Digital Twin Disease Diagnosis Using Machine Learning

by

Rahatara Ferdousi

A thesis
submitted to the University of Ottawa
in partial fulfillment of the
thesis requirement for the degree of
Master of Computer Science

Declaration of Authorship

I hereby certify that this thesis is entirely my own original work except where otherwise indicated. I am aware of the University's regulations concerning plagiarism, including those concerning consequent disciplinary actions. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Abstract

COVID-19 has led to a surge in the adoption of digital transformation in almost every sector. Digital health and well-being are no exception. For instance, now people get checkups via apps or websites instead of visiting a physician. The pandemic has pushed the healthcare sector worldwide to advance the adoption of artificial intelligence (AI) capabilities.

Considering the demand for AI in supporting the well-being of an individual, we present the real-life diagnosis as a digital twin(DT) diagnosis using machine learning. The Machine Learning (ML) technology enables DT to offer a prediction. Although several attempts exist for predicting disease using ML and a few attempts through ML of DT frameworks, those do not deal with disease risk prediction. In addition, most of them deal with single disease prediction after the occurrence and rely only on clinical test data like- ECG report, MRI scan, etc.

To predict multiple disease/disease risks, we propose a dynamic machine learning algorithm (MLA) selection framework and a dynamic testing method. The proposed framework accepts heterogeneous electronic health records (EHRs) or digital health status as datasets and selects suitable MLA upon the highest similarity. Then it trains specific classifiers for predicting a specific disease/disease risk. The dynamic testing method for prediction is used for predicting several diseases.

We described three use cases: non-communicable disease(NCD) risk prediction, mental well-being prediction, and COVID-19 prediction. We selected diabetes, risk of diabetes, liver disease, thyroid, risk of stroke as NCDs, mental stress as a mental health issue, and COVID-19. We employed seven datasets, including public and private datasets, with a diverse range of attributes, sizes, types, and formats to evaluate whether the proposed framework is suitable to data heterogeneity. Our experiment found that the proposed

methods of dynamic MLA selection could select MLA for each dataset at cosine similarity scores ranging between 0.82-0.89. In addition, we predicted target disease/disease risks at an accuracy ranging from 94.5% to 98%.

To verify the performance of the framework-selected predictor, we compared the accuracy measures individually for each of the three cases. We compared them with traditional ML disease prediction work in the literature. We found that the framework-selected algorithms performed with good accuracy compared to existing literature.

Acknowledgements

At first, I am grateful to the Almighty. I want to extend my heartfelt gratitude and appreciation to the people who helped us bring this study into reality with boundless love and appreciation. My cordial thanks to Professor Abdulmotaleb El Saddik for supervising this work. His consistent guidance and advice helped me to accomplish this research. Also, I express my heartiest gratitude to Dr. Anwar Hossain Dr. Fedwa Laamarti for their continuous motivation and suggestions to conduct this thesis. Furthermore, I express my warm gratitude to my beloved MCRLab colleagues. Lastly, I want to thank my dear husband, Khan Tanvir Hossain, for his patience and love during this research.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter presents the background, research problem, application scenario, research objective and statement, thesis contribution, organization, and scholarly outputs.

1.1 Background

Digital Twin (DT) has gained success in the production industry, and it is now getting attention for health and well-being. In the context of health and well being a digital twin is defined as follows.

Digital Twins enable the monitoring, understanding, and optimization of all functioning of humans, and provide constant health insight to improve quality of life and well-being [31].

The key benefit of the DT system is that DT utilizes Artificial Intelligence (AI) to remove the barriers of interoperability between heterogeneous data [78]. In addition, the Machine Learning(ML) part of the AI enables prediction or decision making from heterogeneous digital data [30]. Therefore, DT can be advantageous to preventive, cost-effective,

and guided healthcare. Furthermore, it can benefit early identification of health issues before they developed.

Recently, DT has been studied rapidly for predictive healthcare in the context of health and well-being. Some notable of those are elderly health monitoring [54], hospital management [49], cardiovascular monitoring [24], heart condition classification [32], fitness management [19], medicine prediction [24], etc. A few works are available for clinical decision support system (CDSS) [69]. However, DT connecting heterogeneous data sources and involving various machine learning algorithms (MLA) could support CDSS through multiple disease prediction at an early stage.

There are four major types of DT frameworks including 3-Dimensional [56], Cloud CPS based DT [12], Intelligent DT [30], and Industry 4.0 DT [11]. The Intelligent DT mainly includes an AI inference module that particularly employs ML and Data Mining(DM) technologies to act as the brain of the DT. As the intelligent frameworks focus more on ML and DM, it can benefit a wide range of applications for health and well-being.

More details of Digital Twin and its evolution in health and well-being are elaborated in Chapter 2. In the next section, we discuss the research problems identified in this thesis.

1.2 Research Problem

Digital Twin has been embraced to support digital well-being due to its capability of performing prediction on a vast range of data [34]. The existing research has shown that AI through ML can address several well-being problems like heart disease prediction, hospital monitoring, elderly health monitoring, etc. However, most of these works have aimed at predicting disease or health issues after the occurrence of the disease. They are mainly

designed to handle only a specific disease, such as heart problem prediction, cancer prediction, diabetes prediction, etc. Also, these works depend on a single type of data source for the prediction, for instance, ECG record for heart disease prediction.

The sign, symptom risk factors represent epidemiological factors of disease. Interestingly, an individual's behavioural, biological, clinical or environmental data can represent signs, symptoms, risk factors of various disease risks and diseases [16]. Therefore, heterogeneous data representing epidemiological disease factors for an individual could be utilized by the ML part of a digital twin. In that case, a DT of health could support early-stage disease prediction virtually. More precisely, DT could aid digital diagnosis of disease, which is beneficial to improve an individual's well-being and reduce the economic burden of late treatment.

In the next section, we discuss some application scenarios to understand the requirements of a well-being DT system.

1.3 Application Scenario

Let us consider a person who feels some health issues and wants to have a health checkup. For this purpose, she tries to book an appointment with a doctor and finds no availability before next month. If a disease like Type-1 diabetes develops during this time, the patient may suffer from a long-term loss.

Another application scenario is that a person wants to maintain a healthy lifestyle to prevent non-communicable diseases, e.g., diabetes, stroke, cardiovascular disease, etc. Regular monitoring of the patient's health may fulfill this objective. However, it becomes expensive in terms of time and cost to have a regular specialist checkup.

The patients may find some health checkup apps and websites online. She would see that there are different apps for each disease, for example, apps for heart disease prediction, liver disease prediction, stress prediction, etc. It makes the patient confused and frustrated again.

The above scenarios motivated us to work on the idea of a digital twin disease diagnosis to predict multiple diseases from various digital data representing a patient's health status. The design of such a framework would have the following requirements:

- First, the framework shall predict multiple diseases by utilizing multiple data sources and suitable machine learning algorithms.
- Second, the framework shall consider various health information sources such as Electronic Health Record (EHR) data, sensor data, and clinical data.

