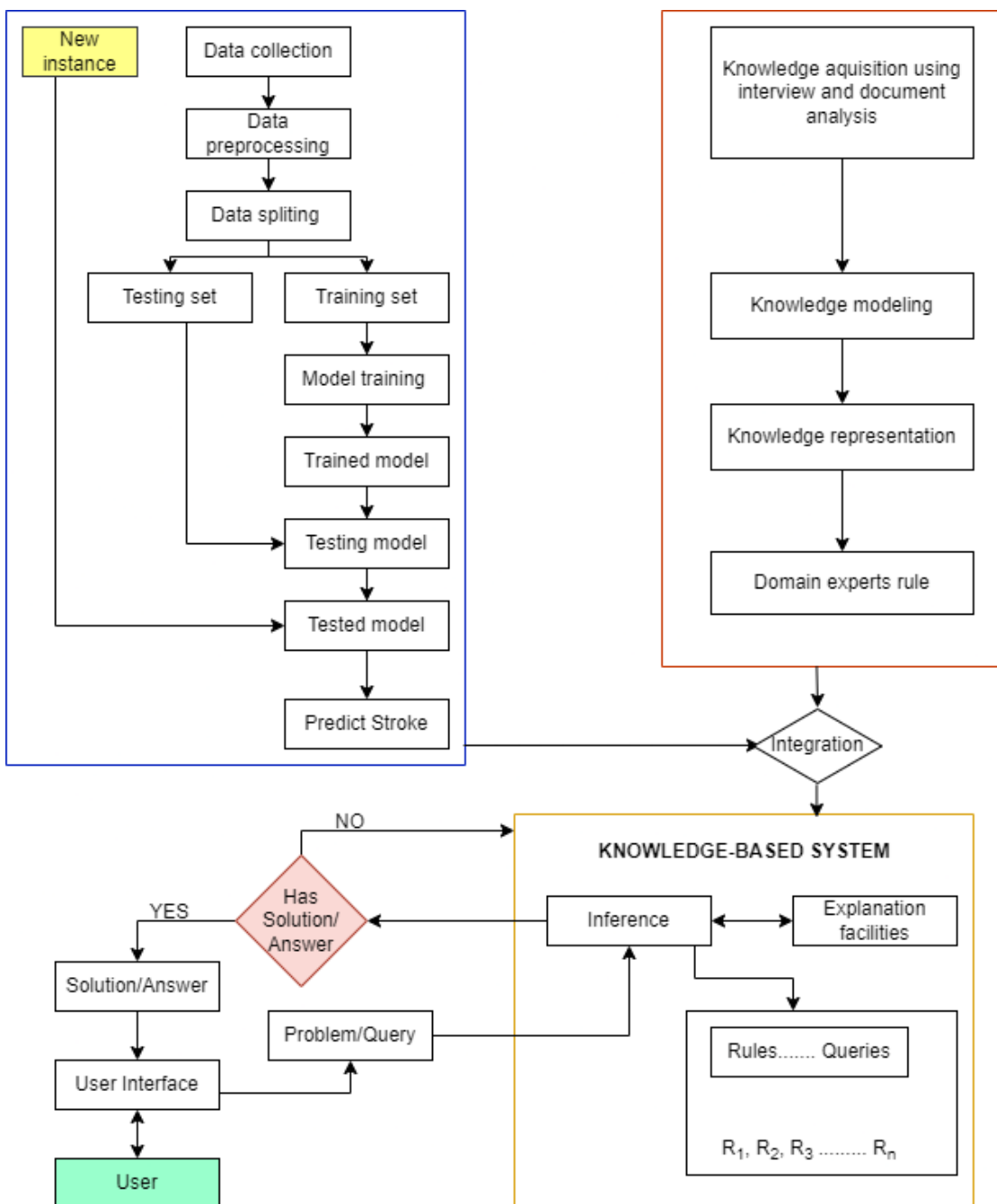


FIGURE 3.2: Framework for Proposed System

3.3.1 Data Collection

This study utilized Keggles website datasets for pre-processing and predicting Stroke for appropriate diagnosis and treatment, details of which will be discussed in Chapter 4.

3.3.2 Data Preprocessing

The prepared data for machine learning need a preprocessing to get cleaned dataset, in this research work, five preprocessing tasks will be performed to get cleaned dataset that become suitable input for machine learning. These are detecting of outliers, removal of noisy values, handling of missing value, normalization the attribute value and balancing of sample size for each corresponding class labels. Description of how each data preprocessing task will be performed presented below.

3.3.2.1 Detecting Outliers

Data points that are significantly different from the majority of the data may happen due to data entry error, measuring instrument error, communication barrier and etc. In this study, instances that have extremely high or extremely low value detected as having outliers will be rejected from the dataset. The logical step to detecting outliers using the Inter Quantile Range(IQR) is shown below.

3.3.2.2 Detecting of Noisy Value

Data points that do not accurately represent the true value of the variable can be caused by errors in data entry, measurement, or transmission, and can have a significant impact on the accuracy of any analysis or modeling that is performed on the data. Detecting this noisy values in a dataset is an important task in data preprocessing and data cleaning. In this work, the simplest and most effective ways to identify noisy data, QcleanNOISE algorithm [91] is implemented. Thus, for the i -th instance, we compute

$$Q_{-i} = (r_{-i} - d_{-i}) / \max(d_{-i}, r_{-i}) \quad (3.1)$$

Algorithm 1 Inter Quantile Range(IQR) outliers detecting algorithm

```

1: Input: Dataset before IQR (D); dataset after IQR( $D_i$ )
2: Where ( $D_i$ ) =  $\emptyset$ 
3: Process Logic
4: Sort the dataset in ascending order
5: Calculate the 1st and 3rd quartiles(Q1, Q3)
6: while Q1 = 25% of (D) and Q3 = 75% of (D) do
7:   Compute IQR(D)
8:   IQR = Q3 - Q1
9:   Compute lower and upper bound
10:   $lower\_bound = (Q1 + 1.5) * IQR$ 
11:   $upper\_bound = (Q3 + 1.5) * IQR$ 
12:  if i in (D) then
13:    If  $i < lower\_bound$  or  $i > upper\_bound$ 
14:  end if
15: end while
16: Output: Data Outliers( $D_i$ )

```

where d_i is the distance of the i -th instance to the centroid of its class and r_i is the minimum distance of the i -th instance to the centroid of the classes where does not belong to. Each instance with $Q < 0$ is regarded as a candidate to be a noisy instance. The next step is to find out the k nearest neighbors to each noisy instance candidate. Finally, we count the number of neighbors that belong to the same class. If the majority of the neighbors does not belong to the same class the candidate instance is then regarded as noisy and discarded from the training set. The logical step to detecting Noisy Value in this research work are as follows:

3.3.2.3 Handling Missing Values

As missing values can have a significant impact on the accuracy and reliability of the model handling missing values is an important aspect of machine learning. In this research work, the missing values of each feature will be either rejected or imputed using mean values and most frequent values for numerical and nominal data type respectively. Rejection of instances will be done if three or more features have missing values in that corresponding instance; whereas if the missing value is two or less, it will be imputed using most frequent value or mean value of the corresponding features. The logical step to handle missing value [92] is shown below.

Algorithm 2 Procedure: Calculate Noise Factor(string FUNC, dataset X)

```

1: input: Dataset  $X = [x_{ij}]_{n \times m}$  with  $n$  observations and  $m$  attributes, where  $x_{ij}$ 
   is the value of the  $j^{\text{th}}$  attribute for the  $i^{\text{th}}$  observation.  $x_{*j}$  denotes the  $j^{\text{th}}$ 
   attribute.
2: output: Noise Factor  $S_i$  for instance  $i, 1 \leq i \leq n$ .
3:  $s_{ikj} = 0 \forall i = 1, \dots, n$  and  $j, k = 1, \dots, m$  // initialize values
4: for  $j = 1$  to  $m$ 
5:   Transform attribute  $x_{*j}$  into the partitioned attribute  $\hat{x}_{*j} \in \{0, \dots, L - 1\}$ 
   where  $L = \#$  of bins
6:   for  $k = 1$  to  $m$ 
7:     if  $k \neq j$ 
8:       calculate Mean ( $x_{*k} \mid \hat{x}_{*j} = l$ ) and Std ( $x_{*k} \mid \hat{x}_{*j} = l$ ) for  $l = 0, \dots, L - 1$ 
9:       for  $i = 1$  to  $n$ 
10:         $s_{ikj} = |x_{ik} - \text{Mean}(x_{*k} \mid \hat{x}_{*j} = \hat{x}_{ij})| / \text{Std}(x_{*k} \mid \hat{x}_{*j} = \hat{x}_{ij})$ 
11:        //  $s_{ikj}$  is the standardized value of the attribute value  $x_{ik}$  for the  $i^{\text{th}}$ 
        observation given the partitioned attribute value  $\hat{x}_{ij}$ 
12:      endfor
13:    endif
14:  endfor
15: endfor
16: if FUNC = ' SUM '
17:    $S_i = \sum_{k=1}^m \sum_{j=1}^m s_{ikj}$ 
18: endif
19: elseif FUNC = ' MAX '
20:    $S_i = \max_{\{j=1, \dots, m\} \{k=1, \dots, m\}} \{s_{ikj}\}$ 
21: endif
22: return  $S_i$ 

```

3.3.2.4 Encoding and Embedding

When working with data, especially in the fields of data science and machine learning, one of the initial challenges faced is the representation of categorical or non-numeric data. Encoding and embedding are two different ways to represent data in a machine learning model. Both techniques are used to convert categorical data into numerical representations that can be processed by the model.

As choosing the right encoding technique is crucial and depends on the nature of the data (nominal or ordinal) and the specific use case; in this researcher work One-hot encoding is selected.

In **One-Hot Encoding**, each of a categorical variable is represented using a binary vector. This means that if a categorical column has "N" unique values, it will

Algorithm 3 Imputing algorithm for missing value handle

```

1: Input: Dataset before Imputing (D); Imputed dataset ( $D_i$ )
2: Where ( $D_i$ ) =  $\emptyset$ 
3: Process Logic
4: For each attribute x of (D)
5: Step 1: Identify the missing value
6: if no missing value in the attribute, then
7:     go to step 4
8: else go to step 2
9: end if
10: Step 2:
11: if missing value in instance > 10% (> 2features), reject that instance then
12:     Go to step 1
13: else go to Step 3
14: end if
15: Step 3: Identify data type of attribute
16: if data type is nominal then
17:     Identify the most frequent value of the attribute
18:     Impute the missing value with the most frequent value
19: else Go to step 4
20: end if
21: Calculate the mean value of the attribute
22: Impute the missing value with the mean value
23: Go to step 4
24: Step 4: Insert the attribute to the prepared  $S = \leftarrow X$ 
25: Output: Imputed dataset ( $D_i$ )

```

result in "N" new binary columns. Each of these columns will have a value of "1" for the rows where the categorical column matches the respective category and "0" otherwise. Pseudocode for One-Hot encoding process in this research work are:

Algorithm 4 One-Hot Encoding

```

1: Input: Original masks with integer labels
2: Output: Masks with one-hot encoded labels
3: function ENCODE_LABELS(original_masks)
4:     Determine size of one-hot vectors with maximum value of integer label
5: for mask in original_masks
6:     encode integer label to one-hot vector
7:     Repeat step 4 for all integer labels in mask
8: end for
9:     return One-Hot encoded masks
10: end function

```

The primary drawback of One-Hot encoding is increase in dimensionality, especially if the categorical variable has many unique categories. This can lead to the "curse of dimensionality", making the dataset more sparse and harder to work with, potentially slowing down learning algorithms and requiring more memory. To solve this gape in this research word embedding is implemented.

An Embedding is a mathematical representation of a set of data points in a lower-dimensional space that captures their underlying relationships and patterns. Embedding is often used to represent complex data types, such as images, text, or audio, in a way that machine learning algorithms can easily process.

This embedding improve the performance of machine learning algorithms on text data by capturing the meaning and context of words in a way that is more accurate than traditional bag-of-words representations, Better handling of out-of-vocabulary words, provide more interpret-able results than traditional bag-of-words representations by showing the relationships between words in a visual space, and efficient computation.

From different type of embedding in machine learning and artificial intelligence; Word embedding is used in this research that used to represent words as vectors in a low-dimensional space. These embedding capture the meaning and relationships between words, allowing machine learning models to better understand and process.

Word2Vec is one of the most common technique of Word embedding used in this research. It can represent the data precisely in the embedding space with a larger dataset by Continuous Bag-of-Words Model(CBOW). A Continuous Bag-of-Words model basically takes "n" words before and after the target word (wt), and predicts the latter. Where, n can be any number. Mathematically it can be determined as follows:

$$\frac{1}{T} \sum_{t=1}^T \log P(W_t | \sum_{-c \leq j \leq c, j \neq 0} W_{t+j}) \quad (3.2)$$

To calculate the probability of context word given the center word each word represented by two sets of vectors, \mathbf{Uw} and \mathbf{Vw} . We will use \mathbf{Uw} when \mathbf{w} is the

context word and $\mathbf{V}\mathbf{w}$ when \mathbf{w} is the center word. Using these two vectors, our probability equation for center word \mathbf{o} and context word \mathbf{c} will look like this:

$$p(C = c|O = o) = \frac{\exp(U_o^T V_c)}{\sum_{w \in V_{ocab}} \exp(U_o^T V_w)} \quad (3.3)$$

Basic steps involved in the process of word embedding with Continuous Bag-of-Words are:

Algorithm 5 word embedding with Continuous Bag-of-Words(CBOW)

```

1: Initialize feature vector bg feature = [0, 0, . . . .0]
2: for token in text.tokenize() do
3:   if token in dict then
4:     token idx = getindex(dict, token)
5:     bg feature[token idx] + +
6:   else
7:     continue
8:   Endif
9: Endfor
10: return bg feature

```

3.3.2.5 Synthetic Minority Over-sampling Technique

SMOTE (Synthetic Minority Over-sampling Technique) is a popular imbalanced data handling technique used to balance the dataset when there is a significant difference in the class distribution. It is commonly used for classification problems where one class has a significantly larger number of instances than the other.

The goal of SMOTE is to create synthetic samples of the minority class by interpolating new instances between existing minority class samples. The idea is to increase the number of minority class samples in the dataset, which can help the model to learn more about the minority class and improve its performance on that class. This SOMTE works as follows that adopted from [93]:

Machine learning: This study aims to create a predictive model for Stroke diagnosis and treatment using various machine learning algorithms like Decision trees, Random Forests, and Support Vector Machines (see Chapter 2). The model's are evaluated and compared before being deployed. Furthermore, the study aims to

Algorithm 6 SMOTE algorithm

```

1: Input:  $X$  : Minority data
2:    $n$  : number of instances of the Minority data
3:    $p$  : Oversample percentage
4:    $K$  : Num of nearest neighbors
5: Output:  $S$  : Synthetic data samples.
6:   for each  $i$  in  $n$  do
7:     Find the  $K$  nearest Neighbors of  $x_i$ 
8:     while  $p! = 0$  do
9:       Select one of the  $k$  nearest neighbors of  $x_i$ 
10:      Select random number between 0 and 1 :  $\text{rand}(0, 1)$ 
11:       $X_s = X_i + \text{random}(0, 1) * (X'_i - X_i)$ 
12:      Add  $X_s$  to  $S$   $p = p - 1$ 
13:     Endwhile
14:   Endfor
15:   Return  $S$ 

```

integrate expert knowledge with the machine learning model for the development of Knowledge-Based System .

❶ **Decision Tree Algorithm :** In machine learning, Classification is a two-step process, learning step and prediction step [94]. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data. Decision Tree is one of the easiest and popular classification algorithms to understand and interpret.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. CART (Classification And Regression Tree) [95] is one of a decision tree algorithm variation that use Gini Impurity in the process of splitting the dataset into a decision tree. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, the purity of the node increases with respect to the target variable. Mathematically, Gini Impurity can be determined as following :

$$I_Gini = 1 - \sum_{i=1}^j P_i^2 \quad (3.4)$$

where j is the number of classes present in the node and P is the distribution of the class in the node.

The Gini Index considers a binary split for each attribute. We can compute a weighted sum of the impurity of each partition. If a binary split on attribute **A** partitions data **D** into **D1** and **D2**, the Gini index of **D** is:

$$Gini_A(D) = \frac{|D1|}{|D|}Gini(D1) + \frac{|D2|}{|D|}Gini(2) \quad (3.5)$$

In the case of a discrete-valued attribute, the subset that gives the minimum gini index for that chosen is selected as a splitting attribute. In the case of continuous-valued attributes, the strategy is to select each pair of adjacent values as a possible split point, and a point with a smaller gini index is chosen as the splitting point.

$$\Delta Gini(A) = Gini(D) - Gini_A(D) \quad (3.6)$$

The attribute with the minimum Gini index is chosen as the splitting attribute.

In this approach, we create different training set for each of the decision trees to train on. It is done by randomly sampling the training set, so there would be repeated training set. Then we train decision tree on the new training samples and this process repeats until we get **B** number of decision trees. The final prediction is based on output of these decision trees.

Algorithm 7 Pseudocode of Decision Tree Algorithm

```

1: GenDecTree(Sample S, Features F)
2: if stopping_condition (S, F) = true then
3:     Leaf = createNode ()
4:     leafLabel = classify(s)
5:     return leaf
6: root = createNode ()
7: root.test_condition = findBest Spilt(S, F)
8: V = {v | v a possible outcomecroot.test_condition }
9: For each value v ∈ V :
10:     S_v = {s | root.test_condition (s) = v and s ∈ S};
11:     Child = TreeGrowth (S_v, F);
12:     Add child as descent of root and label the edge {root → child} as v
13: return root
14: end if

```

❷ **Random Forest Algorithm** : Similarly, Random forest [96] is a supervised ensemble learning algorithm that creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method that is better than a single decision tree because it reduces the over-fitting by averaging the result.

Random forest ensures that the behavior of each individual tree is not too correlated with the behavior of any other tree in the model by using Bagging or Bootstrap Aggregation and Random feature selection.

In this algorithm, we choose subset of features that has size equal to the 'k' number. Then we random sample from the original training examples with 'k' features to create the new training samples.

Algorithm 8 Pseudo code for the random forest algorithm

```

1: To generate  $c$  bootstrap samples:
2: for  $i = 1$  to  $c$  do
3:   Randomly sample the training data  $D$  with replacement to produce  $D_i$ 
4:   Create a root node,  $N_i$  containing  $D_i$ 
5:   Call BuildTree ( $N_i$ )
6: end for
7: BuildTree(N):
8: if  $N$  contains instances of only one class then
9:   return
10: else
11:   Randomly select  $x\%$  of the possible splitting features in  $N$ 
12:   Select the feature  $F$  with the highest information gain to split on
13:   Create  $f$  child nodes of  $N, N_1, \dots, N_f$ , where  $F$  has  $f$  possible values
      ( $F_1, \dots, F_f$ )
14:   for  $i = 1$  to  $f$  do
15:     Set the contents of  $N_i$  to  $D_i$ , where  $D_i$  is all instances in  $N$  that match
       $F_i$ 
16:     Call BuildTree ( $N_i$ )
17:   end for
18: end if

```

❸ **Support vector machines(SVM) Algorithm** : The objective of applying SVMs is to find the best line in two dimensions or the best hyperplane in more than two dimensions in order to help us separate our space into classes [97]. The hyperplane (line) is found through the maximum margin, i.e., the maximum distance between data points of both classes.

Mathematically, a hyperplane is a subspace whose dimension is one less than the ambient space. Meaning if the ambient space is a line, the hyperplane is a point. If the ambient space is a two-dimensional plane, the hyperplane is a line, and in three dimensional plan, the separating hyperplane is a plane.

A hyperplane of an "N-dimensional" space "V" is a subspace of dimension $N - 1$, or equivalently, of co-dimension 1 in "V". The space "V" may be a Euclidean space or more generally an affine space, or a vector space or a projective space, and the notion of hyperplane varies correspondingly since the definition of subspace differs in these settings; in all cases however, any hyperplane can be given in coordinates as the solution of a single (due to the "co-dimension 1" constraint) algebraic equation of degree 1.

Each data point is a vector in the feature space. The data points that are closest to the separating hyperplane are called support vectors because they support or aid the classification. Since a miss-classified data point inside the margin soft margin classifier allows minimal miss-classifications if the data points were linearly separable where as if the data points were distributed; projecting the points onto a higher dimensional features space requires us to map the data points from the original feature space to the higher dimensional space which is called kernel trick.

Algorithm 9 Pseudocode for the SVM algorithm

Required: X and Y located with training labeled data, $\alpha \Leftarrow 0$ or $\alpha \Leftarrow$ partially trained SVM

- 1: $C \Leftarrow$ some value (10 for example)
 - 2: **repeat**
 - 3: **for all** $\{x_i, y_i\}, \{x_j, y_j\}$ **do**
 - 4: Optimize α_i and α_j
 - 5: **end for**
 - 6: **until** no changes in α or other resource constraint criteria meet
 - 7: **Ensure:** Retain only the support vectors ($\alpha_i > 0$)
-

Expert knowledge modeling: Before building a Knowledge-Based System, knowledge must be gathered from domain experts. Models are used in systems development to simplify communication and depict system designs. Decision tree structures are used to model knowledge from experts, as they represent rules and

logical sentences. The prototype of Knowledge-Based System is developed based on this decision tree structure, as it is easily converted to IF-THEN rules, making it suitable for computer programs [98].

Knowledge representation: The study used rule-based knowledge representation to convert knowledge from knowledge discovery techniques into rules and domain expert interviews about Stroke diagnosis and treatment. This method uses a decision tree and production rule, making it easy to understand and efficient in detecting IF-THEN forms. Backward chaining was used as the inference engine due to SWI PROLOG's backward chaining mechanism.

3.3.2.6 Tools and Libraries

Python: Python is a high-level interpreted programming language used for algorithm development, data manipulation, and visualization in the Jupyter Notebook platform. It is easy to learn and can connect with system-level languages when needed. Python's benefits include available tools and libraries, making it attractive for workloads in data science, machine learning, and scientific computing [99].

Anaconda: Anaconda [100] is an open-source distribution for data science, offering over 1,500 packages suitable for Linux, Windows, and MAC platforms. It includes a GUI called Anaconda Navigator, Spyder, R studio, Jupyter notebook, and orange. Python has become a popular scientific programming language due to its increasing importance in scientific software. Anaconda is a good solution for data science practitioners [101].

Jupyter: Jupyter is a popular open-source web application that enables users to create documents with explanatory text, equations, visualizations, and live codes. It has become a popular tool for sharing computational information and explanations, and has become the data scientists' computational notebook of choice. The application simplifies data analysis, making it easier to record, understand, and reproduce, as argued by Helen Shen in Nature [102].

Flask Python: It is a Python-based back-end framework for creating applications and websites. It allows modeling data classes based on attributes and provides HTML output for front-end purposes.

Libraries: This thesis uses various libraries for data analysis, including Numpy for array handling, Pandas for data loading and analysis, Matplotlib for data visualization, Sklearn for data modeling, train test split for data splitting into training and tests, StandardScaler for data scaling and normalization, and Seaborn for data visualization tasks. These libraries are essential for efficient data processing and analysis.

Hardware tool: The software tools were utilized alongside an Intel Core i3-6100U CPU processor Lenovo Laptop computer with 8 Giga Byte RAM for documentation and implementation programming code.

3.3.3 Demonstration

The demonstration stage involves showcasing the artifact's functionality and potential impact through simulations, prototypes, or experiments. The artifact is used to solve problems and demonstrate the use of a knowledge-base system. This phase integrates machine learning and expert knowledge, such as developing a prototype for stroke diagnosis and treatment. Test cases are used to demonstrate the prototype's functionality.

3.3.4 Evaluation of Classifier Models

Knowledge-Based System evaluation process involves to determine the suitability and desirability of the prototype [103]. Effective Knowledge-Based System evaluation process incorporates both technical and non-technical aspects. The technical aspects include exploring of the code, examining the correctness of reasoning techniques, checking the efficiency and performance of the system and debugging errors in the primary stage of a system development. The non-technical aspect includes system compatible with users satisfaction, the easiness of the system, the quality of the user interface and the acceptability of the system in the real-world environments. According to Juristo and Morant [103] verification, validation, usability and usefulness are four types of evaluations to be conducted on Knowledge-Based System .