1.4 Research Objective and Statement

The key objective of our thesis is to propose a dynamic framework that can transform real-life diagnosis into virtual diagnosis. More specifically, the followings are our research objectives.

- We aim to involve heterogeneous digital health data sources to gather comprehensive well-being data.
- We plan to map and pre-process heterogeneous data.
- We focus on considering multiple classifiers for multiple disease/disease risk prediction.

Our research statement in this thesis is: *To what extent a well-being digital twin (WDT) can support the early prediction of multiple diseases anytime and from anywhere for an individual?*

1.5 Thesis Contribution

The prime focus of this work is to design and evaluate a dynamic framework for multiple disease risk prediction. During this work, we have made the following contribution.

- Design and evaluation of algorithms to select suitable ML algorithm dynamically for disease dataset.
- Design and evaluation of a dynamic testing method for prediction.
- Evaluation of the performance of the proposed framework for three different use cases.

1.6 Thesis Organization

We have organized our thesis as follows:

- In Chapter 2, we present a brief overview of the digital twin for health and well-being. This chapter helped us to understand the requirements for designing the proposed framework. In addition, we discussed existing works to design a framework that can address the requirements.
- In Chapter 3, we explain the proposed framework and method. Individual sections of the chapter include the description of proposed methods, algorithms, and the competitive advantage.

- In Chapter 4, we discuss the implementation. This chapter includes tools, technology and implementation of proposed algorithms.

- In Chapter 5, we detail our evaluation, including experiments. Firstly, this chapter includes three different use cases for the framework. Then it presents an overall experiment to evaluate the performance of the disease classifier. Finally, it verifies comparing the performance with existing works.

- In Chapter 6, we conclude and discuss scopes for future work. This chapter includes issues we addressed, limitations and possible improvements to our research.

1.7 Scholarly Output

The scholarly outcomes from our thesis are given following.

- Ferdousi, Rahatara, M. Anwar Hossain, and Abdulmotaleb El Saddik. Early-Stage Risk Prediction of Non-Communicable Disease Using Machine Learning in Health CPS. IEEE Access 9 (2021): 96823-96837.

- Hossain, M. A., Ferdousi, R., Hossain, S. A., Alhamid, M. F., & El Saddik, A. (2020). A Novel Framework for Recommending Data Mining Algorithm in Dynamic IoT Environment. IEEE Access, 8, 157333-157345.

- Hossain, M. Anwar, Rahatara Ferdousi, and Mohammed F. Alhamid. Knowledge-driven machine learning-based framework for early-stage disease risk prediction in edge environment. Journal of Parallel and Distributed Computing 146 (2020): 25-34.

Chapter 2

Literature Review

This chapter is organized into two parts. In the first part, we present the background study of a digital twin for health and well-being to understand the requirements for designing a disease diagnosis framework. In the second part, we present related works and it's gaps to address the requirements found in the first part.

2.1 Background

Digital Twin (DT) has gained success in the production industry, and it is now getting attention in the healthcare industry in the form of well-being Digital Twin (WDT). In this section, we present an overview of WDT to understand its potential scope, architecture and impact. We discuss the evolution of DT architecture, the definition of WDT, the difference between product DT (PDT) and WDT, the benefits, the challenges, and the considerations for WDT, the different types of DT for health and well-being and the progress WDT is having in development.

2.1.1 Definition of Well-being Digital Twin

This section presents definitions of DT in health and well-being. The goal of this section is to pick the suitable definition from the literature for WDT model.

From 2019-2020 researchers have increasingly studied DT in the well-being industries. The authors in these publications have described DT from diverse perspectives to fit well-being. Based on these studies, we discuss four remarkable definitions of DT in the context of well-being in Table 2.1.

Table 2.1: Definition of DT in the context of well-being

Reference	**Definition**	**Features**
[31]	Digital Twins enable the monitoring, understanding, and optimization of all functioning of humans, and provide constant health insight to improve quality of life and well-being.	Data visualization Prediction, Intelligence, Analysis, Decision making, Feedback loop
[16]	Personal Digital Twins are data-driven solution that depicts individuals' health status based on regularly collected health parameters.	Data visualization, Data monitoring
[78]	DT is a digital representation of a human in a computer or a server on the cloud.	Data monitoring, Data visualization
[65]	A digital twin is a simulation technology that delivers digital health insights while also allowing prediction and recommendation within a feedback loop.	Data visualization Prediction, Simulation, Analysis, Recommendation, Feedback loop

It can be observed from the above table that Data visualization and monitoring are common as well as innate features of DTs [30, 31]. In addition, predictive well-being and personalized well-being are the two fields of well-being that can be aided by DT. Interestingly, both fields require intelligence to satisfy the goals of edge applications [36]. The

authors in [31] addressed the importance of this fact and attributed DT with intelligence.

The integration of intelligence in DT has become crucial to support the needs of current well-being applications. In our opinion, since AI acts as the brain of the DT, it is highly required that health twins have proper intelligence to support the decision-making of its application.

The prediction process in DT applications is often supported by data science and machine learning algorithms (MLA). However, the more specific explanation would make the decision-making process of DT trustworthy and meaningful [34, 70]. For instance, if a health twin is used to monitor diabetes risk factors from activity history (e.g., exercise, steps, bpm, etc.), it should also be able to show the contribution of potential risk factors for an individual. This could be an option to embed explainable intelligence health twin.

It can be observed form Table 2.1, that the definition by El Saddik [31] covers most of the features so that WDT fits various applications of healthcare. We found that in this definition DT is defined universally for health and well-being domain.

2.1.2 Benefits of WDT

In recent years, growing research is welcoming Digital Twin in the well-being sector. This section presents the benefits behind the increasing demand for studying DT for health and well-being.

1. WDT can support COVID-19 response applications for virtual health checkup [78].
2. WDT can save the cost and time to test treatments [16].
3. Hidden health insights could be understood [38].

4. Continuous data visualization [71] and test simulation is offered by DT technology.

5. Treatment plan can be evaluated without involving or harming real human [34].

6. "What if analysis" feature of DT could be beneficial for planning treatments [31].

7. Early and emergency prediction facility could be enjoyed anytime [38].

8. Personalized (patient centric), and preventive well-being digitalization could be supported broadly with WDT [31].

It can be observed from the above points that due to the outbreak of COVID-19, early and emergency prediction has become a consideration for the well-being of individuals. Due to the predicting capabilities of DT through ML, the digital twin has gained popularity to design various frameworks for well-being.

2.1.3 Digital Twin Frameworks

In this section we discuss components and features of significant DT frameworks till 2021 to find out which framework suits more in the context of WDT. The goal of this section to find the type DT framework of DT suits our research problem. More specifically, which part of the DT framework require more focus for digital twin diagnosis.

Based on our review, the revolution of DT architecture has been illustrated in Figure 6.2. It can be observed that the DT architecture has been experiencing a rapid transformation from 2014 to 2021 and onwards. The four types of DT frameworks are the following.

1. 3-Dimensional DT [56]

2. Cloud CPS based DT [12]

3. Intelligent DT [30]

4. Industry 4.0 DT [11]

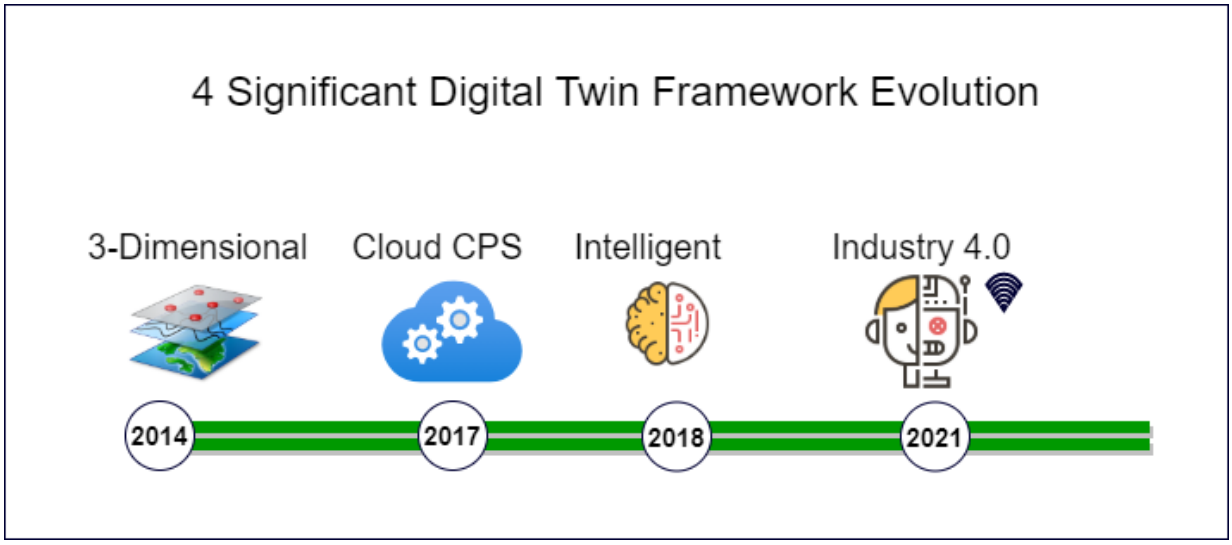


Figure 2.1: Evolution of Digital Twin framework

Grieves first introduced the standard three-dimension DT architecture in his product lifecycle management (PLM) course at the University of Michigan in 2003. This can be regarded as the pioneer DT. The conceptual framework by Grieves proposed three basic components [56].

1. Physical space- consists of aircraft, radar, related real infrastructures, and assisting functionalities.

2. Connection – represents the physical data virtually and provides information for controlling the physical space.

3. Virtual space- presents the virtual counterpart of the original mission-related infrastructures.

The basic 3-layer DT frameworks are comprised of three major processes [12]- calibration, control, and augmentation. In detail, the AR/VR system in the frameworks collects data from the virtual part and intelligently presents it to the user after the calibration procedure. However, these three-dimensional frameworks could not handle open and user-oriented broader applications such as agriculture, well-being, and medicine for real-time decision-making [86]. The reason behind this is that Greive's architecture did not consider the application of the Software as a Service (SASS) category [11]. The SAAS applications are cloud-based software systems that provide services over the Internet. In this era of IoT, most of the applications have shifted to SAAS to deal with Big Data. Dropbox, G suites, Amazon web services are some common examples of SAAS applications [92].

A Digital Twin is perhaps a subset of the Cyber-Physical System (CPS), and both lead to the smart manufacturing idea. There is indeed a strong distinction between the two technologies: the CPS is tied to science, while DTs are intricately linked to technological advances. Both contribute to smart manufacturing [62]. The CPS-based DT framework has been widely proposed by researchers.

CPS-based architectures have opened a door to establish a bridge between IoT and DT [12]. Therefore, DT is no longer only tied to the manufacturing industry and has gained potential to advance other industries including farming and well-being.

Nowadays, well-being applications have a diverse range of goals due to the increasing number of connected things. This scenario has raised the demand for intelligent architectures. Because, the AI interference engine, data mining/analysis techniques to capture qualitative and behavioral information, as well as analytics to provide real-time tracking,

forecasting, and collective decision making. Therefore, the intelligent DT architectures are more suitable for WDT [31]. In addition, potential architectures are the industry 4.0 architecture. However, the core concept of Industry 4.0 focused can be explored through hybrid Cloud CPS and AI DT architectures.

The authors in [31], proposed a ecosystem (Fig. 2.2) that includes a communication model for the interaction between a real Twin and Digital Twin. This communication model includes three major parts.

1. Sensing/Actuating,
2. Intelligence, and
3. Representation of the Twins connected through a Tactile Internet.

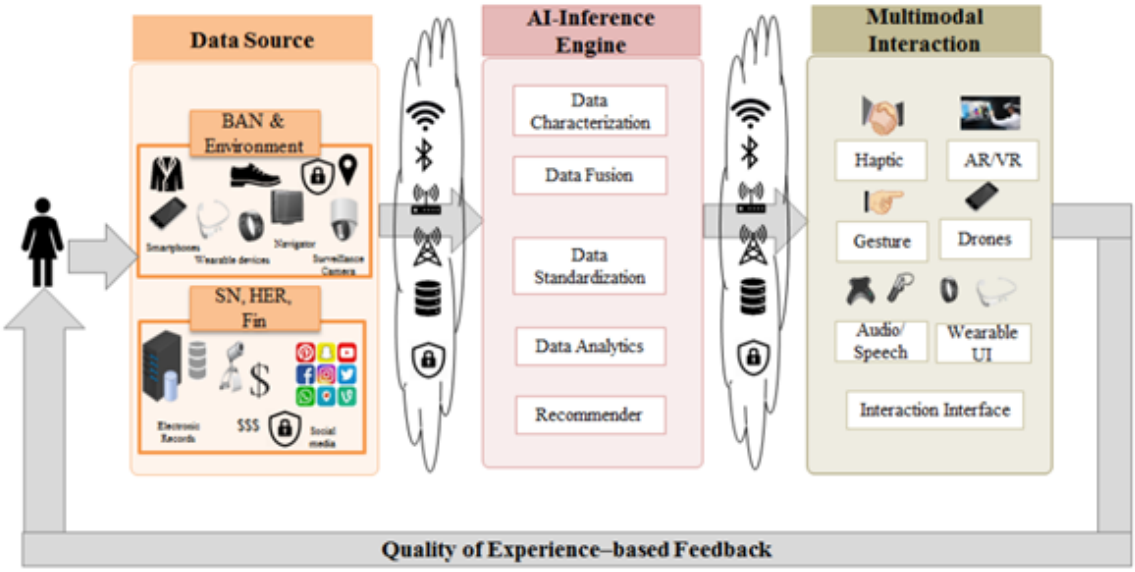


Figure 2.2: Ecosystem of the Digital Twin for health and well-being. [31]

The authors compared Digital twin sensors, for instance, IoT, haptics, etc. to human's five senses of real Twin. The human brain is compared with the machine intelligence of the Digital Twin. An example of the Ml based DT framework can be a DT having the capability of predicting the risk of diabetes from a person's activity history and provide recommendations (e.g., have a walk).