In order to evaluate the performance of the classifier model Accuracy, True Positive, False Positive, Precision, Recall, and F-Measure are commonly used. Confusion matrix helps to show correct and incorrect classified instances. Each performance metrics used for evaluating the generated classifier model is calculated from a confusion matrix [104]. Confusion matrix is shown below in figure 3.3 .

		Predicted class	
		Class Positive	Class Negative
Actual class	Class Positive	True positive	False negative
	Class Negative	False positive	True negative

FIGURE 3.3: confusion matrix

As illustrated in figure 3.3 confusion matrix has value of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN).

- **TP:** the number of correctly predicted positive values or number of correctly classified instance.
- **TN:** the number of correctly predicted negative values or the number of incorrectly classified instance.
- **FP:** the number of instance that is predicted as positive but actually negative class label.
- **FN:** the number of instance that detect as negative class label but actually it is positive class label.

3.3.4.1 Accuracy

Accuracy measures the proportion of instances that are correctly classified by the classifier.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3.7)$$

3.3.4.2 True Positive Rate (TPR) and False Positive Rate (FPR)

TPR: is the proportion of positive or correctly classified instances as positive or correct instances.

$$\text{TruepositiveRate} = \frac{TP}{TP + FN} \quad (3.8)$$

FPR: is measured the proportion of negative instances that are erroneously classified as positive.

$$\text{FalsepositiveRate} = \frac{FP}{FP + TN} \quad (3.9)$$

3.3.4.3 Precision and Recall

Precision: is measuring the proportion of instances that are classified as positive that are really positive. It can be thought of as a measure of correctness.

$$\text{FalsepositiveRate} = \frac{TP}{TP + FP} \quad (3.10)$$

Recall: is what percent of positive or negative tuples the classifier labeled as positive or negative for both True and False Classes. It is a measure of completeness (proportion of positive instances which are predicted to be positive)

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.11)$$

F-Measure: is calculated as the harmonic mean of recall and precision. it is computed by equation.

$$F1 = 2 * \left(\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \right) \quad (3.12)$$

3.3.5 Communication

The research findings are documented and communicated through research papers, presentations, or technical reports, contributing to the existing knowledge in the field. This phase can be the end of a research cycle or the final step of a specific research effort, submitting the findings as a thesis document.

To sum up, DSRM is an iterative process that involves feedback from evaluation and reflection stages, leading to refinements and further iterations in the design and development of the artifact, focusing on practical solutions and knowledge advancement.

Chapter 4

DOMAIN UNDERSTANDING AND DATA PREPARATION

Large databases are rapidly growing due to technological advancements that enable organizations to collect and store vast amounts of data [105]. This has led to a society heavily reliant on information, resulting in the accumulation of valuable information in various areas of human endeavor, from routine tasks like transactional data to more complex tasks like image processing and medical records.

The world's information volume doubles every 20 months, impacting scientific, government, and corporate information systems [106]. This massive data generation and storage leads to large databases of gigabytes and terabytes. To manage this growth, computing power, knowledge extraction techniques, and machine learning algorithms are available.

A machine learning process model [107] is a program that uses data collection to identify patterns or make decisions, aiming to create accurate predictive models. This process defines the workflow of a data science team, following standard steps as illustrated in Figure 4.1 [107]. This study examines data relevance, cleaning, errors, and outliers in real-world data due to its large size and heterogeneous sources, highlighting the importance of data preprocessing.

4.1 PROBLEM DOMAIN UNDERSTANDING

The study emphasizes the importance of domain problem understanding in health-care system planning and strategies, particularly in the construction of stroke treatment and diagnosis system, as it aids in understanding the required dataset.

The researcher focused on business understanding at Debre Birhan Referral Hospital by establishing close relationships with domain experts, identifying specific

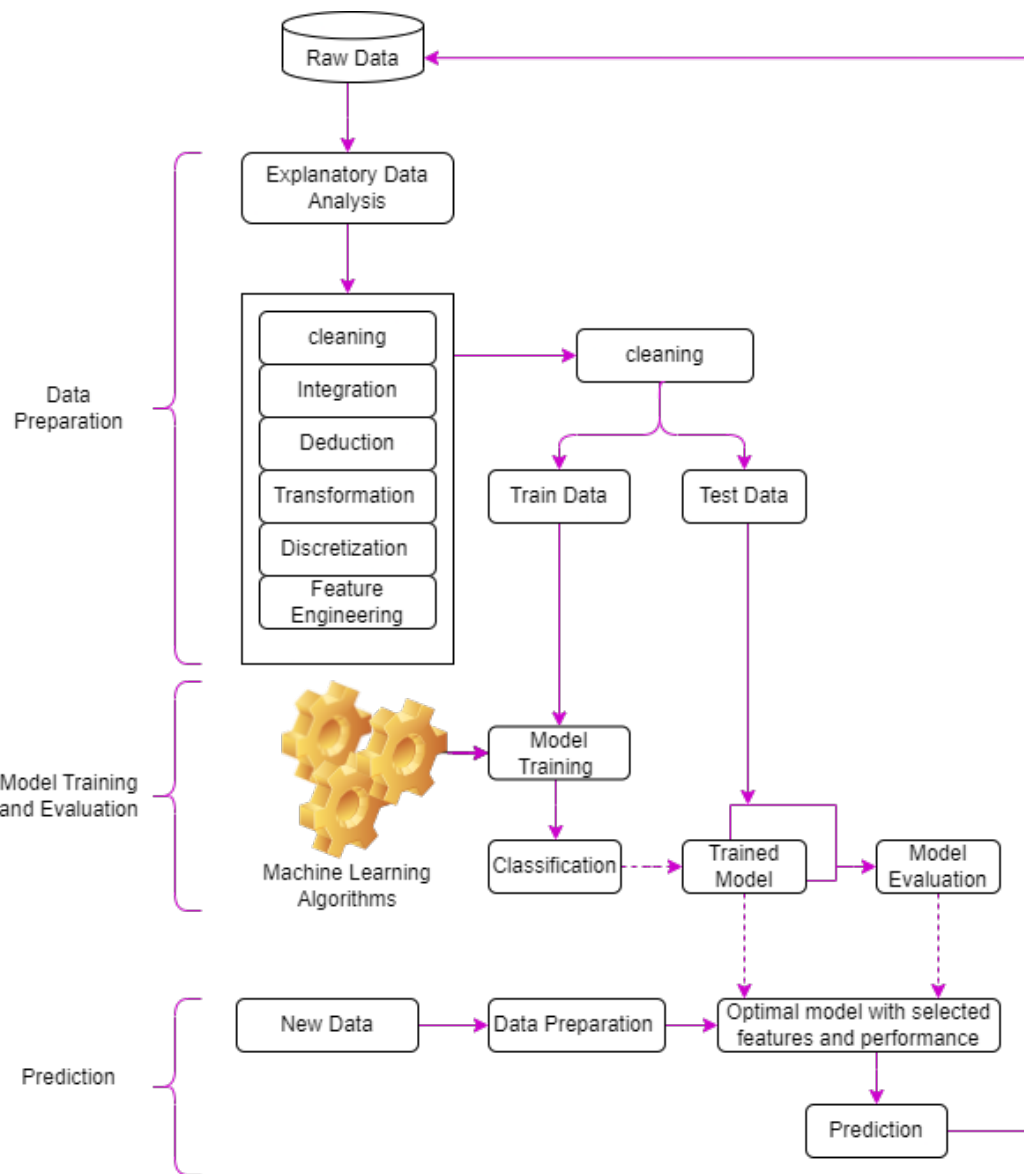


FIGURE 4.1: Machine learning process model

domain experts, learning existing processes, transforming problem into machine learning goals, and using selected algorithms to extract knowledge.

The study identifies Debre Berhan Referral Hospital’s core domain experts and analyzes stroke diagnosis and treatment documents to define the business problem and learning goal. A literature review and pilot study explore the business process, knowledge base application, and machine learning in stroke diagnosis and treatment. The researcher classifies the issue into three major tasks, identifying inputs, processing methods, and expected outputs.

Task-1: Domain experts Knowledge: The first task of this research aims to

understand stroke diagnosis and treatment processes by discussing with domain experts at Debre Birhan referral hospital. The researchers aim to gather relevant knowledge and suggest solutions to improve existing practices and challenges. The method of interview and target group are discussed in Chapter 2, and sample interview questions are presented in the Appendix. From this task the researcher identified pregnancy, Paralysis, Headache, Swallowing, and Balance are as basic attributes that is essential for stroke diagnosis and treatment.

Task-2: Document analysis: This thesis also uses document analysis, similar to interviews or focus groups, to analyze artifacts such as posters, flyers, agendas, training materials, and handbooks found in a study setting, such as stroke diagnosis and treatment guidelines, for the proposed knowledge-based system. Additionally from document analysis task Speech, Dizziness, Renal disease, Face, Arm, and Emotion are mandatory attributes for stroke diagnosis and treatment.

Task-3: Literature review: The researcher conducted a literature review on stroke diagnosis and treatment, identifying relevant theories, methods, and gaps in existing research. This helped us understand the current problem and determine the research goals and solution for the problem.

Task 4: Supporting Health Professionals in Stroke Diagnosis and Treatment Processes: The study aims to support health professionals in stroke diagnosis and treatment processes by integrating machine learning predictive models with domain expert knowledge. Current practices rely on expert knowledge and are not fully integrated. The study aims to quantify issues affecting stroke diagnosis and treatment, focusing on attributes from domain experts' suggestions, document analysis, and literature review. Potential interactions among predictors and diagnosis and treatment were assessed from various sources to create a useful prediction model.

By adding patients' background factors like Pregnancy, Speech, ... etc. the attributes identified from the above tasks as predictors of stroke diagnosis and treatment are shown in Table 4.1. This approach aims to improve the accuracy of stroke diagnosis and treatment.

No	Attribute Name	Type(Values)	Description
1	Speech	string literal(Normal, Abnormal)	it tells about whether the patients speech are normal or not
2	Dizziness	string literal(yes, no)	it tells about whether the patients having dizziness or not
3	Renal disease	string literal(yes,no)	it tells about whether the patients having Renal disease or not
4	Face	string literal(Normal, Abnormal)	it tells about whether the patients face are normal or not
5	Arm	string literal(Normal, weak)	it tells about whether the patients arm are normal or not
6	Emotion	string literal (Conscious, Unconscious)	it tells about whether the patients emotion are Conscious or unconscious
7	pregnancy	string literal(yes, No)	it tells about whether the patients having pregnancy or not
8	Paralysis	string literal(yes, No)	it tells about whether the patients having paralysis or not
9	Headache	string literal(normal, sever)	it tells about whether the patients having sever headache or not

10	Swallowing	string literal(normal, abnormal)	it tells about whether the patients swallowing is normal or not
11	Vision	string literal(normal, trouble)	it tells about whether the patients vision is normal or trouble
12	Balance	string literal(normal, abnormal)	it tells about whether the patients vision is normal or not
13	Bladere	string literal(normal, abnormal)	it tells about whether the patients bladere is normal or not

TABLE 4.1: Selected attributes from the business domain.

4.2 UNDERSTANDING OF THE DATA

The data understanding phase in machine learning involves analyzing and understanding available data, including sample data, deciding on the appropriate format and size, checking for completeness, redundancy, missing values, and acceptability of attribute values [89]. The researcher used secondary data from the Kaggle website healthcare dataset, which was retrieved from the Kaggle website. The data understanding phase focuses on creating a target dataset with relevant variables for knowledge extractions. It is crucial to pay attention to clean data for knowledge extractions, as real-world data is susceptible to noise, inconsistency, and incompleteness. Larger data sizes and heterogeneous sources can reduce predictive performance of a model in machine learning.

4.2.1 The raw data descriptions

This study focuses on analyzing primary data from domain experts at Debre Berhan Referral Hospital and the public dataset accessed from the Kaggle website which uploaded 3 years ago. The dataset, which includes 5110 rows and 12 columns, includes eleven independent variables and one dependent variable. The

study aims to extract knowledge from this data to input into a knowledge base. The data set was initially available in CSV format on the Kaggle website but it was not processed to select relevant attributes. The main attributes in the dataset include I.D., Gender, Age, Hypertension, Heart disease, Ever married, Work type, Residence type, Average glucose level, BMI, Smoking status, and Stroke. The output column 'stroke' has a value of either '1' or '0', with only 249(5%) rows having '1'(stroked) values and 4861 rows having '0'(no stroked)values. The dataset discussed above is summarized in Table 4.2.

No	Attribute Name	Type(Values)	Description
1	id	integer	a unique integer value for patients
2	gender	string literal(male, female, other)	tells the gender of patients
3	age	integer	age of the patients
4	hypertension	integer(0,1)	tells whether the patients has hypertension or not
5	heart disease	integer(0,1)	tells whether the patients has heart disease or not
6	ever married	string literal(yes,no)	tells whether the patients married or not
7	work type	string literal(children, Gov't job, never worked, private, self employed)	it gives different categories for work
8	residence type	string literal(urban, rural)	the patients residence type stored
9	average glucose level	floating point number	gives the value of the patents average glucose level in blood

10	bmi	floating point number	gives the value of the patents body mass index
11	smoking status	string literal(formerly smoke, never smoke,smokes, unknown)	it gives the smoking status of the patient
12	stroke	integer(0,1)	output column that gives the patient stroke status

TABLE 4.2: Stroke dataset description.

4.3 DATA PREPROCESSING

Data preprocessing [108] is a crucial step in machine learning model building to remove unwanted noise and outliers, ensuring optimal training efficiency. This process includes data cleaning, integration, reduction, and transformation. The quality of data is crucial, with measures such as accuracy, completeness, consistency, timeliness, acceptability, and interpretability. Real-world data is highly susceptible to noise, missing values, and inconsistency due to its large size and heterogeneous sources. Inconsistencies, incompleteness, and noise are common properties of large databases. To address these issues, data preprocessing techniques like data cleaning can be applied to handle missing values, remove noise to handle outliers, and correct inconsistencies in the data. These techniques help to ensure the accuracy, completeness, consistency, timeliness, acceptability, and interpretability of the data.

4.3.1 Data visualization

As mentioned above the dataset taken for this research has 12 attributes, to begin with data preprocessing, the overall data were visualised with respect to target variable. The following figures from figure 4.2 - 4.4 shows the distribution of numerical, categorical and target variables distribution.

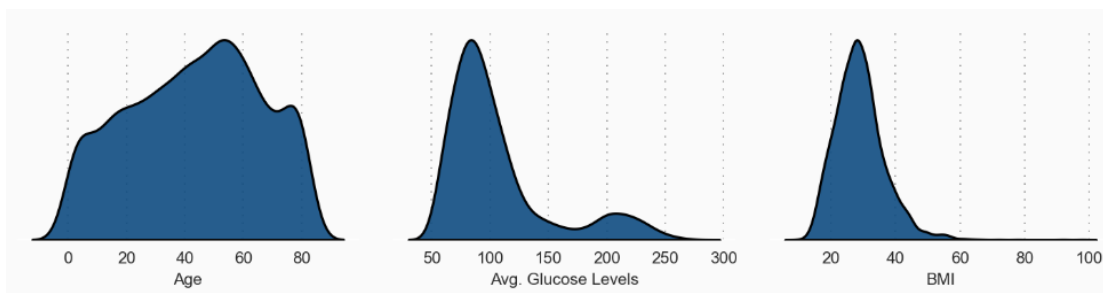


FIGURE 4.2: numerical variables distribution

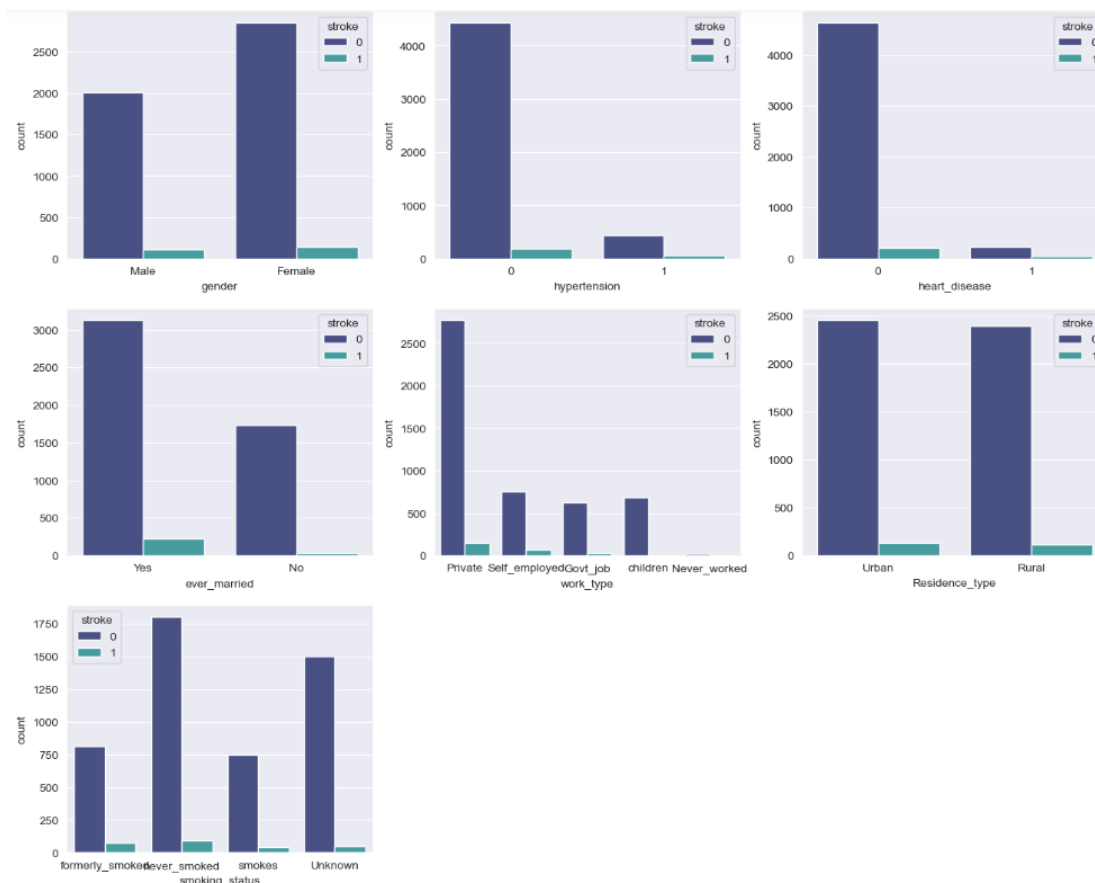


FIGURE 4.3: categorical variables distribution

From the overall visualizations number of female is greater than number of male in data set. More people are free from hypertension and heart disease and more of them are married. There is a balanced distribution between people living in rural and urban areas in the data set. More people never smoke, but people whose preferences unknown are also in majority. Maybe these people smoked and didn't want to say it. The dependent variable, stroke, is not evenly distributed in the data set. Numerical Variables; Avg glucose level and Bmi variable make a hill around 80 and 25 respectively. But it is a right-skewed distribution. Additionally

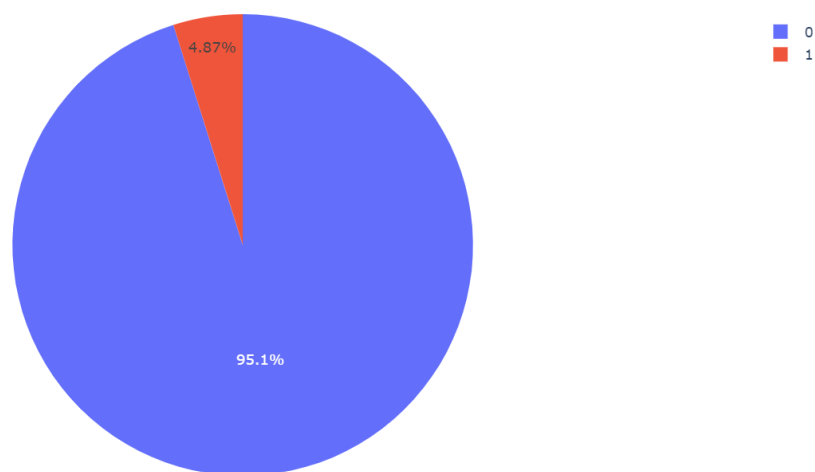


FIGURE 4.4: proportion of target variables(stroke)

Age variable include wide distribution.

From basic data analysis the most dependent variables that affect stroke status are hypertension, heart_disease, age_group, and glucose_status. From numerical variables age and avg_glucose_level variable seems to be effective in the occurrence of stroke. There is a clear difference between medians. Where as the bmi variable seems to be non effective in the occurrence of stroke. The relationships between variables are examined as below in figure 4.5.

In the following figures; figure 4.6 - figure 4.8 shows the risk of having a stroke based on some of independent variables.

From the risk level analysis; people with more than 70 age has a high to have stroke rate and children has a high to don't have stroke rate also; More stroke cases were observed in elderly women than in men. People with more than 200 avg_glucose_level has a high to have stroke rate according who don't have a stroke. But there is no clear distinction in BMI distribution

The effect of categorical variables on the dependent Variable is not the same for all such as; Gender and resident_type variables do not seem to have much effect on the dependent variable. but Other variables have an significant effect on the dependent variable. In some variables we make arrangements such as for the work_type variable, private and govt_jobs groups have the same effect on

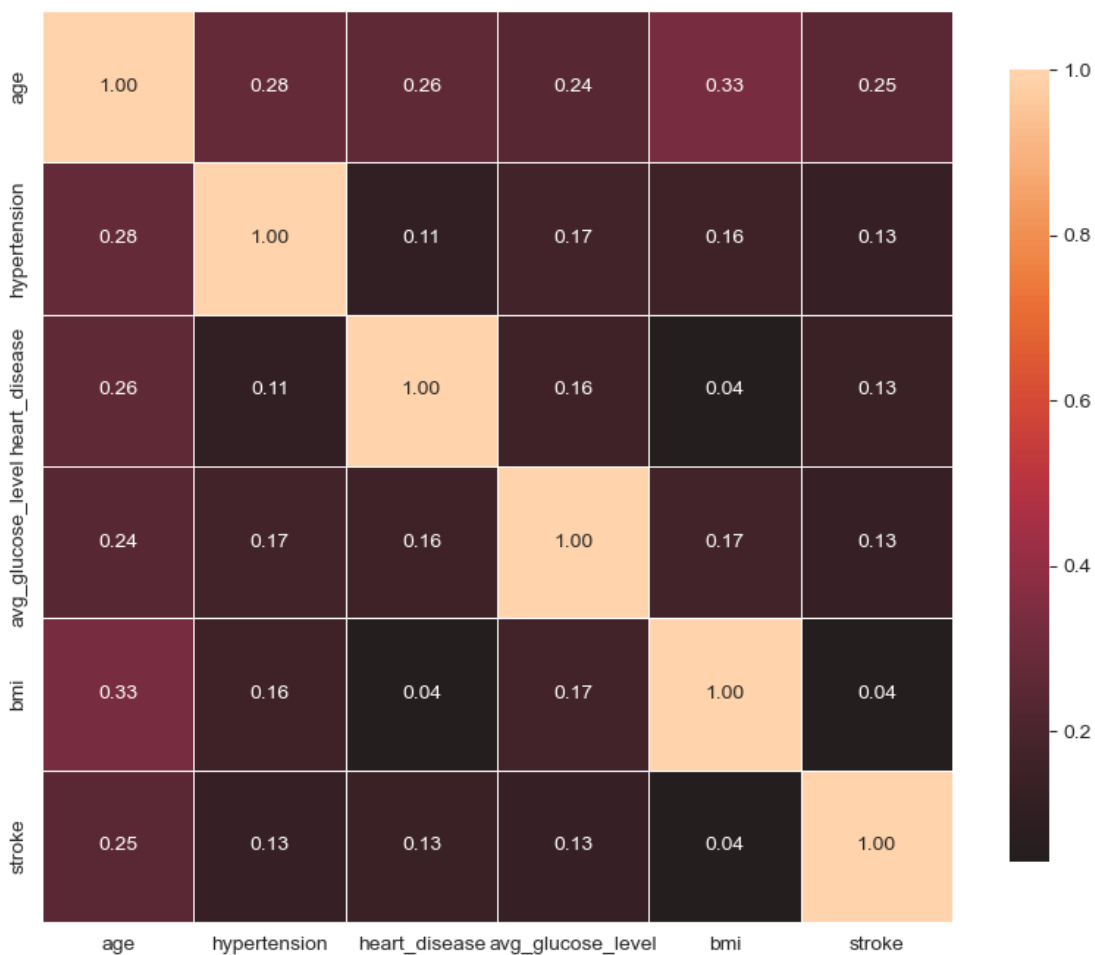


FIGURE 4.5: Correlations of variables

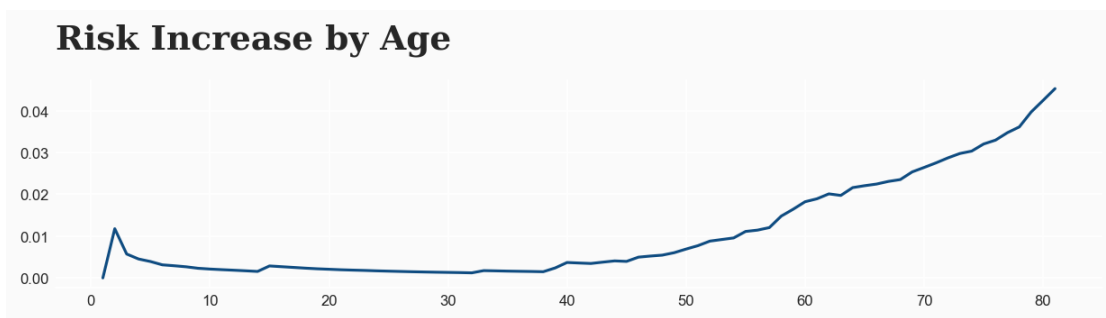


FIGURE 4.6: Risk level by age

the dependent variable. Also On the smoking_status variable, never smoked and smokes groups have the same effect on the dependent variable.