Therefore, we found that the intelligent DT are more suitable for our research problem-disease diagnosis. In addition, the AI inference engine is the prime focus to design a framework for digital twin diagnosis.

2.1.4 Technologies for WDT Framework

This section discusses the underlying technologies for a WDT framework. The purpose of this discussion is to understand the required technologies and capabilities for a WDT framework.

DT could be used to enjoy the full potential of AI-enabled healthcare. Although AI is related to IoT and CPS, we have focused on the basic technologies to develop a full-fledged WDT. Let us understand this by an example.

Suppose that we want to create a DT of mental well-being with a goal of personalized stress prediction and monitoring. To implement this, we will need the following steps.

1. Initially, we will need smartwatch exercise data (daily activities), social media histories, phone logs, etc. [51] Here we will need IoT.

2. Then, to use these sources to get data we may need computation algorithms, which can be done using CPS [91].

3. After that, we will need to prepare the heterogeneous data from diverse sources that we have collected for predicting stress. Pre-processing is done to fit data to the prediction algorithm. Data Mining is employed for this purpose [65].

4. Then with the preprocessed data, classifiers will be built by training classification algorithms (Support Vector Machine, Decision Tree, Random Forests, Bayesian Nets, etc.). This is supported by ML technology [38].

5. The DM and ML will be combined to build Artificial Intelligence in the WDT [31].

6. Finally, the desired mental health twin will be prepared with the capabilities of stress management.

From this example, we can observe how the emerging key technologies represent mental well-being to predict a health issue (stress). The WDT from the above process will provide a way to visualize numerous data views of stress and predict whether an individual is stressed. However, to control the risk factors (e.g., physical activities, social activities, bio-sensor readings), the WDT needs to support multiple diagnoses. Because the risk of disease varies from person to person, the same individual can suffer from multiple problems. Therefore, a DT framework for health and well-being needs to consider heterogeneous data and employ multiple disease/disease risk prediction mechanisms using a dynamic ML process.

2.1.5 Special Considerations for WDT

We have compared the well-being Digital Twin (WDT) with Product Digital Twin (PDT) and discovered the distinct subjects between these two types of DTs. The comparison was conducted to understand special considerations for HDT.

In an earlier period, the key research concern was how PDT can be constructed [92]. Health domain applications usually involve patient monitoring, health monitoring, disease prediction, and other well-being applications. As soon as DT has been considered to aid the health domain, the research focus was moved to address - "how the digital replica of a human can be created?" [78]. The key differences between product DT and WDT are tabulated in Table 2.2.

Table 2.2: Differences between Product Digital Twin and Health Digital Twin

Subject	**Product Digital Twin (PDT)**	**Health Digital Twin (WDT)**
Mental State	Product does not have mental state correlated with all other factors.	Health risk factor can be governed by human mental state.
Nature of Rules	The rules of product physics are almost similar and fixed for same category.	Rules for human physics may vary from person to person, even day to day.
Social Media	Product life cycle cannot be affected by social media.	DT has strong correlation with social media.
Mapping Complexity	Mapping product status digitally is less complex than health status.	Mapping health status digitally is more complex than product status.
Data	Capturing product data needs less pre-processing.	Health data are more heterogeneous and unstructured.
Data Preprocessing	Less rigorous than Health Twin	Rigorous data preprocessing is required.
Ethical Consideration	Product DT is free of complex ethical consideration.	Human life is precious and systems for human health require to consider several ethical concerns.

Based on the differences presented in Table 2.2, there exists a couple of challenges, for example, the Fitbit can collect and represent the beats per minute (bpm), step count, etc. using the accelerator, gyroscope, and pressure sensor. However, various physical states are extremely complex to capture. For instance, "Skin rubbing count" as irritation, and "Count of food intake" as polyphagia are two important health insights to predict diabetes risk

[45]. Specially designed biosensors are required to retrieve this status. Furthermore, some biosensors may not be viable to implement or may require complex and time-consuming development [61].

Another notable difference between PDT and WDT is social media. in this digital era of connectivity, human has an additional life affecting their mental as well as physical state- the social media life. For instance, if a person is disturbed by Facebook posts, comments, messages, only physical data will not be enough to predict his stress level [16].

Several researchers have suggested data mining as a vital technology to mitigate health data mapping complexity [41]. In one of our works [42], we proposed a knowledge-driven epidemiology libraries to identify and predict associated risk factors of disease as well as the recommendations. Precisely, some missing data can be mapped using ML and AI [33]. The key idea here is predicting the values of an attribute based on other correlated attributes. However, here comes the new challenge- heterogeneity of data [65]. To avail the benefit of ML or AI for mapping health insights, we need to handle heterogeneous data from diverse sources. For instance, electronic health records (EHR), historical health data, continuous health status, social media activity, raw sensor data are some notable sources of health data [16].

Finally, human-related systems have to deal with several ethical facts, especially when it is an autonomous system. Comparatively, the degree of ethical concern in WDT is more complex than the PDT [23, 26].

2.1.6 Key Challenges

In this section, we present various challenges to design a WDT framework for diagnosis.

- **Technical Issues.** The seamless feature and capability of handling heterogeneous data with interoperability standard proposed in [51], has made DT a four-in-one technology to represent digital health [46]. DT could be used to enjoy the full potential of AI-enabled smart well-being. Although AI is connected to IoT and CPS, here we have focused on the basic technologies to develop a full-fledged WDT. Let us take the following scenario to explain the WDT.

- **Data Bias.** The prediction by WDT can suffer from racial or other biases and cause inequalities in health care [87]. A model trained with computer-identified features one wrong label is threatening to the well-being. For instance, if a classifier is trained with available sensor data and ignores notable features of diabetes prediction because those were not measurable by the sensor, the result of prediction will lose reliability [23].

- **Level of Autonomy.** To what extent patients can access autonomy is another ethical concern. Autonomous Clinical Decision Support System (CDSS)is very risky in some cases. If the classifier makes an irrational prediction and associated recommendation and the patient starts to follow those recommendations it may harm health. For example, if sleep 8-10 hours is identified as insomnia and the system recommends increasing sleep hours as a precaution, it can put health at more risk. Which level of autonomy can be offered to the patient is one of the key ethical concerns [87].