From the dataset some Outliers and Noisy values are their so this should be solved. The column 'id' is dropped because its existence does not make much difference in model building, as it is unique for each patient record and also under 'gender' column their is one instance with value of 'other' so it is omitted as it doesn't have

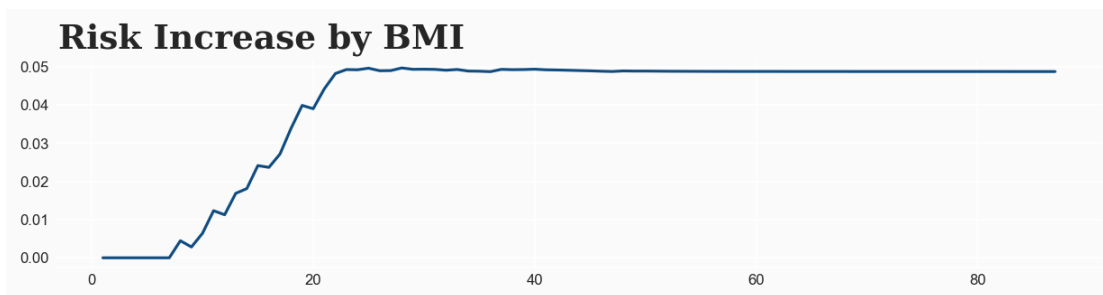


FIGURE 4.7: Risk level by BMI

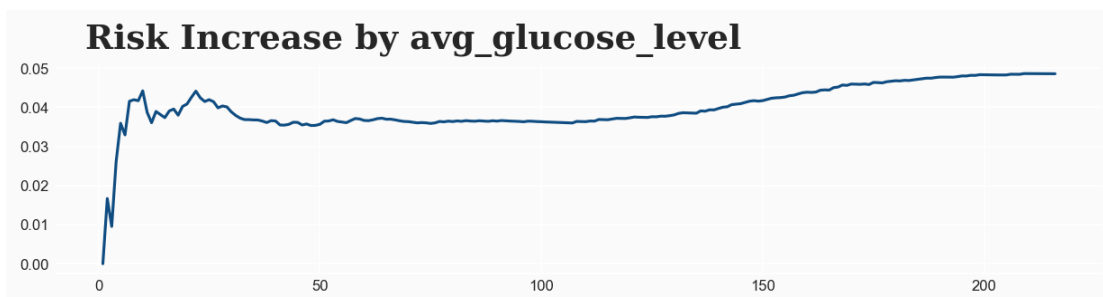


FIGURE 4.8: Risk level by avg_glucose_level

more effect. Figure 4.9 below shows before and after omitting 'other'.

<pre>dt.gender.value_counts().sort_values(ascending=False).head() Female 2994 Male 2115 Other 1 Name: gender, dtype: int64</pre>	<pre># Dropping Other gender Other_gender = dt[dt['gender'] == 'Other'].index[0] dataset11 = dt.drop(Other_gender, axis=0) dataset11.gender.value_counts().sort_values(ascending=False).head() Female 2994 Male 2115 Name: gender, dtype: int64</pre>
before droop	after droop

FIGURE 4.9: 'other' value drooped from gender column

There are also some skewed and close to normally distributed columns from the visualization. So, the researcher expected to handle the outliers of those columns. From outliers handling technique; Trimming will remove the outliers from the dataset. The process is simple, but if there's more outliers, the data frame will be thin. Whereas Capping will set some min max rule to the outliers. They will round up to the given min or max value. No data will be lost but it is time consuming. From those the researcher use Capping technique. The following figure 4.10 shows before and after skewed handled.

4.3.2 Handling missing values

The Dataset originally had 201 missing values for Body Mass Index (BMI) feature. These values were filled by calculating the mean BMI for the whole dataset. Also,

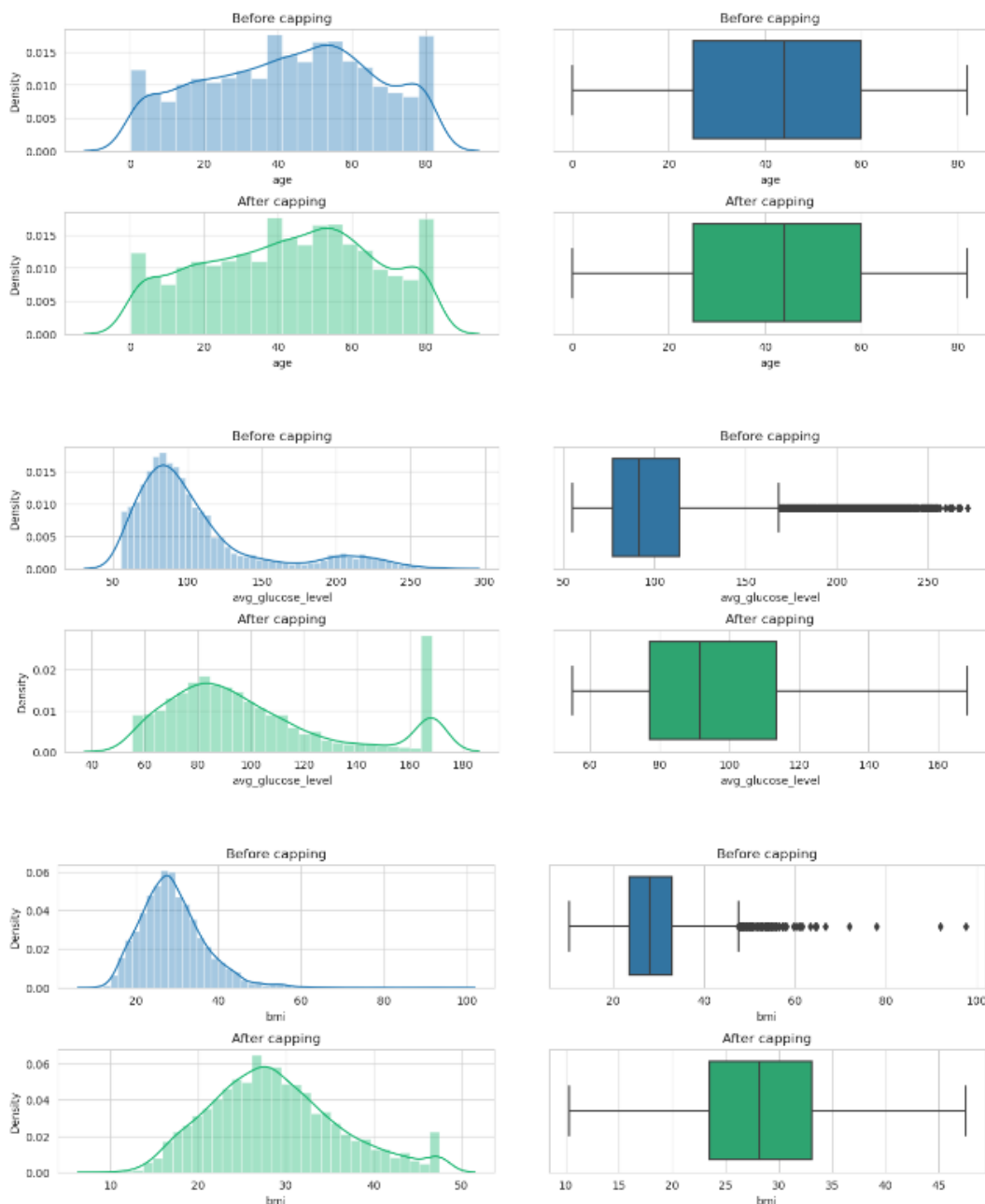


FIGURE 4.10: Before and after the Skew of variables

it was observed that more than 30% of the individuals have an 'Unknown' smoking status, which can be also considered as missing data or not having enough information about this feature values. In order to solve this we can't omit because of the amount of it, so that we decided to re-categorize those individuals by making some assumptions. As people younger than 18 years old, are less likely to smoke or have smoked, the 'Unknown' values present in these individuals were changed to 'never'. This reduced the number of 'unknowns' from 1544 to 909, which were

then deleted from the Dataset. Another re-classification made was to change every work_type values from 'children' to 'never worked'. This is because children should not have been considered as a work type in the first place and may imply 'never worked' values. The following figure 4.11 shows handling of missing values for BMI attribute.

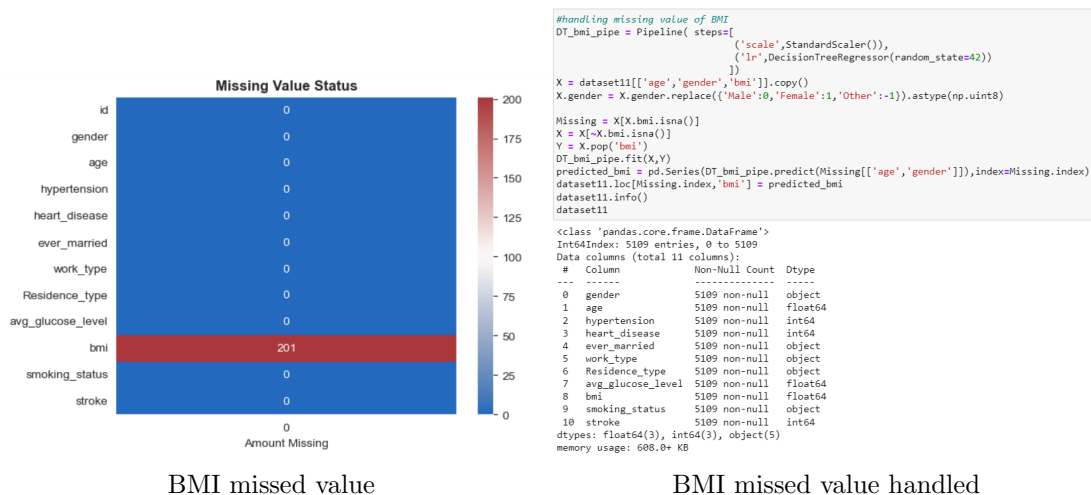


FIGURE 4.11: Missed value handling

4.3.3 Handling imbalance dataset

The classification techniques typically consider a balanced class distribution between two or more classes in machine learning. The problem of imbalanced class division occurs when one class is represented by a large number of instances while the remaining other is represented by only a few [109].

Almost 95% of the instances in our target variable haven't experienced with 'stroke' representing 4861 patients. On the other hand 5% of the instances in our target variable go through 'Stroke' representing 249 patient. This shows that Class labels were imbalanced and to avoid this, Synthetic Minority Oversampling Technique (SMOTE) was used before conduct the experiments. As a result the dataset increases from 5109 to 9720. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining of the minority class nearest neighbors. Figure 4.13 below shows the imbalanced and balanced class variables of the dataset.

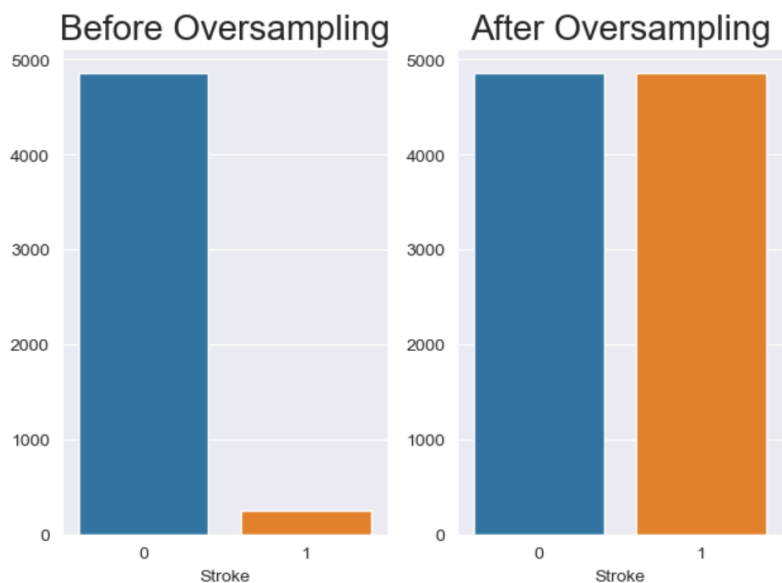


FIGURE 4.12: Before and after oversampling

After SMOTE analysis the next step was doing experimentation to extract knowledge from the processed dataset for using this extracted knowledge for stroke diagnosis and treatment in knowledge-based system and manage the patients using this developed knowledge-based system to get earlier services and take remedial action on their life.

4.3.4 Encoding and Embedding

Encoding is a way of changing character values that convert into numerical values. Machine-learning algorithms accept the input data contain only numbers, not strings (categories). The original data used in this research work contain categorical variables that cannot be directly input into the model such as gender, ever married, work type, residence type and smoking status.

In this study we have used One-hot encoder. One-hot encoding [110] can convert categorical variables into a numerical form that can be used in machine learning algorithms and it can enable algorithms to better understand categorical variables; but this conversion of categorical data to numerical data result-out high dimensionality and this affect the efficiency of machine learning prediction performance.

To mitigate the impact of this encoding in this research Word embedding with

Continuous Bag-Of-Word(CBOW)is used to represent words as vectors in a low-dimensional space. The following figure 4.12 shows one-hot encoding and embedding result in our research work.

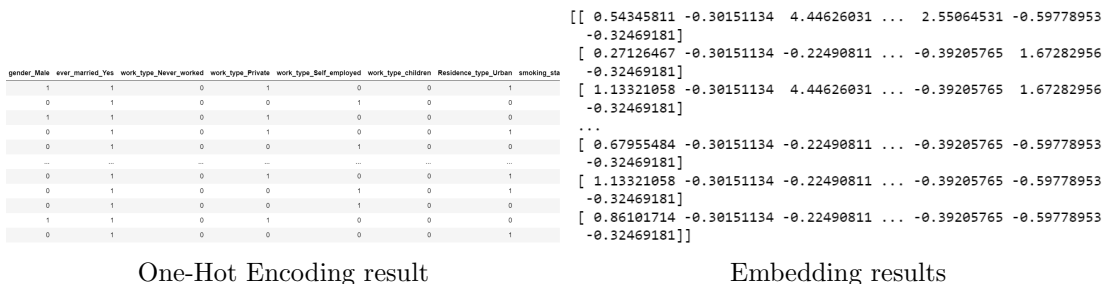


FIGURE 4.13: Sample Encoding and Embedding Results

4.3.5 Data Normalization

The dimensions and value ranges of the different parameters vary greatly and so they cannot be compared horizontally, nor can they be directly input into the model. In order to ensure that the dimensions and value ranges between different features will not have an adverse effect on the model training, the input data need to be normalized before they are input into the model for training [111]. This research chooses the Standardization using StandardScaler function. It is a scaling technique wherein it makes the data scale-free by converting the statistical distribution of the data into the entire data set scales with a zero mean and unit variance altogether [112]. Mathematically it can be described as the following in equation 4.1.

$$Z = \frac{x - \mu}{\delta} \tag{4.1}$$

Processing data with standardization solves the problem of large differences in dimensions and value ranges between different features, making different features comparable and helping to improve the training speed and model performance. This method of normalization are implemented for all attributes after encoding and embedding.

4.3.6 Dataset Splitting

The next stage is to construct the model after finishing data preparation and managing the imbalanced dataset. To improve the accuracy and efficiency of

this job, We separate the columns (attributes or features) of the dataset into input patterns (X) and output patterns (y). Farther more we split the X and Y data into a training and test dataset with a ratio of 80% and 20% respectively. The training set use to prepare the models, while the test set use to make new predictions, from which we evaluate the performance of the model. For this we use the `train_test_split()` function from the scikit-learn library.

After splitting, the model is trained using a variety of classification methods. Random forest, decision tree and Support vector machine are the classification algorithms utilized in this study.

Chapter 5

EXPERIMENTATION

This chapter discusses the experimentation process for developing a classifier model to extract actionable classification rules from a dataset and knowledge from domain experts in DBRH. A total of 9 experiments were conducted using classification machine learning techniques to derive knowledge from preprocessed data for stroke diagnosis and treatment. The methodology involved preprocessing 9720 data records and training (7,776) and testing (1,944) with three conventional machine learning algorithms.

5.1 EXPERIMENTAL SETUP

The study focuses on developing a predictive model using selected classifier algorithms, Decision trees, Random Forests, and Support vector machines. These algorithms are chosen for their ease of understanding, high tolerance to noise, and ability to classify unseen patterns, making them suitable for interpreting model results. A total of 9 experiments are conducted with two scenarios, one with all attributes and the other with selected attributes, using three classification algorithms. Two methods are used to classify the dataset: k-fold (10-fold) cross-validation and percentage split. The 10-fold method involves randomly partitioning datasets into ten parts, with 90% for training and 10% for testing. This method is statistically sound and has low bias and variations. The learning scheme is trained ten times using 9-tenths of the total data, while the remaining one-tenth is used for testing. The researcher used a percentage split test (80%) for training and a 20% test for classification, using 7,776 instances out of 9720 records as training data and 1944 instances as testing data. The models' performance was evaluated using standard metrics like Accuracy, Precision, Recall, and F-measure. The experimental Parameters with separate values are illustrated in Table 5.1.

scenario	Experiment No	No of attribute	Algorithm	Test option
Scenario-1	Experiment 1	12	Decision tree	10-fold cross validation
	Experiment 2	12	Random forest	10-fold cross validation
	Experiment 3	12	Support vector machine	10-fold cross validation
Scenario-2	Experiment 4	8	Decision tree	10-fold cross validation
	Experiment 5	8	Random forest	10-fold cross validation
	Experiment 6	8	Support vector machine	10-fold cross validation
	Experiment 7	8	Decision tree	Percentage split
	Experiment 8	8	Random forest	Percentage split
	Experiment 9	8	Support vector machine	Percentage split

TABLE 5.1: Experimental parameter with separate value

5.2 CONSTRICTING PREDICTIVE MODEL

5.2.1 Scenario 1: Experiments with all attributes

The scenario involves three 10-fold cross-validation experiments using Decision tree, Random forest, and Support vector machine algorithms, with each experiment's results explained as follows.

5.2.1.1 Experiment 1 Decision tree classifier with all attributes under 10-fold

The number of instances correctly classified by Decision tree algorithm with all attribute under 10-fold cross validation were 9720 (100%) and no incorrectly classified instances (0%) from a total of 9720 instances. From confusion matrix of

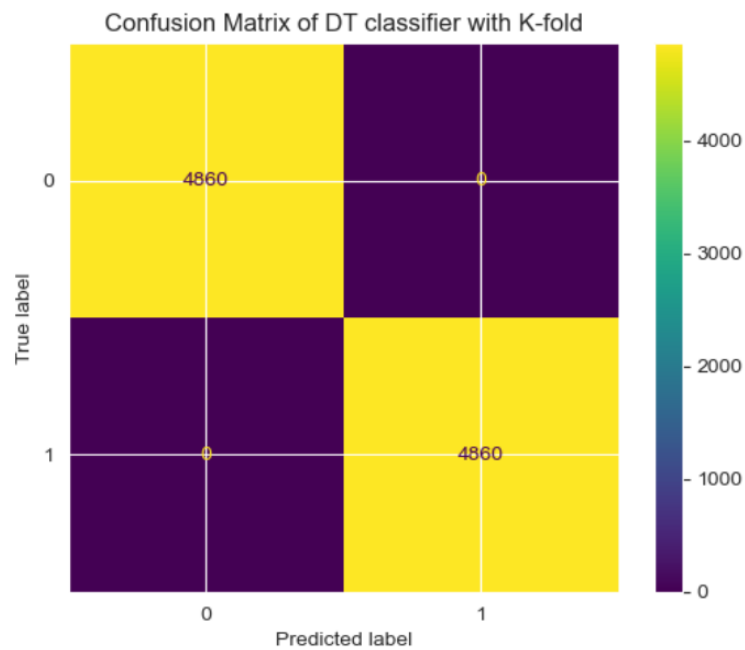


FIGURE 5.1: Confusion Matrix of Decision Tree classifier with all attribute under 10-fold

Decision Tree algorithm with all attributes as shown in figure 5.1 the model correctly classified 4860 as 1(stroked) out of 4860 1(stroked) instance and their was no incorrectly classified as 0(non-stroked); and also 4860 as 0(non-stroked) out of 4860 0(non-stroked) instance classified correctly and their was no incorrectly classified as 1(stroked). The performance result is shown in Table 5.2, with a mean accuracy of 92%, indicating a high level of accuracy in classification.

Class	TPR	FPR	Precision	Recall	F- measure
0	1	0	1	1	1
1	1	0	1	1	1

TABLE 5.2: performance result of Decision Tree classifier with all attribute under 10-fold.

5.2.1.2 Experiment 2 Random forest classifier with all attributes under 10-fold

The number of instances correctly classified by Random forest algorithm with all attribute under 10-fold cross validation were 9720(100%) and no incorrectly classified instances (0%) from a total of 9720 instances.

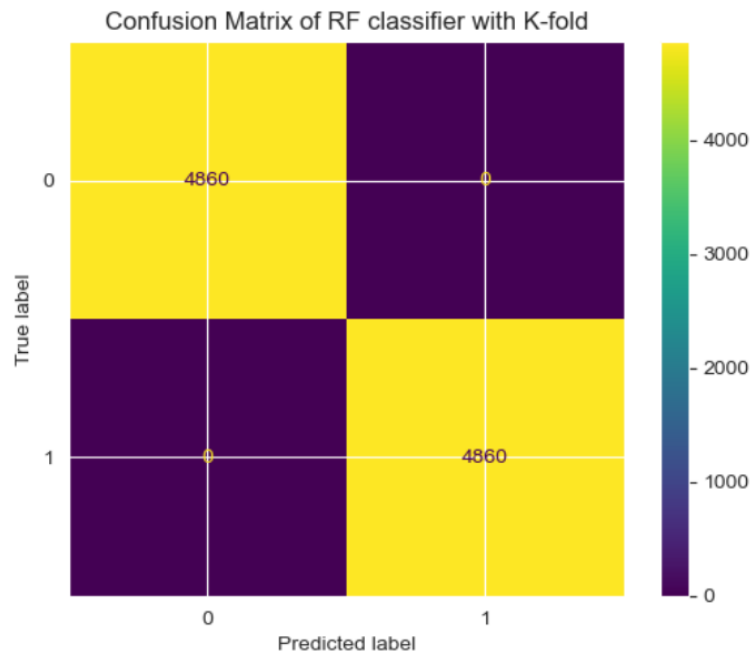


FIGURE 5.2: Confusion Matrix of Random forest classifier with all attribute under 10-fold

From confusion matrix of Random forest algorithm with all attributes as shown in figure 5.2 the model correctly classified 4860 as 1(stroked) out of 4860 1(stroked) instance and their was no incorrectly classified as 0(non-stroked); and also 4860 as 0(non-stroked) out of 4860 0(non-stroked) instance classified correctly and their was no incorrectly classified as 1(stroked). Based on confusion matrix the performance result is shown in Table 5.3 and its mean accuracy is 94%.

Class	TPR	FPR	Precision	Recall	F- measure
0	1	0	1	1	1
1	1	0	1	1	1

TABLE 5.3: performance result of Random forest classifier with all attribute under 10-fold.

5.2.1.3 Experiment 3 Support vector machine classifier with all attributes under 10-fold

The number of instances correctly classified by Support vector machine algorithm with all attribute under 10-fold cross validation were 8380(86.3%) and 1340(13.7%) incorrectly classified instances from a total of 9720 instances.

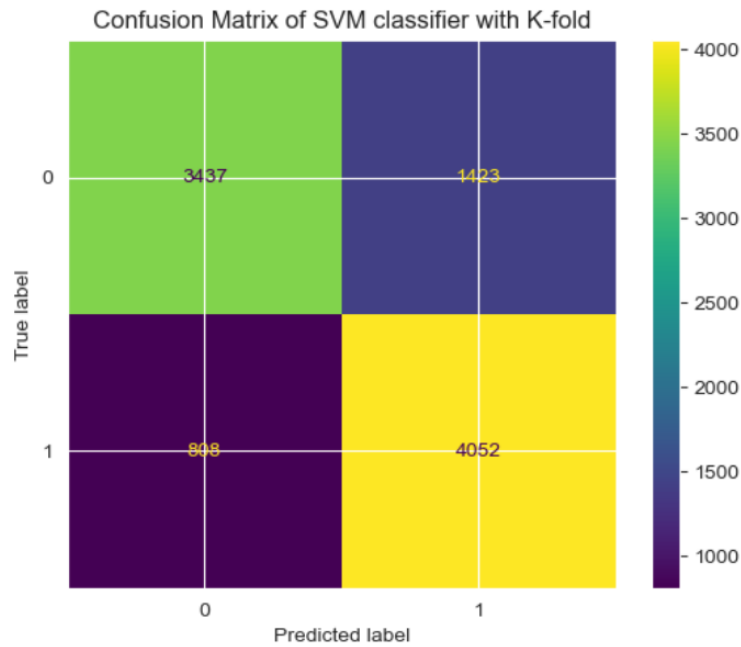


FIGURE 5.3: Support vector machine classifier Confusion Matrix with all attribute under 10-fold

From confusion matrix of Support vector machine algorithm with all attributes as shown in figure 5.3 the model correctly classified 4052 as 1(stroked) out of 4860 1(stroked) instance and their was 808 incorrectly classified as 0(non-stroked); and also 3437 as 0(non-stroked) out of 4860 0(non-stroked) instance classified correctly and their was 1423 incorrectly classified as 1(stroked). Based on confusion matrix the performance result is shown in Table 5.4 and its mean accuracy is 72%.

Class	TPR	FPR	Precision	Recall	F-measure
0	0.707	0.166	0.809	0.707	0.754
1	0.833	0.292	0.742	0.833	0.784

TABLE 5.4: performance result of Support Vector Machine classifier with all attribute under 10-fold.

In the first scenario mostly in experiment 1 and 2 the result shown that there is the problem of over-fitting. To ensure optimal performance and accurate predictions in machine learning models, it is crucial to address the issues of over-fitting. These problems occur when a model fails to generalize well to new, unseen data. Fortunately, based on the cause several effective methods can be employed to mitigate these challenges. The researcher implement the action of careful feature selection that plays a pivotal role in avoiding over-fitting. By eliminating irrelevant features, the model becomes less prone to over-fitting and can focus on the most informative attributes. This is done in the next scenario below.

5.2.2 Scenario 2: Experiments with selected attributes

This scenario conducted with selected attributes and incorporates six experiments these are Experiment 4, Experiment 5 and Experiment 6 conducted under 10-fold cross validation test option mode where as Experiment 7, Experiment 8 and Experiment 9 under percentage split described below respectively. To begin with the experiment attributes are selected based on relationship between some important attributes with the target feature. The following Figure 5.4 shows the relationship between important features with the target feature.

5.2.2.1 Experiment 4 Decision tree classifier with selected attributes under 10-fold

The number of instances correctly classified by Decision tree algorithm with selected attribute under 10-fold cross validation were 1790(92.07%) and incorrectly classified instances of 154(7.93%) from a total of 1944 instances. The following figure 5.5 shows the confusion matrix and performance results of Decision tree classifier with selected attributes under 10-fold.

5.2.2.2 Experiment 5 Random forest classifier with selected attributes under 10-fold

The number of instances correctly classified by Random forest algorithm with selected attribute under 10-fold cross validation were 1930 (99.27%) and incorrectly classified instances of 14(0.73%) from a total of 1944 instances. The following

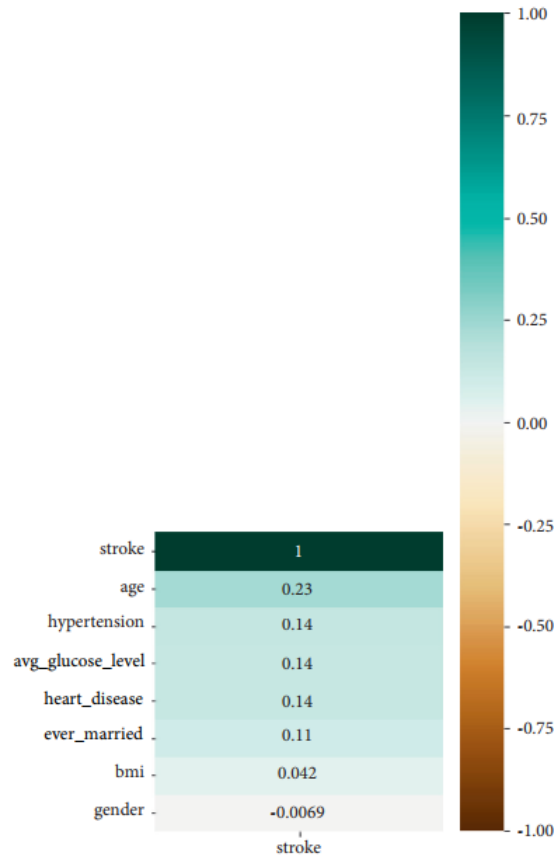


FIGURE 5.4: Selected Independent features with correlation to target feature.

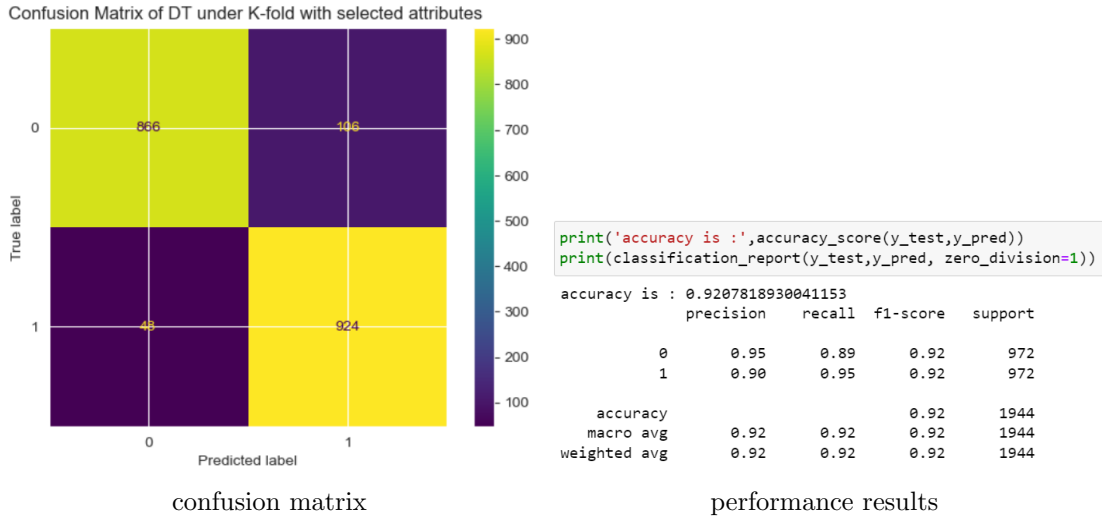


FIGURE 5.5: Decision tree classifier with selected attributes under 10-fold

figure 5.6 shows the confusion matrix and performance results of Random forest classifier with selected attributes under 10-fold.

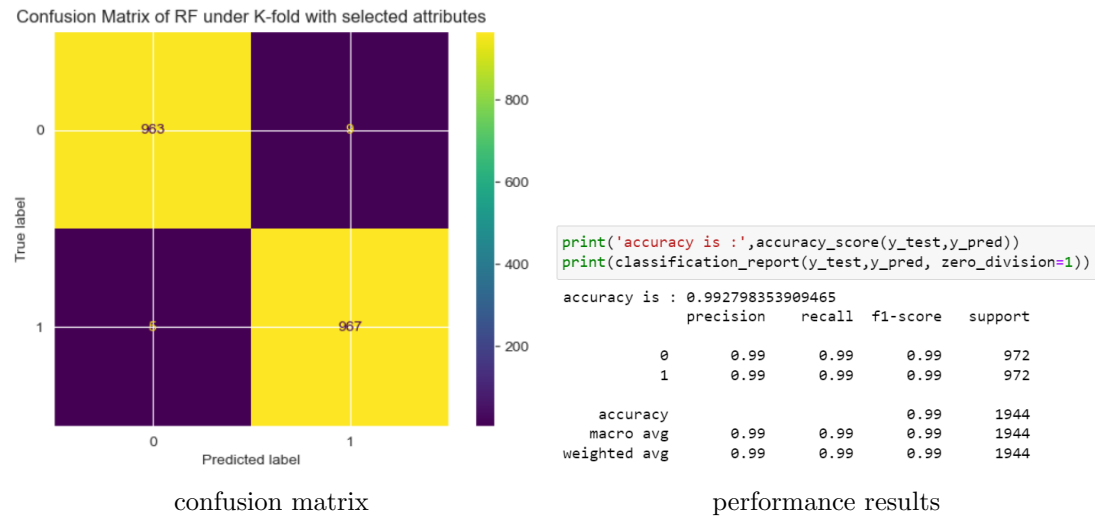


FIGURE 5.6: Random forest classifier with selected attributes under 10-fold

5.2.2.3 Experiment 6 Support vector machine classifier with selected attributes under 10-fold

The number of instances correctly classified by Support vector machine algorithm with selected attribute under 10-fold cross validation were 1469 (75.57%) and incorrectly classified instances of 475(24.43%) from a total of 1944 instances. The following figure 5.7 shows the confusion matrix and performance results of Support vector machine classifier with selected attributes under 10-fold.

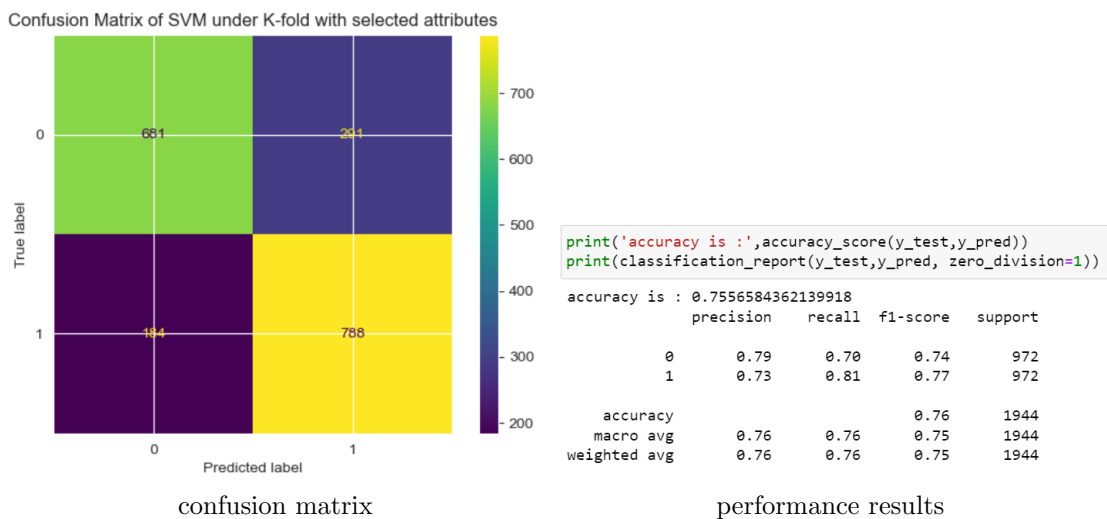


FIGURE 5.7: Support vector machine classifier with selected attributes under 10-fold

5.2.2.4 Experiment 7 Decision tree classifier with selected attributes

The number of instances correctly classified by Decision tree algorithm with selected attribute were 1790 (92.07%) and incorrectly classified instances of 154(7.93%) from a total of 1944 instances. The following figure 5.8 shows the confusion matrix and performance results of Decision tree classifier with selected attributes.

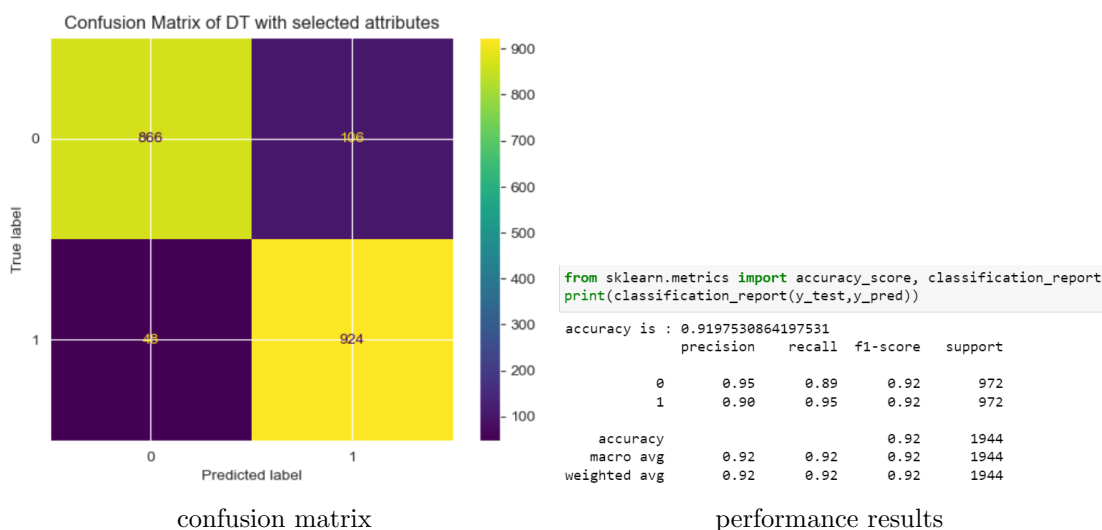


FIGURE 5.8: Decision tree classifier with selected attributes

5.2.2.5 Experiment 8 Random forest classifier with selected attributes

The number of instances correctly classified by Random forest algorithm with selected attribute were 1852(95.27%) and incorrectly classified instances of 14(4.73%) from a total of 1944 instances. The following figure 5.9 shows the confusion matrix and performance results of Random forest classifier with selected attributes.

5.2.2.6 Experiment 9 Support vector machine classifier with selected attributes

The number of instances correctly classified by Support vector machine algorithm with selected attribute were 1469 (75.57%) and incorrectly classified instances of 475(24.43%) from a total of 1944 instances. The following figure 5.10 shows the confusion matrix and performance results of Support vector machine classifier with selected attributes.

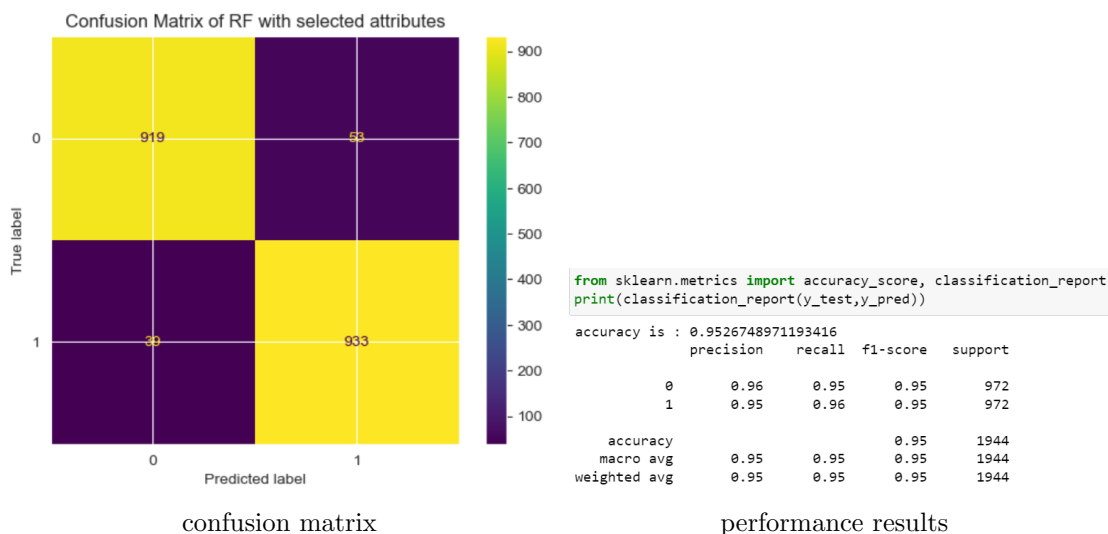


FIGURE 5.9: Random forest classifier with selected attributes

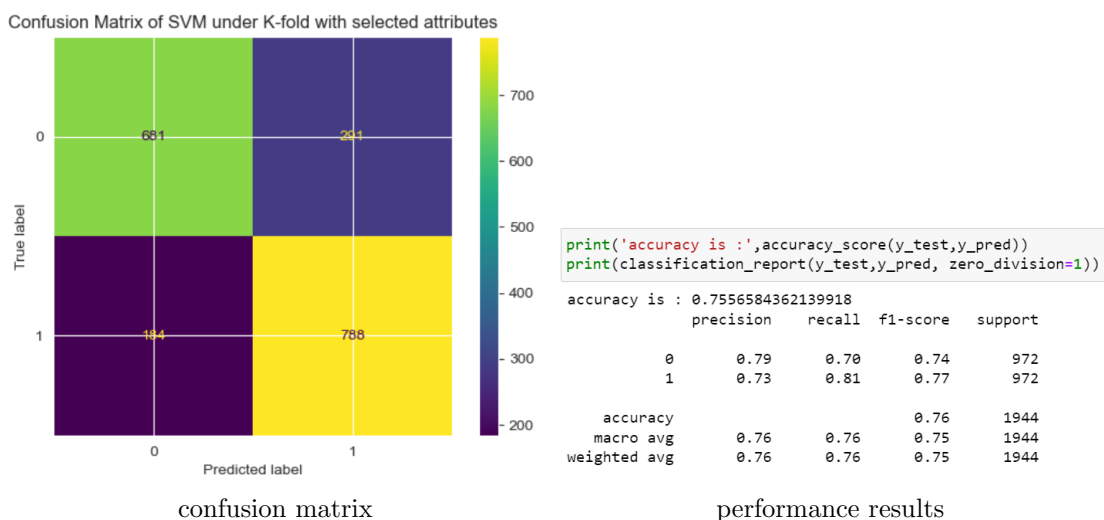


FIGURE 5.10: Support vector machine classifier with selected attributes

5.3 MODEL COMPARISON AND SELECTION

5.3.1 Model Comparison

In order to meet research objective, one task is develop a model that can predict stroke and select with the best performance from these model to develop knowledge-based system . In order to select the best model Decision tree, Random forest and Support vector machine algorithms using cross-validation (10-folds) and percentage split were used for conducting experiments. Basically, the experiments were conducted on two setup the first containing all the attributes and the second with the selected number of attributes. The models were compared using different

performance measures like Accuracy, TPR, FPR, Precision and Recall.

Table 5.5 has compare the output of all the 9 experiments based on accuracy with 10-fold cross-validation and percentage split test option mode to select best classifier model and next take highest accuracy model and compare with the performance of individual class level in terms of TPR, FPR, Precision Recall and Accuracy.

scenario	Experiment No	No of at-tribute	Algorithm	Test option	Accuracy
Scenario 1	Experiment 1	12	DT	10-fold CV	92%
	Experiment 2	12	RF	10-fold CV	94%
	Experiment 3	12	SVM	10-fold CV	72%
Scenario 2	Experiment 4	8	DT	10-fold CV	92%
	Experiment 5	8	RF	10-fold CV	99%
	Experiment 6	8	SVM	10-fold CV	76%
	Experiment 7	8	DT	Percentage split	92%
	Experiment 8	8	RF	Percentage split	95%
	Experiment 9	8	SVM	Percentage split	76%

TABLE 5.5: Performance Comparison of experimental results in accuracy

From the experiment result with considering both scenarios the classification accuracy of the algorithms increased with selected attributes because selected attributes were more relevant in stroke prediction. To support this argument the researcher has investigated the characteristics of the last seven ranked attributes using based on relationship with the target feature with expert opinion. According to expert opinion the least three attributes ranked in correlation with attribute evaluation work type, Residence type and smoking status have less important to stroke prediction. Figure 5.17 below shows correctly classified instance of both with all attribute and with selected attribute in two test option method.