- **Trust in Intelligence.** AI is still in the process of establishing [87]. The context of product and well-being is far different from each other [26]. If the accuracy of a system is 76% [69] it refers to the fact that 24% is incorrect. In the case of well-being,

it may be proven fatal. For example, the classifier may predict that a person has a lower risk of diabetes, while he has a higher risk. *How medical trust can be assured?*, is a big ethical question. In other words, transparency in machine learning-based prediction is required [22, 23].

- **Data Visualization Issue** Rigorous data pre-processing may provide better accuracy, but this may bring another ethical issue which may hide visualization of real health issues.

- **Consent of Human.** In general, human is the key input source for WDT. As the WDT system may need data sharing and collection using third-party applications or labeling by human intervention. Taking patient's consent and allowing them to modify consent should be implemented. Despite ethical challenges, the existing literature indicates some other challenges for WDT. In the next section, we will discuss those challenges.

2.1.7 WDT in Industry

Digital Twins are utilized in a variety of certain other industries to monitor, maintain, and simulate probable consequences if any problems emerge while the apparatus is in use. Since well-being costs are rising globally, and the world's population is growing, it is the right time to adopt Digital twins to improve the system and provide a more efficient solution for both well-being professionals and patients without causing no harm to either.

According to reports, Digital Twins might bring a 900% cost savings in hospitals and a 61% reduction in blue code hospital incidents, which includes emergencies in adult well-being [5]. Many prominent names are competing to be a part of the development of the

Table 2.3: Industrial progress on DT well-being

Company	**Product/Service**
Simens [9]	3D Digital Heart Twin, that facilitates doctors to simulate surgical procedure and to verify tests on patients causing severe injury. One of the first full-fledged ward management twins.
GE well-being [4]	Simulation to assist in sensualizing data from multiple sources in order to generate a Digital Twin of the hospital for testing alternatives. Plethora of hypothetical scenarios to be examined, all at a low risk.
IBM [6]	Efficient and personalized patient centered treatment using Digital Twin of patient.
Dassault Systèmes [2]	3-D models of a live heart, on which artificial silico models can be run and cardio research can be conducted
NHS [7]	A testbed for testing whether low-cost 5G connectivity aids technologically deprived people by offering consistent access to digital community and personal solutions.
Digitwins [3]	a comprehensive modelling system that would allow numerous treatment simulations to be done without causing harm to the patient.
Phillips [8]	AI-enabled cardiac models that can, convert 2D ultrasound images into data that doctors may use to identify issues or automatically analyze scans to help surgeons plan procedures.

first full-fledged Digital Twin. Some remarkable names Siemens, Phillips, and IBM are all leading runners. These leading companies are utilizing their massive databases and financial muscle. Other companies, on the other hand, are beginning to experiment and push the boundaries further to the growth of Digital Twins. In Table 2.3, we have provided information about the industrial progress of DT in well-being.

After reviewing the current industrial advances, we found that digital twinning of equipment, wards, medical information, and patients is critical for the company engaged. Also, the digital twin diagnosis for early risk prediction of disease is still less explored.

2.1.8 Digital Twins for Health and Well Being

This section present an overview of various types of DTs proposed in the well-being domain.

- **Health Twins.** A recent study [71], has investigated Digital Twin studies in the health domain from the perspectives of patient monitoring, the pharmaceutical sector, hospitals, and wearable technologies. They anticipated that AI would play a pivotal role to accelerate DT in the well-being industry. In this study, the authors discussed various use cases of using DT such as clinical decision support system (CDSS), surgery planning, and medicine prediction. The authors emphasized that various remedial forecasts can be produced if the ML and AI approaches are employed in Digital Twins. The predictions can optimize processes and information usage. However, they did not elaborate on how ML and AI could advance WDTs.

  On the contrary, the authors in [31], narrated that the AI-interference engine has treated data characterization, standardization, analysis, prediction, and recommendation together as the enabler of ML and AI in WDT. Similarly, in [34], the authors considered data mining as a powerful tool for enabling simulation in cloud WDT.

In [16], the authors have considered the heterogeneous data collection issues and proposed a standardized ISO/IEEE 11073 DT framework architecture for health and well-being. This model provides the process of collecting data from personal health devices, processing that data, and providing feedback to the user in a closed feedback loop. The ISO/IEEE 11073 enabled AI well-being systems can subdue the data interoperability issues in well-being. However, data mining not only supports obtaining powerful input datasets to feed simulation models but also supports efficient processing to understand simulation outputs. In general, the process is obtaining a balanced dataset for decision support [65].

In [87], the authors recommended DTs as a breakthrough tool for care management of persons with multiple sclerosis (pwMS) to cope with the complexities of this chronic, multidimensional condition at the individual level. They highlighted that AI-enabled analysis could be employed to develop DT of patient characteristics. More specifically, the potentiality of AI on various disease parameters such as clinical and para-clinical outcomes, multi-omics, biomarkers, and patient-related data was discussed for handling the heterogeneous and vast amount of patient-related data. The authors in [16], also emphasized the fact that DT can handle a diverse set of health parameters for decision making.

The authors in [54] proposed CloudDTH to address the issue of real-time supervision and the accuracy of crisis warnings for the elderly in well-being services. Similar to [36, 55] the model in [54] adopted [30]. Although classifier interpretation like [18], could contribute to this framework, the authors preferred the monitoring process as a black-box prediction. The what-if analysis could not be supported fully with the framework.

- **Healthcare Center Management Twin.** In [26], the authors present the concept of agent-based WDT by merging the DT notion with agents in a modeling and simulation framework based on mirror worlds. The key idea of work was designed for Trauma management. In simple words, their DT symbolizes the operative phase of trauma rehabilitation, which begins when the trauma is classified as severe in the preceding phase. It can be observed clearly that prediction and the use of intelligence are required to support this predictive software agent. To implement the semantic reasoning capability of the software the author sought to employ semantic web technology. However, one of the key challenges to implementing semantic web is a heterogeneous representation of evolving ontologies, which is obvious in the health domain. Furthermore, the ethical implication of WDT requires a trustworthy and authorized source of knowledge 'as it's related to humans [22, 23].

  In [49], the authors proposed a HospiT'Win framework that can forecast unforeseen events earlier to determine the impact on the hospital and feasible strategies to mitigate the harm. Furthermore, they proposed a method to connect the HospiT'Win with a real hospital to enable the tracking, monitoring, and validating that the hospital functionality is going in the proper direction and at the correct time. The authors realized that IoT AI, BAN will be the core technologies to implement their idea. Precisely, they employed artificial intelligence interventions to decide the suitable scenario to practice in a real hospital, incorporating numerous parameters regarding risks, finances, and so on. This part of the study was proposed to activate validation.