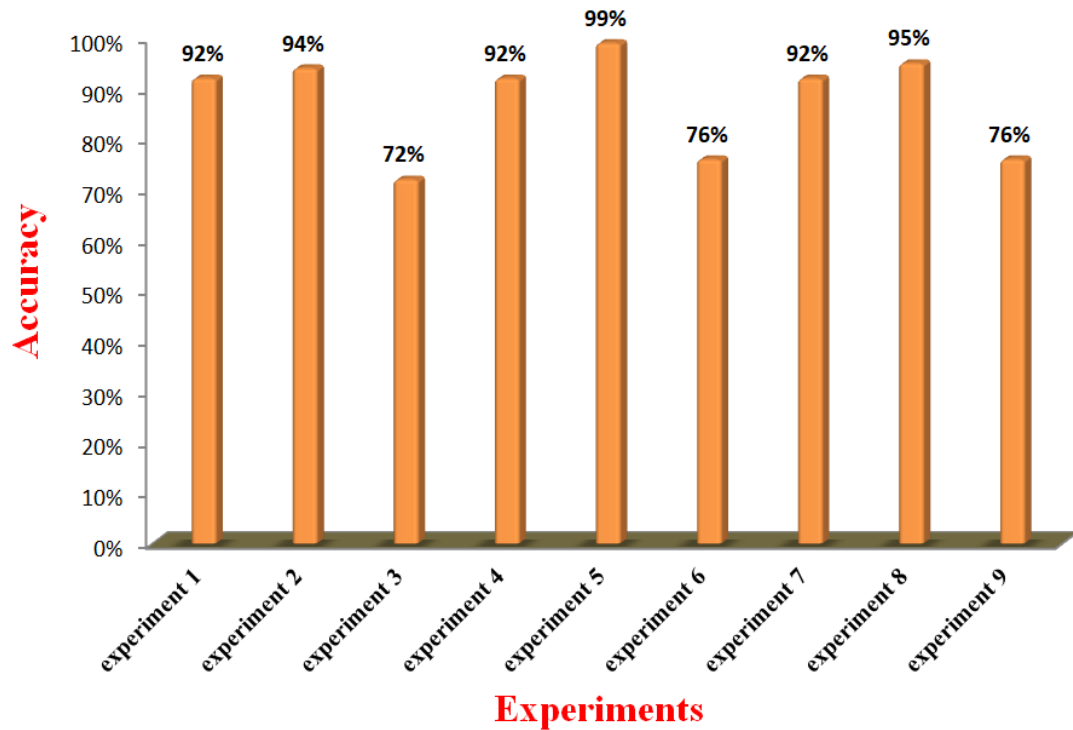


FIGURE 5.11: Experiment result comparison

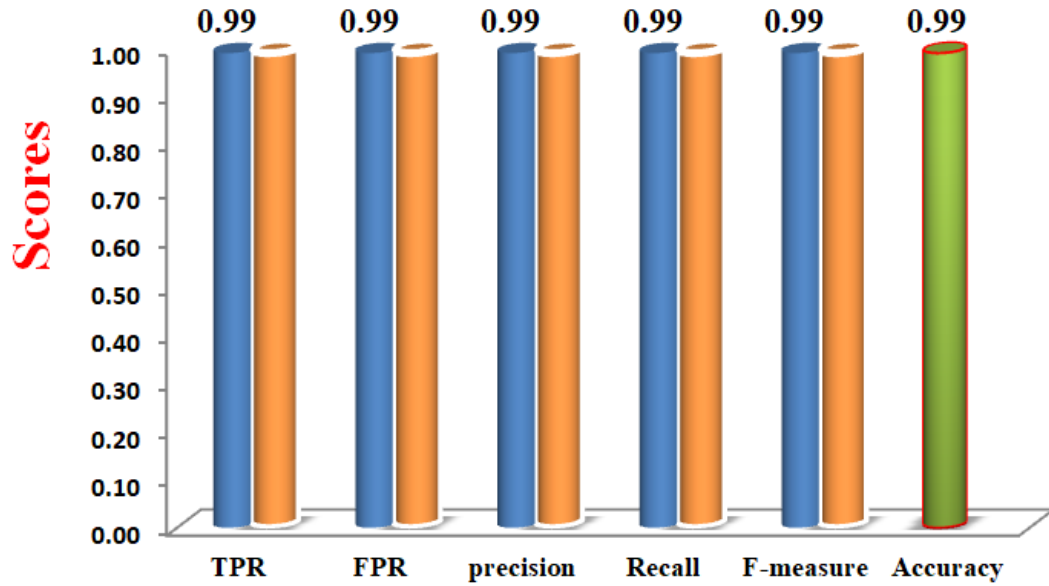
5.3.2 Model Selection

As shown in the above Figure 5.17 Random forest classifier models have 99% accuracy of correctly classified instance under 10-fold cross validation with selected attributes which is best performance than other models in the experiment. As discussed in chapter two to evaluate the performance of the classifier model Accuracy, True Positive, False Positive, Precision, Recall, and F-measure Area of developed model were used and in case random forest classifier models under 10-fold cross validation with selected attributes scores best result. So it is selected to develop Knowledge-based system. Figure 5.18 below shows the performance of selected classifiers with all performance matrix.

5.3.3 Error rate (Mis-classification) of the selected model

There are several metrics that can be calculated with Confusion Matrix and Error rate (Mis-classification) is the one. This measures how often the classifier has got the results wrong. That can be calculated by :

$$ERR = \frac{FP + FN}{TP + FP + TN + FN} \quad (5.1)$$



Performance Matrix

FIGURE 5.12: Random forest classifier performance with all performance matrix

or

$$ERR = 1 - Accuracy \quad (5.2)$$

Based on this idea the selected classifier model that identified from experiment 5 error rate are calculated as follows :

$$ERR = 1 - Accuracy$$

$$ERR = 1 - 0.99 = 0.01$$

The best error rate is 0.0, whereas the worst is 1.0 so based on this the selected classifier mode has best error rate 0.01.

5.4 RULE EXTRACTION

5.4.1 Rule Extraction from Random Forest Classifier with Selected Attributes Under 10-fold Cross Validation Test Option Methods

Random forest classifier with selected attributes under 10-fold cross validation appears to be the best model, among the chosen three different algorithms with

two test option mode. After having generated rules using the Random forest classifier algorithm, the next task is building or constructing the knowledge base. The overall task of this study is to extract rules that predict stroke from the public stroke prediction dataset using classification algorithms and integrate machine learning prediction with expert knowledge to develop knowledge-based system for stroke diagnosis and treatment. The followings are some sample rule extracted from dataset by Random forest algorithm.

Rule 1: $(bmi \leq 24) \text{And} (ever_married \leq 0) \text{And} (age \leq 2) \text{And} (avg_glucose_level \leq 73) \text{class} : 0.0$

Rule 2: $(bmi \leq 20) \text{And} (ever_married \leq 1) \text{And} (age \leq 2) \text{And} (avg_glucose_level \leq 73) \text{class} : 0.0$

Rule 3: $(bmi \leq 24) \text{And} (ever_married \leq 0) \text{And} (age > 2) \text{And} (avg_glucose_level > 73) \text{class} : 0.0$

Rule 4: $(bmi \leq 24) \text{And} (ever_married \leq 0) \text{And} (heart_disease \leq 0) \text{And} (age > 62) \text{class} : 1.0$

Rule 5: $(bmi > 20) \text{And} (ever_married \leq 0) \text{And} (heart_disease \leq 0) \text{And} (age > 62) \text{And} (avg_glucose_level \leq 74) \text{class} : 0.0$

Rule 6: $(bmi \leq 24) \text{Or} (bmi > 20) \text{And} (ever_married \leq 0) \text{And} (heart_disease \leq 0) \text{And} (age > 62) \text{And} (avg_glucose_level > 74) \text{class} : 1.0$

Rule 7: $(bmi > 24) \text{And} (ever_married > 0) \text{And} (heart_disease \leq 0) \text{And} (gender \leq 0) \text{And} (hypertension \leq 0) \text{And} (avg_glucose_level > 72) \text{And} (age > 78) \text{class} : 0.0$

Rule 8: $(bmi > 24) \text{And} (ever_married > 0) \text{And} (heart_disease \leq 0) \text{And} (gender \leq 0) \text{And} (hypertension > 0) \text{And} (age > 73) \text{And} (avg_glucose_level > 72) \text{class} : 1.0$

The above list of rules predict as instances 0(non-stroked) or 1(stroked) by Random forest rule induction classifier. As shown in rule #1 if patient BMI is less than or equals to 18 and he/she was not married and age is less than or equals to 2 and average glucose level is less than or equals to 73 then the patient was non-stroked(class 0). Based on the rules as previously discussed the number of instances

correctly classified by Random forest algorithm with selected attribute under 10-fold cross validation were 1930 (99.27%) and incorrectly classified instances of 14(0.73%) from a total of 1944 instances.

5.5 TESTING MODEL PERFORMANCE

After a model is trained by training dataset and setting different parameters it produces the behavior. This process poses challenges on lack of transparency, Indeterminate modeling outcomes, Generalizability, Unclear idea of coverage, and Resource needs. These issues make it difficult to understand the reasons behind a model's low performance, interpret the results, and assure that our model will work even when there is a change in the input data distribution (data drift) or in the relationship between our input and output variables (concept drift). To ensure that learned model will behave consistently and produce the results expect from it essentially machine learning models should be tested before used in the overall knowledge-based system development.

To performing this model test in this research work A/B testing is performed which is a type of split testing and is commonly used to drive improvements to specific variables or elements by measuring domain experts engagement. The researcher define around 10 rules that is generated by selected Random forest model with the selected feature under 10-fold cross validation. These set of rules are provided to experts and the performance of this rules are measured by comparing with the perspectives of experts.

The confusion matrix is used for comparing the performance of rules extracted from machine learning model with domain expert's results as shown in figure 5.19 below. Model performance testing mainly used to measure how accurate the model is through Precision, Recall, F-measure, True positive rate.

As described above the confusion matrix is shows the matrix of performance test of model by comparing with the insight of experts. Generally, the model has detected 9(90%) test set of rules as correct classifiers of the class of instance out of 10 test set of rules and 1 test set of rule are incorrectly classified which is 10%.

		selected model							
		class	0	1	class	precision	Recall	F-measure	Accuracy
Domain experts Decision	0	0	4	1	0	0.9	0.9	0.9	0.9
	1	1	0	5	1	0.9	0.9	0.9	0.9

confusion matrix
Model performance

FIGURE 5.13: Model Performance Testing

So as clearly illustrated this selected model has best performance in perspective of experts.

Chapter 6

DOMAIN EXPERT KNOWLEDGE EXTRACTION

The knowledge acquisition process for comprehensive knowledge-based systems involves extracting knowledge from domain experts. This process includes knowledge acquisition, knowledge modeling, and knowledge representation, which are discussed as follows.

6.1 KNOWLEDGE ACQUISITION METHOD

Knowledge acquisition refers to the process of gathering and incorporating information and expertise into a knowledge-based system [113]. There are several methods used for acquiring knowledge, and the choice of method depends on the nature of the system and the domain being modeled. The integration of various knowledge acquisition methods in stroke diagnosis and treatment, involving experts and real-world data, ensures a comprehensive understanding of the subject, thereby enhancing the accuracy and relevance of the knowledge-based system. When developing a knowledge-based system for stroke diagnosis and treatment, the following knowledge-acquisition methods are employed:

6.1.1 Expert Interviews:

In this study, we conduct expert interviews with neurologists, stroke specialists, and healthcare professionals to gain insights into decision-making processes and guidelines for diagnosing and treating stroke patients.

6.1.2 Medical Records Analysis:

In this study, we analyze stroke patient medical records to understand symptoms, diagnostic tests, treatment methods, and outcomes, providing valuable real-world data and experiences.

6.1.3 Clinical Guidelines and Research:

We review the clinical guidelines [114, 115], research papers [116], academic journals, and publications on stroke diagnosis and treatment [117], which aid in understanding evidence-based practices, advancements, and treatment recommendations.

6.1.4 Collaboration with Stroke Centers:

We collaborate with stroke centers and hospitals to gain firsthand knowledge and insights into their diagnostic and treatment processes.

6.1.5 Observation and Shadowing:

Observation and shadowing stroke specialists help us to capture tacit knowledge and understanding of decision-making, enhancing understanding of their interactions with patients and multidisciplinary teams.

6.1.6 Case Studies:

We study and analyze detailed case studies of stroke patients to gain in-depth knowledge of complexities and challenges faced in diagnosis and treatment.

6.1.7 Workshops and Conferences:

The researcher participate in workshops, conferences, and seminars on stroke diagnosis and treatment, engaging in discussions with experts to learn about latest advancements, emerging practices, and unresolved challenges.

6.2 EXPERT KNOWLEDGE MODELING

The study uses Expert Knowledge Modeling (EKM) to create a knowledge-based system for stroke diagnosis and treatment. It involves gathering and structuring stroke specialists' expertise, extracting relevant information, and presenting it in a structured form. This model aids in accurate diagnosis, treatment decisions, and improves stroke care quality and patient outcomes. The decision tree model (See Figure 6.1) analyzes patient data, provides accurate recommendations, and guides treatment decisions.

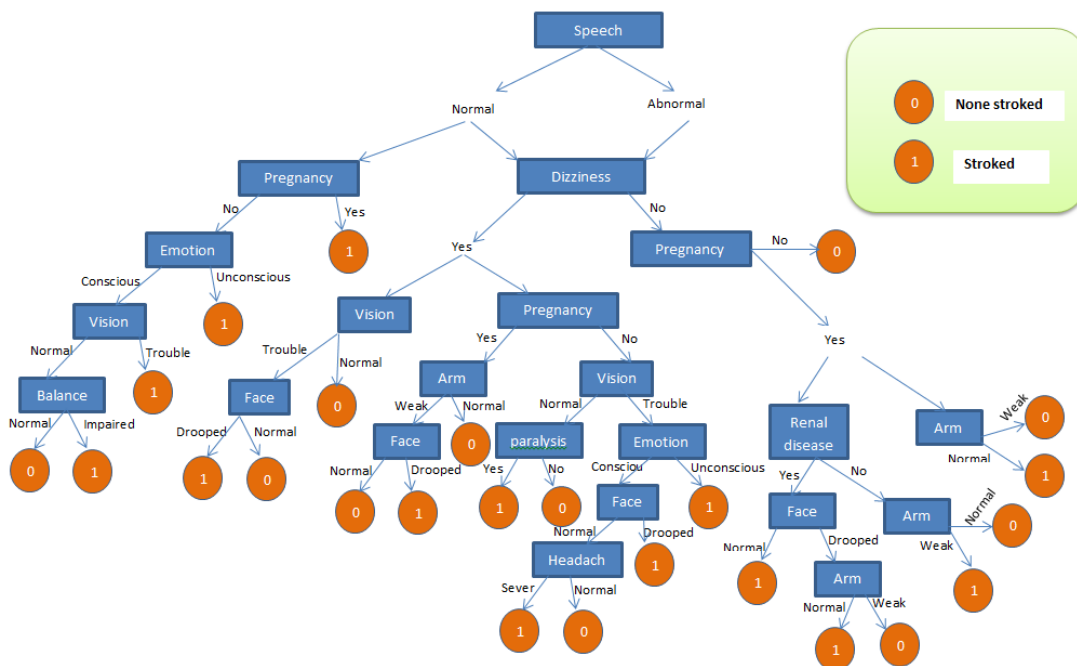


FIGURE 6.1: Decision Tree for stroke diagnosis and treatment acquire from domain expert

Experts detect strokes using neurological examinations, diagnostic imaging, and other tests. They perform tasks or answer questions to identify signs of brain function issues. The next step is to manage the stroke through appropriate diagnosis and treatment, ensuring proper management and recovery.

Healthcare providers often perform tests to suspect a stroke, including Computerized tomography (CT) scans, lab blood tests, Electrocardiogram (ECG/EKG), Magnetic resonance imaging (MRI) scans, and Electroencephalogram (EEG). These tests check for signs of infections, heart damage, clotting ability, blood sugar levels, kidney and liver function, and rule out seizures or related problems [118].

Treatment for strokes depends on the type of stroke. In ischemic strokes, restoring circulation to affected brain areas is the top priority, often involving medication like thrombolytics or catheterization. In hemorrhagic strokes, treatment depends on bleeding location and severity. Reducing blood pressure, improving clotting, or surgery may be necessary to stop bleeding and relieve pressure on the brain from accumulated blood [119].

6.3 EXPERT KNOWLEDGE REPRESENTATION

Knowledge representation is a crucial step in knowledge-based system development, using domain experts' rules in "IF-THEN" formats. Stroke Diagnosis and Treatment can utilize various techniques for expert knowledge modeling and representation.

In this study, a knowledge-based system for stroke diagnosis and treatment use techniques like rule-based systems and decision trees for expert knowledge modeling. Rule-based systems use "IF-THEN" rules to describe conditions and corresponding actions, while decision trees represent knowledge in a tree-like structure. These techniques help capture symptoms, risk factors, and medical guidelines, determining stroke likelihood and suggesting appropriate treatment options. The choice of these techniques depends on the system's specific requirements, available data, expert knowledge, and intended use and context.

6.3.1 Rules extracted from expert knowledge

Rule 1: IF a patient is with normal speech and none pregnant and emotionally conscious and normal vision and normal balance THEN Patient = none stroked.

Rule 2: IF a patient is with normal speech and none pregnant and emotionally conscious and normal vision and impaired balance THEN Patient = stroked.

Rule 3: IF a patient is with normal speech and none pregnant and emotionally conscious and trouble vision THEN Patient = stroked.

Rule 4: IF a patient is with normal speech and none pregnant and emotionally unconscious THEN Patient = stroked.

Rule 5: IF a patient is with normal speech and pregnant THEN Patient = stroked.

Rule 6: IF a patient is with normal speech and dizziness and trouble vision and drooped face THEN Patient = stroked.

Rule 7: IF a patient is with normal speech and dizziness and trouble vision and normal face THEN Patient = none stroked.

Rule 8: IF a patient is with normal speech and dizziness and normal vision THEN Patient = none stroked.

Rule 9: IF a patient is with normal speech and dizziness and pregnant and normal arm THEN Patient = none stroked.

Rule 10: IF a patient is with normal speech and dizziness and pregnant and weak arm and normal face THEN Patient = none stroked.

Rule 11: IF a patient is with normal speech and dizziness and pregnant and weak arm and drooped face THEN Patient = stroked.

Rule 12: IF a patient is with normal speech and dizziness and none pregnant and normal vision and paralysis THEN Patient = stroked.

Rule 13: IF a patient is with normal speech and dizziness and none pregnant and normal vision and no paralysis THEN Patient = none stroked.

Rule 14: IF a patient is with normal speech and dizziness and none pregnant and trouble vision and emotionally unconscious THEN Patient = stroked.

Rule 15: IF a patient is with normal speech and dizziness and none pregnant and trouble vision and emotionally unconscious THEN Patient = stroked.

Chapter 7

KNOWLEDGE-BASED SYSTEM

This study examines the use of knowledge-based systems for stroke diagnosis and treatment, focusing on rule-based knowledge representation. The study reveals that rule-based knowledge representation is useful in various domains due to its common use and powerful application capabilities, particularly in encoding expert knowledge into rules for symptom, test, and treatment recommendations. It uses empirical if-then rules to represent expert knowledge.

The study uses SWI-Prolog and Sublime Text editor to develop a knowledge base from experts and a GUI. Python is used for classifier algorithms for machine learning prediction based on public datasets. Jupiter notebook is used for writing documents with explanatory text, equations, visualizations, and live codes. Flask Python-based back-end micro framework is used for developing applications and websites using HTML. The knowledge base is accessed through SWI-Prolog and Sublime Text editor.

7.1 MAPPING KNOWLEDGE TO KNOWLEDGE-BASED SYSTEM

The study uses machine learning techniques and domain experts to create a knowledge-based system for stroke diagnosis and treatment, i.e., it acquires knowledge from expert and machine learning techniques, merges it, and uses the combined knowledge for knowledge-based system construction. To integrate this extracted knowledge with a knowledge-base system, the researcher constructs a framework that reads machine learning predictive models and expert knowledge, generating Prolog rules for the system. This integration of implicit knowledge is automatically performed.

The study aims to develop a Stroke rule-based knowledge-based system using predictive and expert knowledge. The system uses predictive and expert knowledge

to form rules, which are evaluated and filtered based on exposure measurement. The generated rules are then transformed into facts for the inference engine. The system is designed to support health professionals in early decision-making in stroke diagnosis and treatment processes. The Random Forest rule induction classification algorithm is used for its greater accuracy in building the model. All previous experiments are based on the random forest classification algorithm.

7.1.1 Structure of Random Forest and Prolog Rule

This study focuses on developing a rule-based, knowledge-based system for stroke diagnosis and treatment using machine learning techniques and experts' knowledge. The system uses a Python machine learning algorithm and SWI-Prolog Multi-Treated Versions for modeling and evaluating. From the three algorithms investigated, the Random Forest algorithm, with 10-fold cross-validation, is chosen as the best due to its ability to generate meaningful rules in the form of "IF... THEN...", which includes attributes, comparison operators, and values, as it is illustrated in Table 7.1.

Parameter	Random forest model	Prolog
white space	(" ")	("_")
AND	AND	(";")
OR	OR	(";")
=	=	("=")
>	>	>
<	<	<
>=	>=	>=
<=	<=	<=
Period	". "	". "
If	If	:-

TABLE 7.1: Attributes, comparison operators, and values in knowledge mapping

In this study, mapping that predictive and domain expert knowledge automatically to a knowledge-based system is challenging because the SWI-Prolog knowledge

base uses declarative and logical language, while machine learning uses object-oriented language. To address this, the researcher constructs a Python-based framework as a bridge, calling the random forest classification algorithm and writing Prolog rules simultaneously. The framework reads machine learning files and automatically generates Prolog facts and rules, integrating the prediction model into the knowledge base.

The rules are in "IF... Then" format and it can be interpreted as follows:

IF patient (*BMI* <= 24)*AND*(*ever_married* <= 0)*AND*(*heart_disease* <= 0)*AND*(*age* > 18) **THEN** class:1

However, Prolog does not understand "IF...THEN" format, it works in reverse order. Prolog starts with a goal and then goes to the facts that can prove the goal to be true. Therefore, the above rule has to be formatted as:

1 :- (*BMI* <= 24), (*ever_married* <= 0), (*heart_disease* <= 0), (*age* > 18).

As illustrated above, the conclusion came first with predicate class and followed by ":-" replacing ":" in random forest and then antecedents joined by "," replacing "And" in random forest rule. Finally, Prolog rules terminate by period (.) and the rule obtained from domain experts also first represented in the form of prolog rules and to implement this rule, a random forest rules format is used to develop the knowledge-based system.

In general, reversing the order of rules is needed to represent the rules generated from domain expert (THEN...IF format) into random forest (IF...THEN format) to follow Python syntax. To integrate the rule gained from the domain expert, which is written in prolog, and the rule extracted from the machine learning random forest classifier, the researcher imported the Prolog library from the PySwip package, which is used to interpret the prolog program in the syntax of python. The way of reversing is performed as follows:

For Example :

stroked:- speech = normal, dizzy = yes, pregnant = yes, arm = weak, face = drooped.

is sample rule set from domain expert knowledge; when it mapped to python "IF... THEN" format it beings

*IF a patient is with normal speech AND dizziness AND pregnant AND weak arm
AND drooped face THEN Patient = stroked.*

7.2 KNOWLEDGE BASE CONSTRUCTION

Chapter 3 outlines the process of developing a knowledge-based system for stroke diagnosis and treatment. The first step involves extracting knowledge from data and domain experts. After extracting knowledge, the knowledge engineer has to represent this knowledge in an appropriate format that is easily understandable by the computer system using rule-based and develop the knowledge-based system using the knowledge extracted from domain experts and data. The developing knowledge-based system contains a major module, namely, knowledge base, inference engine, explanation facility, and user interface.

7.2.1 Knowledge Base

Knowledge base [120] holds the essential rules for solving a specific domain problem. It stores all relevant facts, rules, and relationships used by the rule-based system. In other words, a rule base is a set of rules or encoded knowledge about stroke diagnosis and treatment of the prototype system.

In this study, all knowledge that is acquired from domain experts and machine learning was stored in the knowledge base. Basically, we have two pieces of knowledge-based:

- ❖ **Knowledge base A:** It is the one acquired from domain experts by gathering and incorporating information and expertise using experts' insights to decision-making processes for diagnosing and treating stroke patients, analysis of patient medical records to understand symptoms, diagnostic tests, treatment methods, and outcomes, review of clinical guidelines and research which aid in understanding evidence-based practices in stroke diagnosis and treatment recommendations, and observation and shadowing which help to

capture knowledge and understanding of their interactions with patients and multidisciplinary teams. Initially, this knowledge base was constructed by using prolog in "THEN...IF" format as described below:

- **Rule 1:** non-stroked :- speech = normal, pregnant = no, emotion = conscious, vision = normal, balance = normal.
- **Rule 2:** stroked :- speech = normal, pregnant = no, emotion = conscious, vision = normal, balance = impaired.
- **Rule 3:** stroked :- speech = normal, pregnant = no, emotion = conscious, vision = trouble.
- **Rule 4:** non-stroked :- speech = normal, dizzy = yes, vision = trouble, face = normal.
- **Rule 5:** stroked :- speech = normal, dizzy = yes, pregnant = yes, arm = weak, face = drooped.
- **Rule 6:** non-stroked :- speech = normal, dizzy = yes, pregnant = yes, vision = normal, paralysis = no.

❖ **Knowledge base B:** The second knowledge base comes from machine learning. Machine learning [121] relies on effective learning methods, including rich and extensive datasets. Data is crucial for machine learning, making data analytics highly valuable in this field.

Predictive analytics is used in this study to forecast future results based on historical and current data, employing various techniques such as predictive models. Predictive modeling [122], a type of classification, examines historical data to identify trends or patterns and uses these insights to predict future outcomes. Essentially, it aims to answer the questions, "Have I encountered this before?" and "What usually follows this pattern?" In this research, Random Forest outperforms the other two common predictive models based on experiments (refer to Chapter 5) and selected attributes under 10-fold cross-validation.