- **Organ Condition Twin.** In [55], the authors presented the Cardio Twin architecture for detecting ischemic heart disease (IHD) on the edge. They used a CCN to classify non-myocardial and myocardial diseases. The authors classified features

from electrocardiograms and completed the assignment with 85.77% accuracy. In Cardio-Twin, the authors employed CNN-based classification to apply intelligence for meaningful Data visualization.

Similar to [55], machine learning was considered as the key enabler of DT coaching in [36]. Likewise, [55] authors designed a DT coach system based on the DTwins ecosystem in [30]. These two works evaluated the universal DT well-being ecosystem in a specific application context. Inevitably, the authors in [36, 55] obtained better accuracy to prove the implementation of DT on the edge.

The goal of [18] is resembling [36], that aimed at indulging DT as an alternate of "human in the loop" in data noise reduction. Likewise, [36], the authors also worked on a smart fitness management system to monitor athletes and recommend preventive measures for better fitness. The authors preferred KNN for data noise reduction and classifier interpretation to analyze individual health parameters. Precisely, based on the user's activity history the fitness management system can recommend which behavior needs to be changed.

For instance, increase carbohydrate in 3 units. To retrieve the suggestions using ML, the authors employed a Counterfactual classifier explanation algorithm [88, 89]. The key idea of this algorithm is to provide values of different attributes for a particular class prediction.

The authors used greedy algorithms to optimize the top 5 attributes contributing to the classifier's outcome. The works presented in this paper showed a remarkably interesting way to embed intelligence in DT predictive system. However, the counterfactual algorithm cannot show any range or threshold to understand how different parameters are contributing to a decision. Moreover, the suggestions need to be

defined by involving the rule provider (human). In addition, the counterfactual algorithm suffers from the Rashomon effect [88, 89] that causes multiple explanations for each instance. This poses another level of complexity to analyze which explanation to pick.

By contrast, in [69] the authors recommended explainable AI (XAI) based DT as a solution to retrieve classifier prediction for the clinical decision support system. They conducted an empirical analysis on a Liver disease prediction using an SVM classifier. They utilized the Lime XAI algorithm [39, 89] to interpret classifier prediction. The benefits of using Local Interpretable Model-Agnostic Explanations (LIME) [10, 28] are that it is easy to implement, supports heterogeneous types of datasets (e.g., clinical data, local health records) in diverse format image, mixed, nominal, binary, etc. In addition, multiple classification algorithms like decision trees, random forest, Bayesian networks are supported by lime. Another notable feature of Lime is that it can be handled dynamically or in default mode. Therefore, if any CDSS demands using all features it can be supported. Furthermore, if the system requires selective or optimal features it can be supported. In the context of well-being, it's one of the key requirements, that data diversity is handled best. The key difference between the counterfactual algorithm and the XAI algorithm is the way of interpreting the classifier.

The counterfactual algorithm directly shows a value of attributes while the XAI algorithm shows association and comparative relations with a threshold. For example, if the risk class is 1 then how many attributes are contributing at the side of 1 and how many are contributing to 0 is demonstrated. The XAI opens a prospective door for WDT to mitigate the lack of accountability of prediction issues. There exists another algorithm Shapely [28] for implementing XAI, which has modified LIME.

However, the output of Lime is more convenient to visualize. Unlike counterfactual algorithms, it requires less rigorous development for selecting explanations [89].

- **Cardiovascular Twin.** In another work, the authors used neural networks to predict abdominal aortic aneurysm (AAA) and its severity employing the inverse analysis methodology of the DT in [24]. This study also achieved an acceptable level of accuracy of 97.79%. A deep neural model was employed to capture the bi-directional context links between dangerous code phrases in [91]. The authors aimed at confirming cyber resilience on well-being big data on Lung cancer. Their Bi-LTSM model performed with better accuracy than other ML-based DTs.

  The IoT context-aware DT in [32], predicted cardiovascular conditions from ECG data with various machine learning algorithms above 90% accuracy for each. The proposed ECG heart rhythm classification also performs with the highest accuracy [55]. However, how this proposed ML-based prediction is different from regular ML prediction is not clear from the study. Furthermore, how trust can be added to the prediction was not addressed.

The studies discussed above are mainly of two categories. Some of the studies have been conducted to present the state-of-the-art of DT, other have been conducted to address several health issues as well as to satisfy diverse range of goals. In Table 2.4, first, summarize the application-specific literature in terms of the type of well-being domain, application goal key features, fundamental technologies addressed by the contemporary WDT researchers. In addition, we tabulate the proposed twins and associated threats in the studies.

Table 2.4: Summary of work related to WDT

Reference	**Published in**	**well-being Domain**	**Application**	**Key Features**	**Enabling Technologies**	**Threats**
[54]	2019	Personalized, Preventive	Elderly Health Monitoring	Regular monitoring	IoT, AI	Integration of AI
[71]	2019	Participatory, Personalized, Predictive, Preventive	Smart well-being	CDSS, Drug Discovery	DM, ML, AI	Data Heterogeneity, Integration of AI
[69]	2019	Participatory, Personalized, Predictive, Preventive	detection of Liver disease	CDSS	ML, XAI: Lime	Lower accuracy in prediction
[55]	2019	Preventive, Predictive, Participatory	Ischemic Heart Disease Detection	CDSS, Regular Monitoring	ML: CNN	Recommendation extraction
[49]	2019	Preventive, Predictive	Hospital's Anomaly Prediction	Abnormal event Prediction	ML	Handling data heterogeneity
[51]	2020	Personalized, Participatory	Smart well-being	Regular Monitoring	DM, ISO/IEEE 11073 Standard,	Big Data handling

[19]	2020	Participatory, Personalized, Predictive, Preventive	Fitness Management	Regular Monitoring, Recommendation	ML: KNN, SVM Counterfactual algorithm	Recommend extraction
[20]	2020	Participatory, Personalized, Predictive	Personalized Medicine	Drug discovery	ML AI	Integration of AI Handling multiple health parameters
[36]	2020	Participatory, Personalized, Predictive	Coaching Athletes	Regular Monitoring	ML: CNN	Prediction result is not explainable
[26]	2020	Participatory, Personalized, Predictive	Trauma Management	Monitoring Trauma Center	AI Robotics	Integration of AI Handling multiple health parameters
[87]	2021	Participatory, Personalized, Predictive, Preventive	Health Monitoring of Multiple sclerosis	CDSS	ML DM	Handling multiple health parameters

[32]	2021	Participatory, Personalized, Predictive, Preventive	Heart Condition Classification	CDSS, Regular Monitoring	ML DM	Handling multiple health parameters
[24]	2021	Participatory, Personalized, Predictive	Blood circulation Analysis	Regular Monitoring	LTSM, ML: MLP	Prediction is not explainable

In summary, we highlight the following points presented in Table 2.4:

1. The WDT has been widely proposed for P4-medicine Personalized, Predictive, Preventive, and Participatory. The real-time monitoring, prediction, intelligence and simulation feature of WDT made it possible to support multiple domains of well-being.

2. Either health twin or Organ Twin has been proposed in literature. Some authors have proposed treatment organization management [26, 49].

3. CDSS, precision medicine and regulatory monitoring are common application goal of the WDTs. Since DT can offer the best of four trendy technologies IoT, CPS, DM, ML, it has been warmly accepted for the decision-making and data visualization in well-being.

4. IoT and different ML algorithms have been utilized to construct several types of WDTs.

5. Handling multiple sources and black-box prediction are two common threats in the current studies. The nature of the applications often requires explanation for the prediction.

6. Counterfactual algorithm and XAI algorithm are two solutions that have been proposed in the literature for retrieving explanation of prediction.

7. Some recent studies have taken the ethical challenges of DT into account [76]. In the next section we have discussed the ethical consideration or DT.

2.2 Related Work

We studied specific works for designing a dynamic MLA selection and for proposing a dynamic testing method for prediction. We present the works in Section 2.2.1, and Section 2.2.2 respectively.

2.2.1 Dynamic MLA Selection

This section comments on existing work that are relevant to machine learning algorithm selection. The goal of this section is to understand the gaps in the relevant works.

The huge volume of data from diverse domains has made ML an emerging field of digital twin applications. The practical WDT applications are dependent on numerous MLA to enable decision making. [67].

The selection of MLA based on the criteria or knowledge of goals has been proposed by some researchers [13]. In [13], authors found that defining an unambiguous goal can effectively solve the selection problem of machine learning algorithms. They offered an

expert group-based criteria selection method, called optimum performance ranking, which is based on evaluation metrics like fitness function, statistical measures, and constraints during analysis. This work mainly focused on the supervised machine learning methods to conduct the empirical study with numerous experimental prove. However, as IoT is also trending toward closed-loop systems, such as cyber physical systems (CPS), the MLA should also use unsupervised learning to ensure the robustness of the systems, where the processing is independent of a large training dataset [85].

Again in [13], authors found that a meta-learning based framework can resolve the problem of finding a best-fit MLA to classify a particular dataset. They also mentioned a selection of mapping algorithm using meta-feature of problem, performance, feature, and algorithm space.

The authors in [64] emphasized that data characteristics have a significant impact on the performance of different classification methods. Thus, they conducted a study by considering both accuracy and time complexity to demonstrate how different characteristics of the dataset as an independent variable impact their results with decision tree algorithms. Instance spaces [59] and feature selection [21] have also been addressed as a criterion of selecting appropriate MLAs [35]. The works discussed above mainly focused either on mining traditional datasets for classification or on different supervised machine learning approaches that require increased human intervention. on the contrary, the work in [60] considered minimal human intervention and supervision into account in the context of big data analytics. This work described the potential research methodologies and activities comparing and describing the implementation process and tools for the development of a big data system. The selection of MLA based on the criteria or knowledge of goals has been proposed by some researchers [21, 59].

In a nutshell, least attention has been given on dynamic MLA selection for predictive

system. The manual approach of selecting MLA for AI inference generally provides sub-optimal performance [27]. Those methods are not only inefficient but also require significant human intervention as the selected algorithm works for any particular dataset [60].

Moreover, to satisfy our research objective- multiple disease/disease risk prediction, such an approach is not a good fit. Because this trial and error based MLA selection approach to find which one gives the best accuracy result in redundant efforts, complexity, and costs [14, 53]. The proposed Dynamic MLA Selection method aims to fill this gap.

2.2.2 Dynamic Testing for Prediction

This section comments on existing work that are relevant to AI-based approach such as deep learning (DL), ML-based approach in predicting health issues. The goal of this section is to understand the gaps in the existing work for healthcare prediction.

The survey in [79] reveals that the emerging health applications increasing need to include machine learning to provide innovative and smart services. The authors further describe the conversion of raw physiological inputs into functions and how those are used in ML, analyze the suitable ML algorithms, and describe how decisions are made and propagated to the user.

In [40], the authors introduce a detail taxonomy for CPS in healthcare based on a comparative review of components and procedures. The taxonomy includes information about HCPS application, architecture, sensing approaches, data handling, computation, communication, security, and control, which can be consulted when developing HCPS applications.

The Quality of Service(QoS) issues have been studied in the context of remote healthcare in [92]. The work discusses the resolution of QoS challenges in urban healthcare big

data system [25]. While it addresses the problems of healthcare and physical CPS systems, information about how IoT-sensor data can be analyzed intelligently for NCD predictions has not been made available.

The author in [44] proposes a CPS that incorporate localization information on the sensing, analyzing and sharing of patient data for continuous health monitoring, however, there is no indication of risk prediction of any particular disease in the work.

In the area of general healthcare monitoring, the work in [58] shows a CPS implementation to monitor blood pressure (BP), blood glucose (BG), body temperature (BT), and heart beat rate (HR) based on embedded and cloud-based technology. This approach interconnects the communication, computation, and control aspect of CPS for continuous monitoring of patients and actuate remote treatment method when necessary.

The authors in [52] propose a CPS architecture for timely detection of stroke, a common NCD in patients, to minimize the risk in people. The system analyzes electroencephalography (EEG) data and connects to physician when it identifies stroke occurrence, and sends alerting message to the concerned personnel. However, it does not focus on early prediction of stroke.

The work in [82] proposes a model to predict, monitor, and control the risk of coronary heart disease in CPS context. They authors use ANFIS fuzzy inference system to identify the different levels of risk assessment. They define 800 plus rules to determine the risk level and the consideration of additional attributes will require them to add even more rules, thereby increasing the overhead. Contrary to this work, our proposed method introduces verified training dataset against which ML classifier is built to predict the early risk level of diabetes from wearable sensor data.

2.3 Summary of Requirements

We conclude that the existing work are designed for specific problems and mostly limited to predicting single problem. Therefore, we propose a dynamic testing method for prediction for supporting multiple disease prediction. We summarize the requirements for digital twin diagnosis that we find important to tackle and implement in this thesis.

1. **Heterogeneous Data Source:** A WDT framework connecting multiple health data sources, including EHR, Social media, Wearable sensors, etc., will be able to support multiple diagnoses. For example, various non-communicable diseases (NCDs) like diabetes, stroke, a heart attack can be diagnosed from the same model.

2. **Dynamic Algorithm Selection:** Different algorithms are good at handling datasets with different properties [43]. Therefore the framework will require selecting suitable machine learning algorithms dynamically.