This predictive model extracts a set of rules in "IF...THEN" format, as presented below sample rule :

Rule 1: $(bmi \leq 24) \text{OR} (bmi > 20) \text{AND} (evermarried \leq 0) \text{AND}$
 $(heartdisease \leq 0) \text{AND} (age > 18) \text{class} : 1$

Rule 2: $(bmi \leq 24) \text{OR} (bmi > 20) \text{AND} (evermarried \leq 0) \text{AND}$
 $(heartdisease \leq 0) \text{AND} (age > 62) \text{AND} (avg_glucoselevel \leq 74) \text{class} :$
 0

Since the rule-based approach applies to the knowledge of human experts, knowledge can be captured in the form of "IF...THEN" rules and facts. A rule-based representation technique represents the validated knowledge gained from both domain experts and machine learning prediction, and the rules are codified to the knowledge base of the prototype system using Python programming language by changing the prolog formatted knowledge of domain experts to Python as a machine learning knowledge format because Python is more interactive to create graphic user interface to be the developed system more user friend.

7.2.2 Inference Engine

An inference engine [123] is the brain of the Knowledge-Based System, which directs the system on how it can derive a conclusion by looking for possible solutions from the knowledge base and recommending the best possible solution. It consists of an inference mechanism and control strategy that enables deriving a conclusion for a given query. It comprises formal reasoning involving matching and unification, similar to the one performed by human experts to solve problems in a specific area of knowledge.

The rule base developed by Python is one of the significant parts of the system prototype. As discussed in the above section, this rule base integrates Prolog and Python formatted rules connected by pyswip by loading the knowledge base using the Prolog consult predicate, a built-in predicate in standard Prolog.

The backward chaining mechanism is used in this study during the system prototype development. When building a Backward Chaining system, start with the highest-level rules and add additional detailed rules. At the highest level, the system has one rule; Such inferring mechanisms are discussed below with sample rule inferring.

1:- $(bmi \leq 24)OR(bmi > 20)AND(evermarried \leq 0)AND$
 $(heartdisease \leq 0)AND(age > 18)$.

Typically, a command in the system defines the initial Top-Level Goal. In this case, it is: "Determine if the class should be 1." The system looks through the rules (only 1 rule so far) to find rules with the top goal in the "THEN" part. This rule is tested since it could potentially set the value for the goal. To determine if the relevant rule is true, and can set a value for the goal, the system must determine whether the "IF" conditions are true. That requires determining whether "patient $(bmi \leq 24)OR(bmi > 20)AND(evermarried \leq 0)AND(heartdisease \leq 0)AND(age > 18)$." which becomes the new Top-Level Goal.

The inference engine temporarily stops trying to set a value for the "Determine if the class should be 1." goal, and concentrates on the new top Goal, "patient $(bmi \leq 24)OR(bmi > 20)AND(evermarried \leq 0)AND(heartdisease \leq 0)AND(age > 18)$ ".

Since there are no other rules in the system, there is no way of deriving the value so the system must ask the end user. Once the user answers the question, the system knows the value for "patient $(bmi \leq 24)OR(bmi > 20)AND(evermarried \leq 0)AND(heartdisease \leq 0)AND(age > 18)$ " and that goal drops off the Goal list. The Goal list returns to the original goal of "determining if the class should be 1".

If the system determines that this is a "patient $(bmi \leq 24)OR(bmi > 20)AND(evermarried \leq 0)AND(heartdisease \leq 0)AND(age > 18)$ " the one rule in the system determines the value for that Goal, and the session is complete. If it cannot determine that this is a "patient $(BMI \leq 24), (ever_married \leq 0), (BMI > 20), (heart_disease \leq 0), (age > 62), (BMI \leq 22), (age > 18)$ " there are no rules in the system for setting a value for the "Determine if the class should be 1." variable.

Further more for class "0" rule inferring it works in the same way as follows:

0:- $(bmi \leq 24)OR(bmi > 20)AND(evermarried \leq 0)AND$
 $(heartdisease \leq 0)AND(age > 62)AND(avg_glucoselevel \leq 74)$.

A command in the system defines the initial Top-Level Goal. In this case, it is: "Determine if the class should be 0." The system looks through the rules to find rules with the top goal in the "THEN" part. This rule is tested since it could potentially set the value for the goal. To determine if the relevant rule is true, and can set a value for the goal, the system must determine whether the "IF" conditions are true. That requires determining whether "patient ($bmi \leq 24$)OR($bmi > 20$)AND($evermarried \leq 0$)AND($heartdisease \leq 0$)AND($age > 62$)AND($avg_glucoselevel \leq 74$)." which becomes the new Top-Level Goal.

The inference engine temporarily stops trying to set a value for the "Determine if the class should be 0." goal, and concentrates on the new top Goal, "patient ($bmi \leq 24$)OR($bmi > 20$)AND($evermarried \leq 0$)AND($heartdisease \leq 0$)AND($age > 62$)AND($avg_glucoselevel \leq 74$)." .

Since there are no other rules in the system, there is no way of deriving the value so the system must ask the end user. Once the user answers the question, the system knows the value for "patient ($bmi \leq 24$)OR($bmi > 20$)AND($evermarried \leq 0$)AND($heartdisease \leq 0$)AND($age > 62$)AND($avg_glucoselevel \leq 74$)." and that goal drops off the Goal list. The Goal list returns to the original goal of "determining if the class should be 0" .

If the system determines that this is a "patient ($bmi \leq 24$)OR($bmi > 20$)AND($evermarried \leq 0$)AND($heartdisease \leq 0$)AND($age > 62$)AND($avg_glucoselevel \leq 74$)." the one rule in the system determines the value for that Goal, and the session is complete. If it cannot determine that this is a "patient ($bmi \leq 24$)OR($bmi > 20$)AND($evermarried \leq 0$)AND($heartdisease \leq 0$)AND($age > 62$)AND($avg_glucoselevel \leq 74$)." there are no rules in the system for setting a value for the "Determine if the class should be 0." variable.

7.2.2.1 Prioritized Inferring

Prioritized inferring from two knowledge bases in a knowledge-based system (KBS) is making inferences and prioritizing them based on their importance or relevance to a particular task or problem. After identifying the two knowledge bases used for inference and aligning them by identifying the common concepts and relationships

between them, the researcher uses rule-based reasoning to generate inferences from them.

Inferences are Prioritized based on their relevance, or importance, to the task or problem. In this research work, the researcher highly prioritizes domain expert knowledge base over machine learning knowledge base based on relevance. As a result of this, every inferring task is started from domain expert knowledge, and if the inference engine gets the correct match, the goal would be executed unless the inference engine continues to infer from the machine learning knowledge base.

A pseudocode of our inferring in a structured and readable format is as follows:

Algorithm 10 pseudocode for Prioritized Inference algorithm

function prioritize_inferences(knowledge_bases)

- 1: **Step 1:** Identify the goals and objectives of the system
 - 2: goals = get_goals_and_objectives()
 - 3: **Step 2:** Evaluate the reliability and credibility of the knowledge bases
 - 4: reliability_scores = evaluate_reliability(knowledge_bases)
 - 5: credibility_scores = evaluate_credibility(knowledge_bases)
 - 6: **Step 3:** Determine the relevance of the inferences to the goals and objectives
 - 7: relevance_scores = evaluate_relevance(inferences, goals)
 - 8: **Step 4:** Consider the confidence levels of the inferences
 - 9: confidence_scores = evaluate_confidence(inferences)
 - 10: **Step 5:** Evaluate the potential impact of the inferences
 - 11: impact_scores = evaluate_impact(inferences)
 - 12: **Step 6:** Use a weighted scoring system to prioritize the inferences
 - 13: weights = [relevance_scores, confidence_scores, impact_scores]
 - 14: prioritized_inferences = sort_inferences_by_weighted_scores(inferences, weights)
 - 15: **Endfunction**
-

7.2.3 User Interface

The user acceptance of a knowledge-based system is influenced by the quality of the user interface, which serves as the communication channel between the user and the system. The researcher proposes a knowledge-based system for health workers, including interns and general practitioners, with a simple graphical user interface and Command Line Interface. To interact with users, a web application and a flask application are developed. Using user input, the web page uses simple

HTML code to predict stroke class. The application uses the user’s input values when clicking the ”predict” button, as illustrated in Figure 7.2.

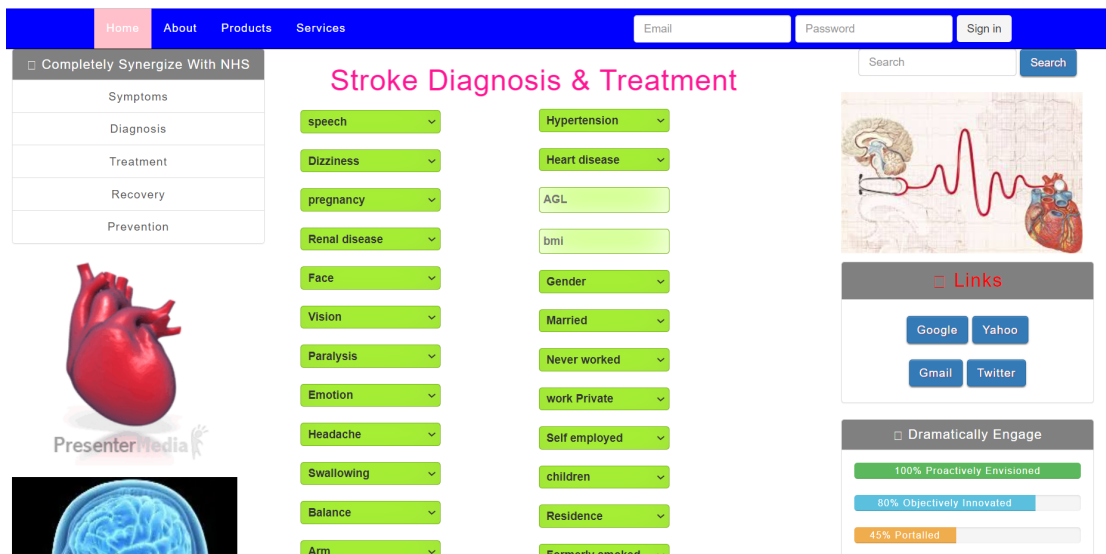


FIGURE 7.1: User interface of proposed system

The flask application is basically a python code which is a bridge between the web page and the trained machine learning model and experts knowledge as shown in Figure 7.3. The input values are sent to the flask application which in sends the values to the model for prediction.

7.2.4 Explanation Facility

The explanation facility is part of the reasoning-based module, which explains the case if the user needs further explanation about a diagnosis and treatment implemented in the system. In steps towards determining the class of stroke and explaining how to manage or treat it, This module is incorporated into the user interface as Help controls. When the user clicks the ”Diagnosis” button, the entire parameter is sent to the flask application, and based on the parameters; the recommended diagnosis will be provided as output. In the same way when the user clicks the ”Treatment” button, the entire parameter is sent to the flask application, and based on the parameters that can determine the type of stroke, the recommended treatment will be provided to users as output. Figures 7.4 and 7.5 show sample facilities for stroke diagnosis and treatment, respectively.

```
# Load the Lib
from flask import Flask, render_template, request
import numpy as np
import pickle
from pyswip import Prolog
import time
import json

# Load the Random Forest Classifier model
app = Flask(__name__)
model = pickle.load(open('pppp.pkl', 'rb'))

#read file that load prolog KB to append
def read_file(pytholog):
    fh = open(pytholog, "a")
    try:
        return fh.read()
    finally:
        fh.close()

|

@app.route ( '/' )
def home():
    return render_template('checkup.html' )
```

FIGURE 7.2: Flask Application Linking the Model and Web Page.

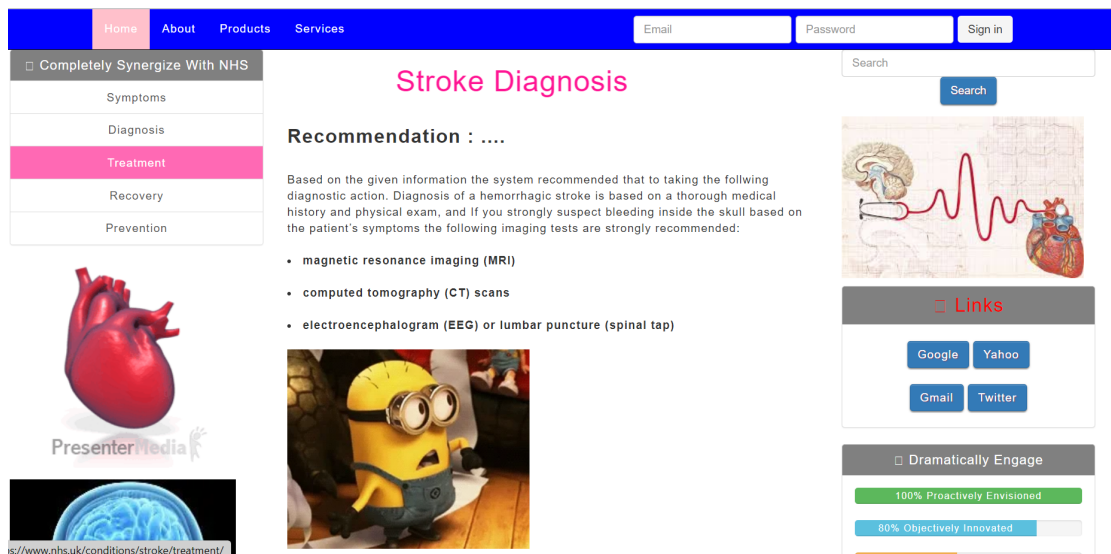


FIGURE 7.3: Stroke additional Diagnosis facilities page

7.3 EVALUATION OF SYSTEM

Evaluation [124] verifies whether the proposed system meets the objectives set by the researcher and the user’s requirements. Before the developed knowledge-based system deploys the user in the real operative environment, the system must be evaluated by usability and performance testing. Therefore, in this study, the

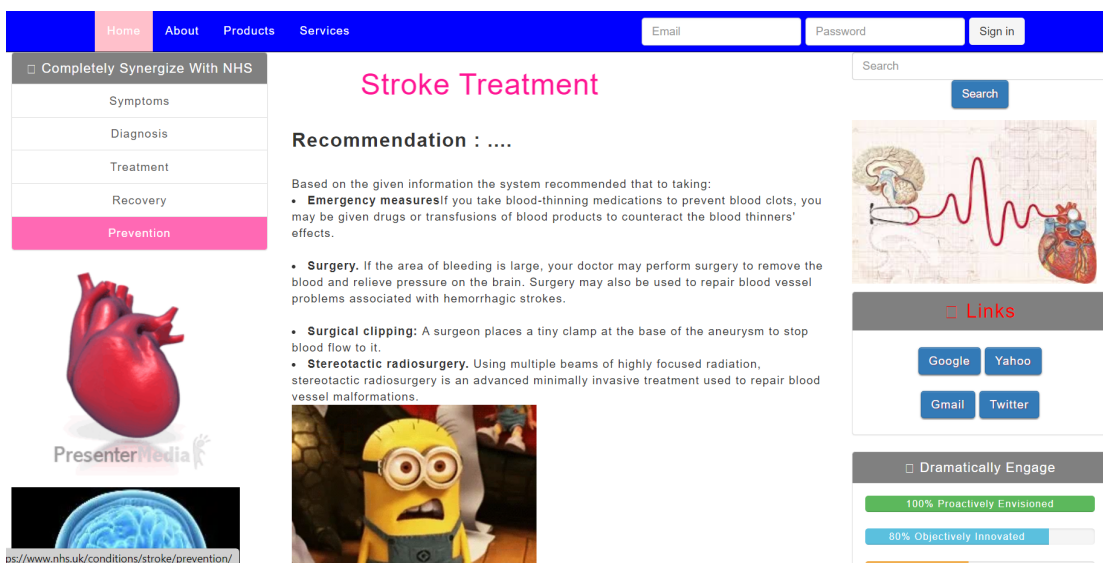


FIGURE 7.4: Stroke treatment facilities page

evaluation of the prototype system has two aspects. These are system performance testing and user acceptance testing. One way of evaluating the knowledge-based systems is by preparing test cases and feeding the proposed system with these test cases. Give the same test cases to domain experts and compare the results of the proposed system and the domain experts to ensure that the proposed system could replace the expert in his or her absence. In addition to this, evaluation can be done by conducting user acceptance testing, which will help to make sure whether the proposed system is user-friendly and whether the proposed system could replace the domain experts or not.

7.3.1 System performance testing using test cases

Test cases are one of the major evaluation mechanisms for evaluating the performance of the proposed system, which helps to compare and contrast the domain expert’s judgment and the proposed system’s response so that the researcher can conclude whether the proposed system could work in the absence of the expert or not. The test cases include samples of stroke instances taken from the DBRH manually recorded system.

To prepare the test cases, the researcher divides the function of the proposed system into two categories: clinical diagnosis and treatment. The test cases are unlabeled and delivered to domain experts to label them as stroke or non-stroke

class and used for recommending proper diagnosis, and the treatment determines what type of therapy should be given to the patient based on the type of stroke identified from the test cases. About 41 instances, including 15 attributes, are taken for the test. These test instances are provided to the combined stroke diagnosis and treatment system, and the outputs are compared to the domain expert’s decision.

The confusion matrix is used to compare the performance of the knowledge-based system with the domain expert’s results, as shown in Figure 7.6 below. System performance testing is mainly used to measure how accurate the system is through Precision, Recall, F-measure, and True positive rate.

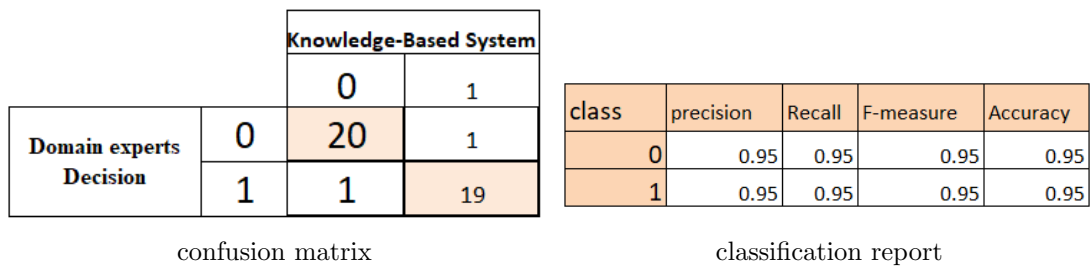


FIGURE 7.5: Performance evaluation of proposer system with test case

The confusion matrix in Figure 7.6 shows a matrix of test cases evaluation by stroke diagnosis and treatment system and domain experts’ decision. The rows illustrate the evaluation of the domain experts, and the columns illustrate the results of the stroke diagnosis and treatment systems. Generally, the system has detected 39 test instances as correct classes out of 41 test cases, and 2 instances are incorrectly classified, which is 5%. This justifies the overall representation of the proposed system, which showed 95% detection accuracy for all stroke types. But, this measure alone is not enough to measure the performance of the knowledge base system since it only tells us the overall performance. As clearly illustrated in Figure 7.7 below, the system’s performance is evaluated in terms of accuracy, Precision, Recall, and F-measure, which enables us to view in detail how accurate the system is in stroke diagnosis and treatment. This result is encouraging for using the system.

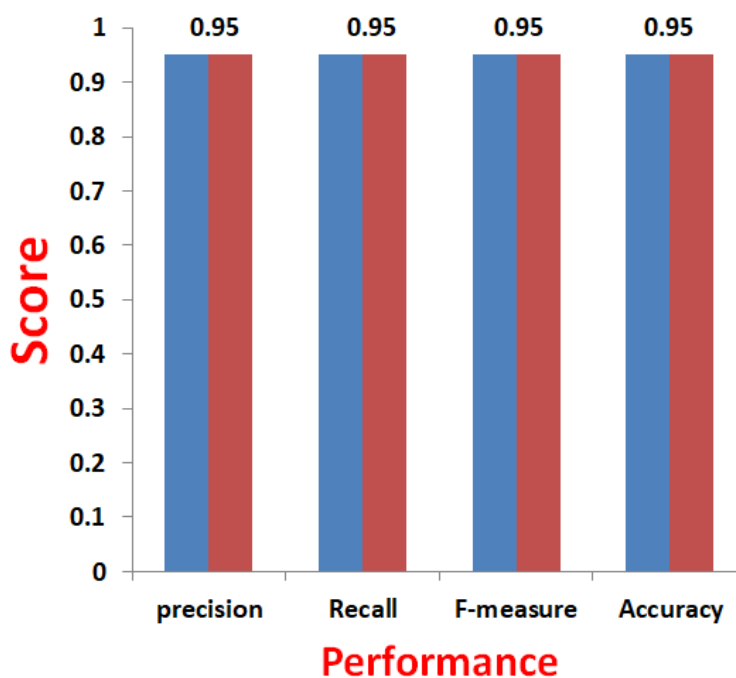


FIGURE 7.6: Performance of developed system

7.3.2 User Acceptance Testing

The other way of evaluating the Knowledge-Based System is user acceptance testing through which to evaluate whether the potential users would like to use the proposed system or not since this study is conducted to be used by potential users and to check whether the proposed system fulfills users requirements and ensures that the proposed system would be operational and usable by end-user. Five health workers (2 neurologists and 3 general practitioners) were selected to test the system from DBRH. During the knowledge-based system development, these domain experts were actively involved in the different stages of pilot study, knowledge acquisition, prototype development, and consulting on the content of knowledge. The informal discussion with domain experts has a significant role in understanding the dimension of the problem. Finally, experts get training about how the system works, and the test cases are given to evaluate the system.

Even if there are different types of user acceptance testing evaluation criteria in this study, the researcher uses questionnaires to evaluate the model called ResQue (Recommender systems quality of user experience) from the user's point of view to test the performance of the prototype system by domain experts. Questions are

close-ended, which helps system evaluators to check on the user interface design aspects, easiness of the system to use, attractiveness, the correctness of the decision, adequacy of knowledge content, problem-solving ability, and significance of knowledge-based system in triage patient categorization service. Questionnaires are found in the appendix part of this document. The weight scale for this study has been used such that Excellent = 5, Very Good = 4, Good = 3, fair = 2, and Poor = 1. The values indicate a number of evaluators who evaluate the system as Poor, Fair, Good, Very Good, and Excellent concerning evaluation criteria. Thus, this method helps the researcher to manually examine the user acceptance based on evaluator's response. The user acceptance of the system is measured manually as follows:

$$AVP = SV1 * \frac{nr1}{tnr} + SV2 * \frac{nr2}{tnr} + \dots SV_n * \frac{nr_n}{tnr} \quad (7.1)$$

Where, SV = scale value, TNR = total number of respondent and NR = is number of respondent. To get the result of user acceptance average performance is calculated out 100%.

$$AVP = (SV1 * \frac{nr1}{tnr} + SV2 * \frac{nr2}{tnr} + \dots SV_n * \frac{nr_n}{tnr}) * 100/NS \quad (7.2)$$

Where NS is number of scale and AVP is average performance.

The evaluators assessed the system by using the criteria listed in the following table figure 7.8 below. As clearly stated in the table, the domain experts evaluate the proposed system and rate them by their point of view concerning the criteria given above. For the first criterion, out of five evaluators, one evaluator rated the simplicity of the proposed system as Excellent, three evaluators rated it as Very Good, and the remaining one evaluator rated it as Good. This tells us 20% of the total evaluators rated as Excellent, 60% of the total evaluators rated as Very Good, and the remaining 20% rated the simplicity of the Knowledge-Based System as Good.

For the second criterion, out of five evaluators, four rated the proposed Knowledge-Based System as Excellent, and the remaining one rated it as Very Good. This tells us 80% of the total evaluators rated Excellent, and the remaining 20% rated the

No	Criteria's for evaluating the proposed KBS	Poor	Fair	Good	V. Good	Excellent	Average
1	Simplicity of KBS			1	3	1	4
2	Does the proposed system contain the required symptoms to stroke diagnosis and treatment ?				1	4	4.8
3	Efficiency of the proposed system relative to its response time ?					5	5
4	Does the proposed system recommend appropriate diagnosis and treatment for stroke?			1	1	3	4.4
5	Is the user interface suitable for the user?			1	2	2	4.2
6	Would you like to use the KBS frequently?			1	3	1	4
7	Is the system effective in diagnosis and treatment of stroke?				2	3	4.6
8	How do you rate the importance of the proposed system?				3	2	4.4
Total Average							4.4

FIGURE 7.7: User acceptance criteria with their corresponding answer

model as Very Good. For the third criterion, all of the evaluators rate the efficiency of the proposed Knowledge-Based System as Excellent because the system takes only a fraction of a seconds to display the results. This tells us 100% of the total evaluators rate the response time of the system as Excellent.

For the fourth criterion, out of five evaluators, three rated the proposed system as Excellent, one rated it as Very Good, and the remaining evaluator rated it as Good. This tells us 60% of the total evaluators rated Excellent, 20% of the evaluators rated as Very Good, and the remaining 20% rated the proposed system as Good.

For the fifth criterion, out of five evaluators, two evaluators got the user interface suitable for use and rated it as Excellent, two evaluators rated it as Very Good, and the remaining evaluators rated it as Very Good. This tells us 40% of the total evaluators rate Excellent, 40% of the evaluators rate as Very Good, and the remaining 20% rated the system as Very Good.

For the sixth criterion, out of five evaluators, one evaluator rates the Knowledge-Based System as Excellent, three evaluators rate as Very Good, and the remaining one evaluator rate as Very Good. This tells us 20% of the total evaluators rated

Excellent, 60% of the evaluators rated as Very Good, and the remaining 20% rated as Good.

For the seventh criterion, out of five evaluators, three evaluators found the proposed system effective in stroke diagnosis and treatment and rated it as excellent, and the remaining two evaluators rated it as very good. This tells us 60% of the total evaluators rate Excellent, and the remaining 40% rate the effectiveness of the system as Very Good.

For the eighth criterion, out of five evaluators, two evaluators rate the model as Excellent, and the remaining three evaluators rate it as Very Good. This tells us 40% of the total evaluators rate Excellent, and the remaining 60% rate the Knowledge-Based System as Very Good.

From the above evaluation, the researcher understands that the proposed Knowledge-Based System works well because it contains the necessary knowledge required for stroke diagnosis and treatment. As shown in Figure 7.8, all evaluators liked the speed of the Knowledge-Based System in stroke diagnosis and treatment; generally, 62.5% of the evaluators rated the proposed Knowledge-Based System as Excellent, 32.5% of the evaluators' rate as Very Good and the remaining 5% rated as Good.

To summarize Figure 7.8 above, based on the responses of five system evaluators, the average performance obtained is 4.4 on a scale of 5. This value is the result obtained from the values assigned for each close-ended question. The result indicates that about 88% of users are satisfied with the performance of the knowledge-based system. It means that the proposed knowledge-based system gains about 88% of user acceptance.

7.4 DISCUSSION AND RESULTS COMPARISON

In this study, a knowledge-based system was developed using machine learning integrated with expert Knowledge. This makes the developed knowledge-based system different from the previous study. As pointed out in previous sections, the accuracy of the prototype system is 95%, and the performance evaluation result by the domain experts is 88%, respectively. Based on the conducted research, in

this section, the researcher will discuss insight into the result in relation to the research objectives itemized in Chapter One. As a dispatch, the major objective of this proposed work was to construct a System for stroke diagnosis and treatment through integrating Knowledge acquired from domain experts and machine learning to increase the effectiveness and efficiency of the system, and this thesis work attempts to answer the list of research questions below.

In this work, the researcher identifies the most determinant factors of stroke diagnosis and treatment from both public datasets and domain experts. From the dataset based on explanatory analysis and correlations of different variables with the target variable's seven variables, age, hypertension, average glucose level, heart disease, ever married, body mass index, and gender are the most determinate factors for stroke prediction. In the same considerations of domain expert knowledge acquired by in-depth interviews and relevant document analysis, the recognition of face, arm, speech, emotion, dizziness, and paralysis are some of the most common signs and symptoms of stroke diagnosis and treatment.

Based on nine conducted experiments with two scenarios, one is with all attributes and the second with selected attributes using three classifier algorithms counting decision tree, random forest, and support vector machine under ten-fold cross-validation and percentage split the random forest algorithm with selected attributes under 10-fold- cross-validation is the best classification algorithm to develop the prediction model that can predict stroke because it performs better performance with 99% Precision, Recall, F1-score and accuracy so the researcher decided to use the results for further use in the development of knowledge-based system.

After the researcher stores all extracted Knowledge that was collected from domain experts as a set of rules using production rule by using a prolog programming language with SWI-Prolog 7.6.4 open-source software understandable format and converted to python-format to integrate with the rule extracted from random forest classifier model; Several rules are generated by the random forest algorithm in machine learning and by decision tree rule induction algorithm from the domain

experts to develop knowledge-based system. Finally, python programming language with anaconda distribution navigator Jupiter notebook used to construct the rule base module, HTML with sublime text editor used to develop GUI then to call (use) the rule from the knowledge base connecting the Knowledge base that is constructed with python to HTML graphical user interface by added pyswip file in python library serve as an interface between Python and Prolog.

To sum-up, the evaluation results of system performance testing and user acceptance testing showed that the proposed system registers better performance. As shown in Table 7.1, comparing the result achieved by the proposed system with prior research works helps to show the difference in terms of the accuracy of the used algorithm for achieving their research objective. The performance of the proposed system for stroke diagnosis and treatment is 99.27% accuracy. This demonstrates a good boost and enhancement in stroke diagnosis and treatment. Table 6.1 illustrates the summary of performance comparison results with previous studies.

No	Reference study's	Algorithm	Accuracy
1	Cheon S. et al. [75]	DNN	84.03%
2	Zhang S. et al. [76]	SSD	89.77%
3	Tazin T. et al. [31]	RF	96%
4	Sailasya G. and Kumari GLA. [77]	Naive Bayes	82%
5	Thammaboosadee S. and Kansadub T.[78]	ANN	84%
6	Almadani O. and Alshammari R.[79]	C4.5	95.25%
7	Alberto J. et al. [80]	RF	92%
8	Chun M. et al.[81]	ensemble (RSF and GBT)	80%
9	proposed System	RF	99%

TABLE 7.2: Performance Comparison of previous studies in accuracy

Chapter 8

CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

Stroke is a serious medical condition that requires immediate treatment to prevent complications. To reduce its occurrence and mortality rate, we must address barriers, raise awareness, and establish effective screening and prevention services for early detection and treatment. This study investigates the effectiveness of various machine learning algorithms in accurately predicting stroke based on physiological variables. It also explores the integration of machine learning models with expert knowledge, which has proven to be beneficial in healthcare.

Furthermore, developing a Knowledge-Based System is crucial for documenting the guidelines, knowledge, and experiences of neurologists. This system can be made accessible to healthcare workers, facilitating knowledge transfer. This study collected and preprocessed data from the Kaggle international website's stroke dataset, resulting in 5110 data points. From the total of 12 available attributes, 7 attributes highly relevant to stroke prediction were selected. Additionally, missing values were handled, categorical data was encoded, embedding is used to reduce the high-dimensionality of encoded data and imbalanced datasets were managed using SMOTE techniques as classification preprocessors.

To find the best prediction model for stroke diagnosis and treatment, the researcher conducted 9 experiments using two scenarios: one with all attributes and another with selected attributes. Three classification algorithms (Decision Tree, Random Forest, and Support Vector Machine) were utilized, employing 10-fold cross-validation and the percentage split test option method.

After evaluating both objective and subjective measures, it was determined that the rules generated by the Random Forest classification algorithm, using selected

attributes and under 10-fold cross-validation, showed superior performance with 99% accuracy. These rules were incorporated into a knowledge base system, integrating expert knowledge.

The developed prototype Knowledge-Based System offers advice to health workers regarding stroke diagnosis and treatment. It employs Rule-Based Reasoning (RBR), utilizing knowledge from both machine learning models and domain experts. This knowledge is represented in the form of rules. The system uses the rules to reason about new situations and make predictions or decisions by following these steps:

Step 1 : The system receives new information or data about a patient.

Step 2 : The system matches the new information to the rules in its knowledge base.

Step 3 : The system applies the rules to the new information to determine the best course of action or prediction whether the patient is Stroked or non-stroked.

Step 4 : Finally, the system generates a response or prediction based on the application of the rules to users with GUI.

The prototype serves as a decision support system for health workers and provides a second opinion for neurologists.

The prototype knowledge-based system and graphical user interface (GUI) were developed using Python 3.10 with the Sublime text editor tool. `pyswip`, a Python library, was added to enable connectivity between Python and Prolog. This bi-directional interface allows the calling of Python objects from Prolog or the calling of Prolog facts and rules from the Python side. The proposed Knowledge-Based System comprises modules for a Knowledge Base, Inference Engines, an Explanation Facility, and a User Interface.

Two evaluation methods were used to assess the performance of the proposed system: system performance testing and user acceptance testing. For system performance testing, 41 test cases were prepared, and a confusion matrix was employed to compare the system's performance against the results of domain experts. The

system achieved a 95% accuracy score. User acceptance testing was conducted using eight evaluation criteria, and trained domain experts used the system to assess how well it met their requirements for stroke diagnosis and treatment. The system received an average user acceptance score of 88%. Consequently, the proposed system demonstrated successful performance in achieving its intended purpose in the absence of neurologists. This success indicates that the Knowledge-Based System possesses the necessary knowledge for effective stroke diagnosis and treatment management.

8.2 FUTURE WORK

The research goals have been achieved, as stated earlier. However, as anticipated, there are still some areas that need improvement and open issues that need to be addressed. Therefore, the next step is to suggest problem areas that have been identified through this research. The researcher suggests the following recommendations for subsequent researchers:

- In this study, a rule-based reasoning technique is used to represent knowledge from different sources. Future researchers can improve and make a more efficient knowledge-based system (KBS) by combining rule-based and case-based reasoning techniques. By doing so, the KBS can benefit from both analyzing conditions and applying previous solutions to new situations.
- The proposed KBS identifies patients as either having a stroke or not, but it doesn't specify the type of stroke. For better diagnosis and treatment, it is recommended to design a knowledge base system that can directly detect each type of stroke.
- The researchers suggested that to apply incremental learning technology in future work. This technology allows for progressive learning and improvement without forgetting previously acquired information. It is particularly useful when data arrives in sequential order or when storing and processing all data is not feasible. The learning process would occur whenever new examples emerge and would adjust what has been learned based on these new examples.

- Furthermore, the researcher recommended that to improve the performance of the machine learning model by adding new features that may capture important information about the Stroke patient and using transfer learning by fine-tuning a pre-trained model on our specific dataset, we the model can adapt to Stroke diagnosis and treatment and improve its performance.
- Lastly, Experiment with different model architectures and hyperparameters. Different types of neural networks model architectures, like convolutional neural networks (CNNs) or recurrent neural networks (RNNs), recommended to see which one performs best on our dataset and Experiment with different hyperparameters, such as learning rate and batch size, recommended to optimize the performance of the model.

Bibliography

- [1] Josephine E Prynne and Hannah Kuper. Perspectives on disability and non-communicable diseases in low-and middle-income countries, with a focus on stroke and dementia. *International journal of environmental research and public health*, 16(18):3488, 2019.
- [2] Christopher JL Murray, Cristiana Abbafati, Kaja M Abbas, Mohammad Abbasi, Mohsen Abbasi-Kangevari, Foad Abd-Allah, Mohammad Abdollahi, Parisa Abedi, Aidin Abedi, Hassan Abolhassani, et al. Five insights from the global burden of disease study 2019. *The Lancet*, 396(10258):1135–1159, 2020.
- [3] Valery L Feigin, Benjamin A Stark, Catherine Owens Johnson, Gregory A Roth, Catherine Bisignano, Gdiom Gebreheat Abady, Mitra Abbasifard, Mohsen Abbasi-Kangevari, Foad Abd-Allah, Vida Abedi, et al. Global, regional, and national burden of stroke and its risk factors, 1990–2019: a systematic analysis for the global burden of disease study 2019. *The Lancet Neurology*, 20(10):795–820, 2021.
- [4] Debraj Mukherjee and Chirag Patil. Epidemiology and the global burden of stroke. *World neurosurgery*, 76:S85–90, 12 2021. doi: 10.1016/j.wneu.2011.07.023.
- [5] Salim S Virani, Alvaro Alonso, Emelia J Benjamin, Marcio S Bittencourt, Clifton W Callaway, April P Carson, Alanna M Chamberlain, Alexander R Chang, Susan Cheng, Francesca N Delling, et al. Heart disease and stroke statistics—2020 update: a report from the american heart association. *Circulation*, 141(9):e139–e596, 2020.
- [6] Connie W Tsao, Aaron W Aday, Zaid I Almarzooq, Cheryl AM Anderson, Pankaj Arora, Christy L Avery, Carissa M Baker-Smith, Andrea Z Beaton,

- Amelia K Boehme, Alfred E Buxton, et al. Heart disease and stroke statistics—2023 update: a report from the american heart association. *Circulation*, 147(8):e93–e621, 2023.
- [7] Biljana Djurić, Katarina Žikić, Zorica Nestorović, Danijela Lepojević-Stefanović, Nebojša Milošević, and Dejan Žikić. Using the photoplethysmography method to monitor age-related changes in the cardiovascular system. *Frontiers in Physiology*, 14, 2023.
- [8] Single Race. *Multiple Cause of Death Data*, 2018-2021.
- [9] C Anthony Burton. Cases in stroke in family medicine. *TAFP Pulse Virtual Conference*, 2023.
- [10] Valery L Feigin, Gregory A Roth, Mohsen Naghavi, Priya Parmar, Rita Krishnamurthi, Sumeet Chugh, George A Mensah, Bo Norrving, Ivy Shiue, Marie Ng, et al. Global burden of stroke and risk factors in 188 countries, during 1990–2013: a systematic analysis for the global burden of disease study 2013. *The Lancet Neurology*, 15(9):913–924, 2016.
- [11] Dragon YL Wong, Mary C Lam, Anran Ran, and Carol Y Cheung. Artificial intelligence in retinal imaging for cardiovascular disease prediction: current trends and future directions. *Current Opinion in Ophthalmology*, 33(5):440–446, 2022.
- [12] Anand Nayyar, Lata Gadhavi, and Noor Zaman. Machine learning in healthcare: review, opportunities and challenges. *Machine Learning and the Internet of Medical Things in Healthcare*, pages 23–45, 2021.
- [13] Alemu Workneh, Dereje Teferi, and Alemu Kumilachew. Knowledge based decision support system for detecting and diagnosis of acute abdomen using hybrid approach. In *Information and Communication Technology for Development for Africa: Second International Conference, ICT4DA 2019, Bahir Dar, Ethiopia, May 28-30, 2019, Revised Selected Papers 2*, pages 57–67. Springer, 2019.

- [14] Saurabh Srivastava and Sachin Dubey. *Knowledge-based systems in medical applications*, pages 189–215. ResearchGet, 01 2020. ISBN 9780128206041. doi: 10.1016/B978-0-12-820604-1.00013-3.
- [15] Eta S Berner and Tonya J La Lande. Overview of clinical decision support systems. *Clinical decision support systems: Theory and practice*, pages 1–17, 2016.
- [16] Li Xu and Ling Li. A hybrid system applied to epidemic screening. *Expert Systems*, 17:81 – 89, 12 2002. doi: 10.1111/1468-0394.00130.
- [17] Mayowa O Owolabi, Amanda G Thrift, Sheila Martins, Walter Johnson, Jeyaraj Pandian, Foad Abd-Allah, Cherian Varghese, Ajay Mahal, Joseph Yaria, Hoang T Phan, et al. The state of stroke services across the globe: Report of world stroke organization–world health organization surveys. *International Journal of Stroke*, 16(8):889–901, 2021.
- [18] Thierry Adoukonou, Oyene Kossi, Pervenche Fotso Mefo, Mendinatou Agbetou, Julien Magne, Glwadys Gbaguidi, Dismand Houinato, Pierre-Marie Preux, and Philippe Lacroix. Stroke case fatality in sub-saharan africa: Systematic review and meta-analysis. *International Journal of Stroke*, 16(8): 902–916, 2021.
- [19] Umar Farooque, Ashok Kumar Lohano, Ashok Kumar, Sundas Karimi, Farah Yasmin, Vijaya Chaitanya Bollampally, and Margil R Ranpariya. Validity of national institutes of health stroke scale for severity of stroke to predict mortality among patients presenting with symptoms of stroke. *Cureus*, 12(9), 2020.
- [20] Lijuan Zhang, Wenwu Sun, Yujun Wang, Xiaopin Wang, Yanli Liu, Su Zhao, Ding Long, Liangkai Chen, and Li Yu. Clinical course and mortality of stroke patients with coronavirus disease 2019 in wuhan, china. *Stroke*, 51(9):2674–2682, 2020.
- [21] Abolfazl Avan, Hadi Digaleh, Mario Di Napoli, Saverio Stranges, Reza Behrouz, Golnaz Shojaeianbabaei, Amin Amiri, Reza Tabrizi, Naghmeh

- Mokhber, J David Spence, and Mahmoud Reza Azarpazhooh. Socioeconomic status and stroke incidence, prevalence, mortality, and worldwide burden: an ecological analysis from the global burden of disease study 2017. *BMC medicine*, 17(1):191, October 2019. ISSN 1741-7015. doi: 10.1186/s12916-019-1397-3. URL <https://europepmc.org/articles/PMC6813111>.
- [22] Bens Pardamean, Robby Christian, Bahtiar Saleh Abbas, et al. Expert-system based medical stroke prevention. *Journal of Computer Science*, 9(9): 1099, 2013.
- [23] Samuel M. Integrating datamining result with knowledge based system for diagnosis and treatment recommendation of stroke: The case of minilik ii referral and korean hospitals. for the partioial fulfillment of grauation of master of since in informatin system, 2020.
- [24] Baiba Vilne, Juris Ķibilds, Inese Sikсна, Ilva Lazda, Olga Valciņa, and Angelika Krūmiņa. Could artificial intelligence/machine learning and inclusion of diet-gut microbiome interactions improve disease risk prediction? case study: coronary artery disease. *Frontiers in Microbiology*, 13:627892, 2022.
- [25] Shraddha Mainali, Marin E Darsie, and Keaton S Smetana. Machine learning in action: stroke diagnosis and outcome prediction. *Frontiers in Neurology*, 12:734345, 2021.
- [26] Anirudha S Chandrabhatla, Elyse A Kuo, Jennifer D Sokolowski, Ryan T Kellogg, Min Park, and Panagiotis Mastorakos. Artificial intelligence and machine learning in the diagnosis and management of stroke: A narrative review of united states food and drug administration-approved technologies. *Journal of Clinical Medicine*, 12(11):3755, 2023.
- [27] Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.
- [28] Philipp Offermann, Olga Levina, Marten Schönherr, and Udo Bub. Outline of a design science research process. In *Proceedings of the 4th International*

- Conference on Design Science Research in Information Systems and Technology*, pages 1–11, 2009.
- [29] Priti Srinivas Sajja and Rajendra Akerkar. Knowledge-based systems for development. *Advanced Knowledge Based Systems: Model, Applications & Research*, 1:1–11, 2010.
- [30] Stephen JX Murphy and David J Werring. Stroke: causes and clinical features. *Medicine*, 48(9):561–566, 2020.
- [31] Tahia Tazin, Md Nur Alam, Nahian Nakiba Dola, Mohammad Sajibul Bari, Sami Bourouis, Mohammad Monirujjaman Khan, et al. Stroke disease detection and prediction using robust learning approaches. *Journal of healthcare engineering*, 2021, 2021.
- [32] Shelagh B Coutts. Diagnosis and management of transient ischemic attack. *CONTINUUM: Lifelong Learning in Neurology*, 23(1):82, 2017.
- [33] Afolabi Abiodun. Stroke (cerebrovascular accident (cva) or brain attack) and its management-literature review. *Int J Innov Healthc Res*, 6:1–9, 2018.
- [34] Diji Kuriakose and Zhicheng Xiao. Pathophysiology and treatment of stroke: present status and future perspectives. *International journal of molecular sciences*, 21(20):7609, 2020.
- [35] Rajendra Akerkar and Priti Sajja. *Knowledge-based systems*. Jones & Bartlett Publishers, 2009.
- [36] Kimiz Dalkir. *Knowledge management in theory and practice*. routledge, 2013.
- [37] Veronique Ambrosini and Cliff Bowman. Tacit knowledge: Some suggestions for operationalization. *Journal of Management studies*, 38(6):811–829, 2001.
- [38] Mark Hoksbergen, Johnny Chan, Gabrielle Peko, and David Sundaram. Illuminating and bridging the vortex between tacit and explicit knowledge: Counterbalancing information asymmetry in high-value low-frequency transactions. *Decision Support Systems*, 149:113605, 2021.

- [39] Don Chathurika Amarathunga, John Grundy, Hazel Parry, and Alan Dorin. Methods of insect image capture and classification: A systematic literature review. *Smart Agricultural Technology*, 1:100023, 2021.
- [40] Chee-Fai Tan. A prototype of knowledge-based system for fault diagnosis in automatic wire bonding machine. *Turkish Journal of Engineering and Environmental Sciences*, 32(4):235–244, 2008.
- [41] KP Tripathi. A review on knowledge-based expert system: concept and architecture. *IJCA Special Issue on Artificial Intelligence Techniques-Novel Approaches & Practical Applications*, 4:19–23, 2011.
- [42] M Mayilvaganan, R Deepa, and S Malathi. Implementation of inference engine in adaptive neuro fuzzy inference system to predict and control the sugar level in diabetic patient. *Pakistan Journal of Biotechnology*, 14 (Special II):102–105, 2017.
- [43] Berhanu Aebissa. *developing A Knowledge based system for coffee disease diagnosis and treatment*. PhD thesis, Addis Ababa University, 2012.
- [44] Renata M Saraiva, João Bezerra, Mirko Perkusich, Hyggo Almeida, and Claurton Siebra. A hybrid approach using case-based reasoning and rule-based reasoning to support cancer diagnosis: a pilot study. In *MEDINFO 2015: eHealth-enabled Health*, pages 862–866. IOS Press, 2015.
- [45] M Sasikumar S Ramani. A practical introduction to rule based expert systems, 2007.
- [46] Boban Vesin, Mirjana Ivanović, Aleksandra Klačnja-Milićević, and Zoran Budimac. Rule-based reasoning for building learner model in programming tutoring system. In *Advances in Web-Based Learning-ICWL 2011: 10th International Conference, Hong Kong, China, December 8-10, 2011. Proceedings 10*, pages 154–163. Springer, 2011.
- [47] Petr Berka. Sentiment analysis using rule-based and case-based reasoning. *Journal of Intelligent Information Systems*, 55(1):51–66, 2020.

- [48] Amir Pourabdollah, Jerry M Mendel, and Robert I John. Alpha-cut representation used for defuzzification in rule-based systems. *Fuzzy Sets and Systems*, 399:110–132, 2020.
- [49] RA Burnashev, IA Enikeev, and AI Enikeev. Design and implementation of integrated development environment for building rule-based expert systems. In *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, pages 1–4. IEEE, 2020.
- [50] Amirhossein Eslami Andargoli, Helana Scheepers, Diana Rajendran, and Amrik Sohal. Health information systems evaluation frameworks: A systematic review. *International journal of medical informatics*, 97:195–209, 2017.
- [51] J Matthew Helm, Andrew M Swiergosz, Heather S Haeberle, Jaret M Karnuta, Jonathan L Schaffer, Viktor E Krebs, Andrew I Spitzer, and Prem N Ramkumar. Machine learning and artificial intelligence: definitions, applications, and future directions. *Current reviews in musculoskeletal medicine*, 13:69–76, 2020.
- [52] Li Li, Nan-Ning Zheng, and Fei-Yue Wang. On the crossroad of artificial intelligence: A revisit to alan turing and norbert wiener. *IEEE transactions on cybernetics*, 49(10):3618–3626, 2018.
- [53] Ylenia Casali, Nazli Yonca Aydin, and Tina Comes. Machine learning for spatial analyses in urban areas: a scoping review. *Sustainable Cities and Society*, 85:104050, 2022.
- [54] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.
- [55] Salim Dridi. Supervised learning-a systematic literature review. *Machine learning*, 2021.
- [56] Yining Dong and S Joe Qin. Regression on dynamic pls structures for supervised learning of dynamic data. *Journal of process control*, 68:64–72, 2018.

- [57] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. Unsupervised learning. In *An Introduction to Statistical Learning: with Applications in Python*, pages 503–556. Springer, 2023.
- [58] Zhi-Hua Zhou and Zhi-Hua Zhou. Semi-supervised learning. *Machine Learning*, pages 315–341, 2021.
- [59] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, Joelle Pineau, et al. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354, 2018.
- [60] Ethem Alpaydm. *Introduction to Machine Learning, (Adaptive Computation and Machine Learning)*. almohrerladbi, 2014.
- [61] Amol A Verma, Joshua Murray, Russell Greiner, Joseph Paul Cohen, Kaveh G Shojania, Marzyeh Ghassemi, Sharon E Straus, Chloe Pou-Prom, and Muhammad Mamdani. Implementing machine learning in medicine. *Cmaj*, 193(34):E1351–E1357, 2021.
- [62] Yogesh Kumar, Komalpreet Kaur, and Gurpreet Singh. Machine learning aspects and its applications towards different research areas. In *2020 International conference on computation, automation and knowledge management (ICCAKM)*, pages 150–156. IEEE, 2020.
- [63] Emily Sullivan. Understanding from machine learning models. *The British Journal for the Philosophy of Science*, 2022.
- [64] Pratap Chandra Sen, Mahimarnab Hajra, and Mitadru Ghosh. Supervised classification algorithms in machine learning: A survey and review. In *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018*, pages 99–111. Springer, 2020.
- [65] Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification. In *Conference on Fairness, accountability and transparency*, pages 107–118. PMLR, 2018.

-
- [66] Murat Koklu and Ilker Ali Ozkan. Multiclass classification of dry beans using computer vision and machine learning techniques. *Computers and Electronics in Agriculture*, 174:105507, 2020.
- [67] Yongjian Zhong, Bo Du, and Chang Xu. Learning to reweight examples in multi-label classification. *Neural Networks*, 142:428–436, 2021.
- [68] Bahzad Charbuty and Adnan Abdulazeez. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01):20–28, 2021.
- [69] Malti Bansal, Apoorva Goyal, and Apoorva Choudhary. A comparative analysis of k-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning. *Decision Analytics Journal*, 3:100071, 2022.
- [70] Weiwei Lin, Ziming Wu, Longxin Lin, Angzhan Wen, and Jin Li. An ensemble random forest algorithm for insurance big data analysis. *Ieee access*, 5:16568–16575, 2017.
- [71] Matthias Schonlau and Rosie Yuyan Zou. The random forest algorithm for statistical learning. *The Stata Journal*, 20(1):3–29, 2020.
- [72] Alaa Tharwat. Parameter investigation of support vector machine classifier with kernel functions. *Knowledge and Information Systems*, 61:1269–1302, 2019.
- [73] Lucas Bernardi, Themistoklis Mavridis, and Pablo Estevez. 150 successful machine learning models: 6 lessons learned at booking. com. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1743–1751, 2019.
- [74] Gwanhoo Lee and Weidong Xia. Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *MIS quarterly*, 34(1):87–114, 2010.

- [75] Songhee Cheon, Jungyoon Kim, and Jihye Lim. The use of deep learning to predict stroke patient mortality. *International journal of environmental research and public health*, 16(11):1876, 2019.
- [76] Shujun Zhang, Shuhao Xu, Liwei Tan, Hongyan Wang, and Jianli Meng. Stroke lesion detection and analysis in mri images based on deep learning. *Journal of Healthcare Engineering*, 2021:1–9, 2021.
- [77] Gangavarapu Sailasya and Gorli L Aruna Kumari. Analyzing the performance of stroke prediction using ml classification algorithms. *International Journal of Advanced Computer Science and Applications*, 12(6), 2021.
- [78] Sotarat Thammaboosadee and Teerapat Kansadub. Data mining model and application for stroke prediction: A combination of demographic and medical screening data approach. *Interdisciplinary Research Review*, 14(4): 61–69, 2019.
- [79] Ohoud Almadani and Riyad Alshammari. Prediction of stroke using data mining classification techniques. *International Journal of Advanced Computer Science and Applications*, 9(1), 2018.
- [80] José Alberto Tavares Rodríguez. Stroke prediction through data science and machine learning algorithms. *Applied Sciences*, 2021.
- [81] Matthew Chun, Robert Clarke, Benjamin J Cairns, David Clifton, Derrick Bennett, Yiping Chen, Yu Guo, Pei Pei, Jun Lv, Canqing Yu, et al. Stroke risk prediction using machine learning: a prospective cohort study of 0.5 million chinese adults. *Journal of the American Medical Informatics Association*, 28(8):1719–1727, 2021.
- [82] Teresa Podsiadly-Marczykowska, Bogdan Ciszek, and Artur Przelaskowski. Development of diagnostic stroke ontology-preliminary results. In *Information Technologies in Biomedicine, Volume 4*, pages 261–272. Springer, 2014.

- [83] Fahim T Imam, Stephen D Larson, Anita Bandrowski, Jeffery S Grethe, Amarnath Gupta, and Maryann E Martone. Development and use of ontologies inside the neuroscience information framework: a practical approach. *Frontiers in genetics*, 3:111, 2012.
- [84] Franziska Schorr and Lars Hvam. Design science research: A suitable approach to scope and research it service catalogs. In *2018 IEEE World Congress on Services (SERVICES)*, pages 25–26. IEEE, 2018.
- [85] YungYu Lin, Yukari Nagai, TzuHang Chiang, and HuaKo Chiang. Design and develop artifact for integrating with erp and ecs based on design science. In *Proceedings of the 3rd International Conference on Information Science and Systems*, pages 218–223, 2020.
- [86] Neetu Aravind. Aligning data architecture and data governance. Master’s thesis, University of Twente, 2021.
- [87] AR Hevner, ST March, J Park, and S Ram. Design science in information systems research mis quarterly vol. 28 no. 1, 2004.
- [88] Alta Van der Merwe, AURONA Gerber, and Hanlie Smuts. Guidelines for conducting design science research in information systems. In *Annual Conference of the Southern African Computer Lecturers’ Association*, pages 163–178. Springer, 2019.
- [89] Nazar Zaki, Elfadil A Mohamed, and Tetiana Habuza. From tabulated data to knowledge graph: A novel way of improving the performance of the classification models in the healthcare data. *medRxiv*, pages 2021–06, 2021.
- [90] Gheorghe Tecuci, Yves Kodratoff, John Boose, and Brian Gaines. *Machine learning and knowledge acquisition: Integrated approaches*. Academic Press Ltd., 1995.
- [91] Luis Daza and Edgar Acuna. An algorithm for detecting noise on supervised classification. In *Proceedings of WCECS-07, the 1st world conference on engineering and computer science*, pages 701–706, 2007.

- [92] Biredagn Kindie. *Developing an Intelligent System to Identify Suitable Crop Types Using Machine Learning Approach*. PhD thesis, computing, 2021.
- [93] José A Sáez, Julián Luengo, Jerzy Stefanowski, and Francisco Herrera. Smote-tpf: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, 291:184–203, 2015.
- [94] Nagesh Singh Chauhan. Decision tree algorithm. *Explained—KDnuggets*. Available online: <https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html> (accessed on 1 November 2021), 2020.
- [95] Cristiano Mauro Assis Gomes, Gina C Lemos, and Enio G Jelihovschi. Comparing the predictive power of the cart and ctrees algorithms. *Avaliação Psicológica*, 19(1):87–96, 2020.
- [96] Jay Doshi, Kunal Parmar, Raj Sanghavi, and Narendra Shekokar. A comprehensive dual-layer architecture for phishing and spam email detection. *Computers & Security*, 133:103378, 2023.
- [97] Vinod Kumar Chauhan, Kalpana Dahiya, and Anuj Sharma. Problem formulations and solvers in linear svm: a review. *Artificial Intelligence Review*, 52(2):803–855, 2019.
- [98] Tianhua Chen, Changjing Shang, Pan Su, Elpida Keravnou-Papailiou, Yitian Zhao, Grigoris Antoniou, and Qiang Shen. A decision tree-initialised neuro-fuzzy approach for clinical decision support. *Artificial Intelligence in Medicine*, 111:101986, 2021.
- [99] Akshansh Sharma, Firoj Khan, Deepak Sharma, Sunil Gupta, and FY Student. Python: the programming language of future. *Int. J. Innovative Res. Technol*, 6(2):115–118, 2020.
- [100] Kevin M Mendez, Leighton Pritchard, Stacey N Reinke, and David I Broadhurst. Toward collaborative open data science in metabolomics using jupyter notebooks and cloud computing. *Metabolomics*, 15:1–16, 2019.

- [101] Jan M Binder, Alexander Stark, Nikolas Tomek, Jochen Scheuer, Florian Frank, Kay D Jahnke, Christoph Müller, Simon Schmitt, Mathias H Metsch, Thomas Uden, et al. Qudi: A modular python suite for experiment control and data processing. *SoftwareX*, 6:85–90, 2017.
- [102] Jiawei Wang, Li Li, and Andreas Zeller. Better code, better sharing: on the need of analyzing jupyter notebooks. In *Proceedings of the ACM/IEEE 42nd international conference on software engineering: new ideas and emerging results*, pages 53–56, 2020.
- [103] N Juristo and José L Morant. Common framework for the evaluation process of kbs and conventional software. *Knowledge-Based Systems*, 11(2):145–159, 1998.
- [104] Alaa Tharwat. Classification assessment methods. *Applied computing and informatics*, 17(1):168–192, 2020.
- [105] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. Big data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, 30(4):431–448, 2018.
- [106] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Hippocratic databases. In *VLDB’02: Proceedings of the 28th International Conference on Very Large Databases*, pages 143–154. Elsevier, 2002.
- [107] Stefan Studer, Thanh Binh Bui, Christian Drescher, Alexander Hanuschkin, Ludwig Winkler, Steven Peters, and Klaus-Robert Müller. Towards crisp-ml (q): a machine learning process model with quality assurance methodology. *Machine learning and knowledge extraction*, 3(2):392–413, 2021.
- [108] Rana Muhammad Adnan Ikram, Barenya Bikash Hazarika, Deepak Gupta, Salim Heddami, and Ozgur Kisi. Streamflow prediction in mountainous region using new machine learning and data preprocessing methods: a case study. *Neural Computing and Applications*, 35(12):9053–9070, 2023.

- [109] Khishigsuren Davagdorj, Jong Seol Lee, Van Huy Pham, and Keun Ho Ryu. A comparative analysis of machine learning methods for class imbalance in a smoking cessation intervention. *Applied Sciences*, 10(9):3307, 2020.
- [110] Taher Al-Shehari and Rakan A Alsowail. An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques. *Entropy*, 23(10):1258, 2021.
- [111] Danijela Protić and Miomir Stanković. Anomaly-based intrusion detection: Feature selection and normalization influence to the machine learning models accuracy. *European Journal of Formal Sciences and Engineering*, 3(1):1–9, 2020.
- [112] Thippa Reddy Gadekallu, Neelu Khare, Sweta Bhattacharya, Saurabh Singh, Praveen Kumar Reddy Maddikunta, and Gautam Srivastava. Deep neural networks to predict diabetic retinopathy. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–14, 2020.
- [113] Arthur K Kordon and Arthur K Kordon. Problem knowledge acquisition. *Applying Data Science: How to Create Value with Artificial Intelligence*, pages 203–219, 2020.
- [114] Annie McCluskey, Angela Vratsistas-Curto, and Karl Schurr. Barriers and enablers to implementing multiple stroke guideline recommendations: a qualitative study. *BMC health services research*, 13:1–13, 2013.
- [115] Steve R Ommen, Seema Mital, Michael A Burke, Sharlene M Day, Anita Deswal, Perry Elliott, Lauren L Evanovich, Judy Hung, José A Joglar, Paul Kantor, et al. 2020 aha/acc guideline for the diagnosis and treatment of patients with hypertrophic cardiomyopathy: a report of the american college of cardiology/american heart association joint committee on clinical practice guidelines. *Journal of the American College of Cardiology*, 76(25):e159–e240, 2020.
- [116] Krishna Mridha, Sandesh Ghimire, Jungpil Shin, Anmol Aran, Md. Mezbah Uddin, and M. F. Mridha. Automated stroke prediction using machine learning: An explainable and exploratory study with a web application for early

- intervention. *IEEE Access*, 11:52288–52308, 2023. doi: 10.1109/ACCESS.2023.3278273.
- [117] Ivan G Ivanov, Yordan Kumchev, and Vincent James Hooper. An optimization precise model of stroke data to improve stroke prediction. *Algorithms*, 16(9):417, 2023.
- [118] Johan Zelano, Martin Holtkamp, Nivedita Agarwal, Simona Lattanzi, Eugen Trinka, and Francesco Brigo. How to diagnose and treat post-stroke seizures and epilepsy. *Epileptic Disorders*, 22(3):252–263, 2020.
- [119] Weinan Yang, Lincheng Zhang, Qigu Yao, Weiyan Chen, Weiji Yang, Suqing Zhang, Lan He, Hong Li, and Yuyan Zhang. Endovascular treatment or general treatment: how should acute ischemic stroke patients choose to benefit from them the most?: A systematic review and meta-analysis. *Medicine*, 99(20), 2020.
- [120] Samy S Abu-Naser and Bashar G Bastami. A proposed rule based system for breasts cancer diagnosis. *International Journal of General Systems*, 2016.
- [121] Michael Frank, Dimitris Drikakis, and Vassilis Charissis. Machine-learning methods for computational science and engineering. *Computation*, 8(1):15, 2020.
- [122] Inés Sittón Candanedo, Elena Hernández Nieves, Sara Rodríguez González, M Teresa Santos Martín, and Alfonso González Briones. Machine learning predictive model for industry 4.0. In *Knowledge Management in Organizations: 13th International Conference, KMO 2018, Žilina, Slovakia, August 6–10, 2018, Proceedings 13*, pages 501–510. Springer, 2018.
- [123] Jayakiran Reddy Esanakula, CNV Sridhar, and V Pandu Rangadu. Development of kbs for cad modeling of industrial battery stack and its configuration: An approach. In *Intelligent Systems Technologies and Applications 2016*, pages 607–618. Springer, 2016.
- [124] Kevin R Murphy. Performance evaluation will not die, but it should. *Human Resource Management Journal*, 30(1):13–31, 2020.

Appendix I

Domain Expert Interview Questions

DEBRE BERHAN UNIVERSITY
COLLEGE OF COMPUTING
DEPARTMENT OF INFORMATION
TECHNOLOGY

Dear Interviewees.

First of all, the researcher would like to thank you for your willingness to make yourself available for the interview. The purpose of this interview is to acquire knowledge about stroke. The knowledge that are going to be collected from you will be used to develop the Knowledge Based System that gives an advice about diagnosis and Treatment of stroke to medical doctor, health workers that do not have deep knowledge about stroke diagnosis and treatment. Your feedback's are very important for the success of the research, which is conducted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology, because the power and usability of the Knowledge Based System heavily relies on its knowledge base.

1. What is stroke ?
2. What are the general symptoms of stroke ?
3. How to diagnosis stroke ?
4. How many types of stroke are their ?
5. What are the main risk factors to stroke ?

6. What are the treatment mechanisms for each conforming stroke ?

7. Any more points you want to add about stroke ?

Appendix II

User Acceptance Testing

DEBRE BERHAN UNIVERSITY
COLLEGE OF COMPUTING
DEPARTMENT OF INFORMATION
TECHNOLOGY

Dear Evaluator This evaluation form is prepared aiming at measuring to what extent does stroke diagnosis and treatment KBS is usable and acceptable by end users in the area of health sectors. Therefore, you are kindly requested to evaluate the system by labeling (✓) symbol on the space provided for the corresponding question. The values for all questions in the table are rated as: Excellent = 5, Very good = 4, Good = 3, Fair = 2 and Poor = 1. I would like to appreciate your collaboration in providing the information.

No	Criteria's for evaluating the proposed KBS	Poor	Fair	Good	V. Good	Excellent	Average
1	Simplicity of KBS						
2	Does the proposed system contain the required symptoms to stroke diagnosis and treatment ?						
3	Efficiency of the proposed system relative to its response time ?						
4	Does the proposed system recommend appropriate diagnosis and treatment for stroke?						
5	Is the user interface suitable for the user?						
6	Would you like to use the KBS frequently?						
7	Is the system effective in diagnosis and treatment of stroke?						
8	How do you rate the importance of the proposed system?						
Total Average							

Appendix III

Sample codes

.1 Information about kaggle stroke prediction dataset

```
1 # Reading Dataset:
2 data = pd.read_csv("healthcare-dataset-stroke-data.csv")
3 data.head(10)
```

```
Out[37]:
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1
5	56669	Male	81.0	0	0	Yes	Private	Urban	186.21	29.0	formerly smoked	1
6	53882	Male	74.0	1	1	Yes	Private	Rural	70.09	27.4	never smoked	1
7	10434	Female	69.0	0	0	No	Private	Urban	94.39	22.8	never smoked	1
8	27419	Female	59.0	0	0	Yes	Private	Rural	76.15	NaN	Unknown	1
9	60491	Female	78.0	0	0	Yes	Private	Urban	58.57	24.2	Unknown	1

.2 Sample Prolog code with the syntax of python

```
1 import time
2 import json
3
4 def prolog_query(query_string):
5     prolog = Prolog()
6     prolog.consult("123.pl")
7     results = []
8     for res in prolog.query(query_string):
9         results.append(res)
10    return results
11 def ask_question(query_string):
12    answers = prolog_query(query_string)
13    return answers
14 def make_json(data):
15    json_str = ""
```

```
16     for c in data:
17         if c == "":
18             json_str += ""
19             continue
20         json_str += c
21     return json_str
22
23 example_kb = pl.KnowledgeBase("stroke")
24 example_kb.from_file("123.pl")
25 example_kb.query(pl.Expr("type21792(speach(C),dizzy(K),pregnanttime(O))"))
26 example_kb.query(pl.Expr("type26(speach(C),dizzy(K),pregnanttime(O),renaldisese(N),
27     visualcomplaints(P),face(A))"))
28 example_kb.query(pl.Expr("type216(speach(C),dizzy(K),pregnanttime(O),vision(D),
29     paralysis(I))"))
30 example_kb.query(pl.Expr("type270(speach(C),dizzy(K),pregnanttime(O),vision(D),
31     eomtion(J))"))
32 example_kb.query(pl.Expr("type240(speach(C),dizzy(K),pregnanttime(O),vision(D),
33     emotion(J),face(A))"))
34 example_kb.query(pl.Expr("type210(speach(C),dizzy(K),pregnanttime(O),vision(D),
35     emotion(J),face(A),headache(F))"))
36 example_kb.query(pl.Expr("type26220(speach(C),dizzy(K),swallowing(L),vision(D))"))
37 example_kb.query(pl.Expr("type160(speach(C),dizzy(K),swallowing(L),vision(D),
38     face(A),balance(E))"))
39 example_kb.query(pl.Expr("type110(speach(C),dizzy(K),swallowing(L),vision(D),
40     face(A),balance(E))"))
41 example_kb.query(pl.Expr("gdm2(speach(C),dizzy(K),pregnanttime(O),renaldisese(N),
42     visualcomplaints(P),face(A),arm(B))"))
43 example_kb.query(pl.Expr("gdm77(speach(C),dizzy(K),pregnanttime(O),renaldisese(N),
44     arm(B))"))
45 example_kb.query(pl.Expr("gdm20(speach(C),dizzy(K),pregnanttime(O),face(A),arm(B))"))
46 example_kb.query(pl.Expr("gdm27(speach(C),dizzy(K),pregnanttime(O),arm(B))"))
47 example_kb.query(pl.Expr("prestroke329(speach(C),pregnanttime(O))"))
48 example_kb.query(pl.Expr("strokefree123(speach(C),pregnanttime(O),emotion(J))"))
49 example_kb.query(pl.Expr("strokefree2(speach(C),pregnanttime(O),emotion(J),
50     vision(D),bladder(G))"))
51 example_kb.query(pl.Expr("strokefree3(speach(C),pregnanttime(O),emotion(J),
52     vision(D))"))
53
54 cc1=list(prolog.query("rule1(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
```

```
55 cc2=list(prolog.query("rule2(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
56 cc3=list(prolog.query("rule3(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
57 cc4=list(prolog.query("rule4(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
58 cc5=list(prolog.query("rule5(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
59 cc6=list(prolog.query("rule6(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
60 cc7=list(prolog.query("rule7(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
61 cc8=list(prolog.query("rule8(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
62 cc9=list(prolog.query("rule9(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
63 cc10=list(prolog.query("rule10(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
64 cc11=list(prolog.query("rule11(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
65 cc12=list(prolog.query("rule12(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
66 cc13=list(prolog.query("rule13(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
67 cc14=list(prolog.query("rule14(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
68 cc15=list(prolog.query("rule15(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
69 cc16=list(prolog.query("rule16(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P)"))
70
71 print("r1 :",cc1)
72 print("r2 :", (cc2))
73 print("r3 :", (cc3))
74 print("r4 :", cc4)
75 print("r5 :", (cc5))
76 print("r6 :", (cc6))
77 print("r7 :", cc7)
78 print("r8 :", (cc8))
79 print("r9 :", (cc9))
80 print("r10 :", cc10)
81 print("r11 :", cc11)
82 print("r12 :", (cc12))
83 print("r13 :", (cc13))
84 print("r14 :", cc14)
85 print("r15 :", (cc15))
86 print("r16 :", (cc16))
```

.3 Sample HTML code for GUI

```
1 <!DOCTYPE html>
2 <!-- Template by Quackit.com -->
3 <html lang="en">
```

```
4 <head>
5   <meta charset="utf-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <title>stroke</title>
9   <!-- Bootstrap Core CSS -->
10  <link href="css/bootstrap.min.css" rel="stylesheet">
11  <link href="css/custom.css" rel="stylesheet">
12 </head>
13 <body>
14 <!-- Navigation -->
15   <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
16     <div class="container">
17       <!-- Logo and responsive toggle -->
18       <div class="navbar-header">
19         <button type="button" class="navbar-toggle"
20           data-toggle="collapse" data-target="#navbar">
21           <span class="sr-only">Toggle navigation</span>
22           <span class="icon-bar"></span>
23           <span class="icon-bar"></span>
24           <span class="icon-bar"></span>
25         </button>
26         <!-- <a class="navbar-brand" href="#">
27         <span class="glyphicon glyphicon-globe"></span> Logo</a> -->
28       </div>
29     <div class="collapse navbar-collapse" id="navbar">
30       <ul class="nav navbar-nav">
31         <li class="active">
32           <a href="#">Home</a>
33         </li>
34         <li>
35           <a href="checkup.html" >check up</a>
36         </li>
37         <li>
38           <a href="#">About</a>
39         </li>
40         <li>
41           <a href="#">Products</a>
42         </li>
```

```
43         <li class="dropdown">
44             <a href="#" class="dropdown-toggle"
45                 data-toggle="dropdown" role="button"
46                 aria-haspopup="true" aria-expanded="false">
47                 Services<span class="caret"></span></a>
48                 <ul class="dropdown-menu" aria-
49                     labelledby="about-us">
50                     <li><a href="#">Engage</a></li>
51                     <li><a href="#">Pontificate</a></li>
52                     <li><a href="#">Synergize</a></li>
53                     </ul>
54                 </li>
55             </ul>
56             <!-- Log In Form -->
57             <form class="navbar-form navbar-right form-inline">
58                 <div class="form-group">
59                     <label class="sr-only"
60                         for="emailAddress">Email address</label>
61                     <input type="email" class="form-control"
62                         id="emailAddress" placeholder="Email">
63                     </div>
64                 <div class="form-group">
65                     <label class="sr-only" for="pwd">Password</label>
66                     <input type="password" class="form-control"
67                         id="pwd" placeholder="Password">
68                 </div>
69                 <button type="submit" class="btn btn-default">
70                     Sign in</button>
71             </form>
72         </div>
73     <!-- /.navbar-collapse -->
74 </div>
75 <!-- /.container -->
76 </nav>
77 </div>
78 </body>
79 </html>
```