3. **Prediction from EHR:** Creating a vast knowledge base is complex, costly, and time-consuming. Using EHRs to train the classifiers and extract rules and ground truth can reduce the cost and time for manual data pre-processing and data labelling. Because EHRs like- prescriptions, diagnosis records, etc., includes medical practitioner's identification by default. In addition, it includes the status of the patient's signs, symptoms, risk factors, and numerous patterns.

Chapter 3

Digital Twin for Disease Diagnosis

This chapter first provides a high-level overview in order to put the thesis objective in context. Then we elaborate the framework for dynamic machine learning algorithm selection. This is followed by our proposed method of dynamic testing for prediction.

3.1 High-level View of The Model

Figure 3.1, demonstrates a high-level view of a WDT model. The proposed WDT model is mainly an ML-powered framework that selects Machine Learning Algorithm (MLA) dynamically for the prediction purpose. The WDT follows an ecosystem similar to the one described in [31]. In [31], the AI Interference Engine is generic. In contrast, the proposed AI Inference focuses on dynamic selection of MLA for multiple disease prediction which we detail in further section.

Overall, the WDT model selects suitable ML algorithms (MLA) for each disease dataset. Multiple classifiers are then trained with these selected MLA. These classifiers predict

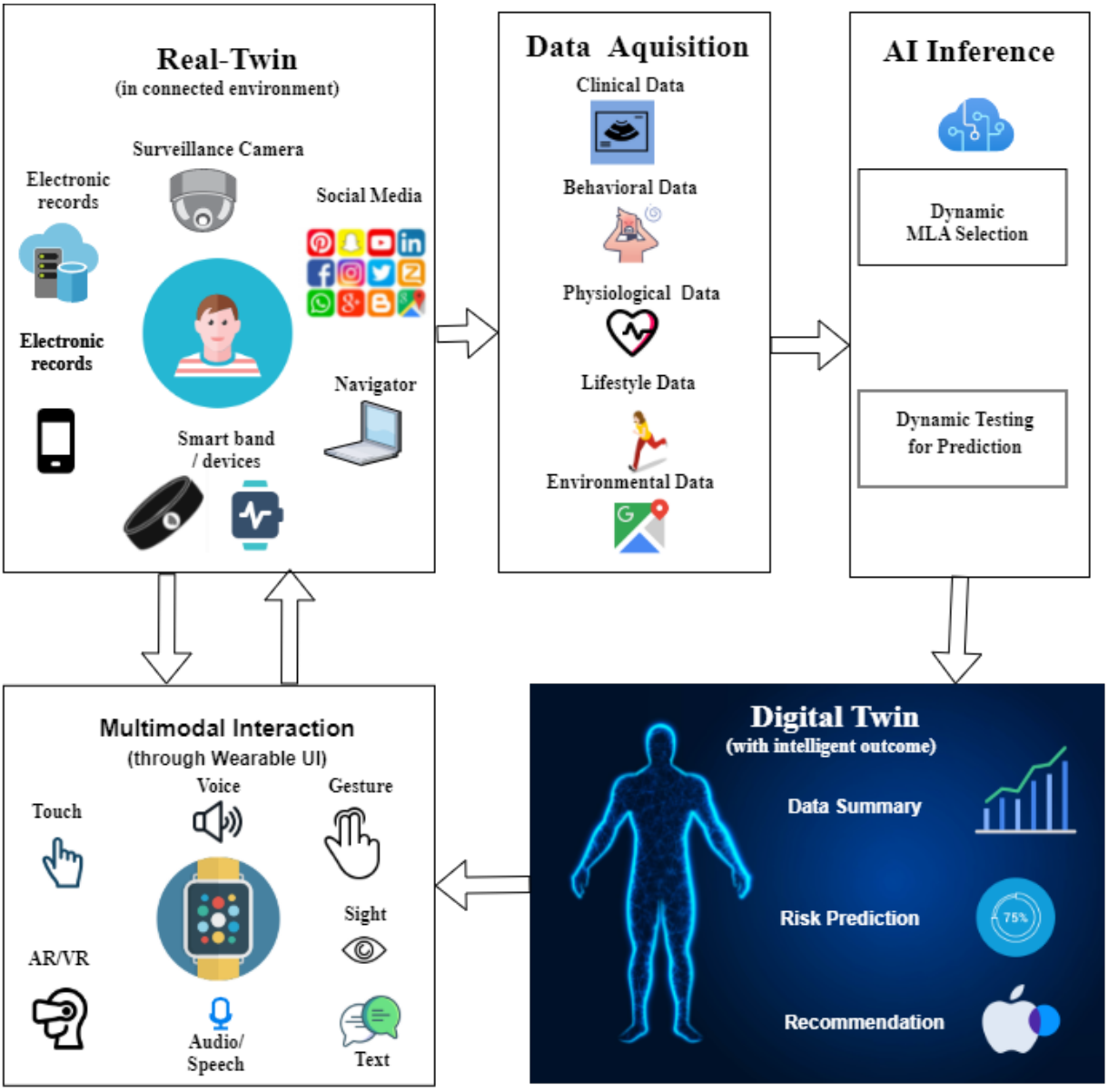


Figure 3.1: High-level view of a WDT model.

multiple disease from sign or symptom of a real-twin employing a dynamic testing process. The input-process-output view of the WDT model is described below:

- **Source of input:** Fig. 3.1 shows that the *Real-Twin and its connected environment* is the key source of input of the model. For example, A person connected to his social media accounts, phone, electronic health records, wearable devices, sensors, and navigators are the source of input of the proposed model.

- **Process:** The WDT model operates on three processes:

    1. *Data Acquisition*: This process collects various health data including, clinical data (e.g., ultrasonography), behavioral data (e.g., mood), physiological data (e.g., bpm), lifestyle data (e.g., step count), environmental data (e.g., location) from the IoT connected environment.

    2. *AI Inference*: AI Inference is the core process of the model. It works as the brain of a WDT model.

        (a) Dynamic MLA Selection process receives a set of dataset, goal, and MLA and outputs matching pairs of Disease dataset and MLA, suitable to the ML prediction.

        (b) Dynamic Testing process predicts the risks of different diseases, using the dynamically selected MLA for those diseases.

    3. *Multimodal Interaction*: This process enables the communication between the Real-Twin and its Digital Twin through Wearable User Interface (e.g., smartphone). For example, a Digital-Twin notifies his Real-Twin that he needs to walk for one hour by displaying text or playing audio from the smartwatch. The rel twin touches his smartwatch screen and visualizes the data summary of his activities and health risks. Here, touch, sight, audio is enabling multi-modal interaction.

- **Output:** The output of the WDT model is a Digital Twin providing that provides health data summary, prediction and recommendation.

  In this research our main contribution is to design the frameworks for Dynamic MLA selection. Another contribution is to propose a method for Dynamic Testing for disease/disease risk Prediction. In the following sections we detail the contributions one by one.

3.2 Dynamic MLA Selection

This section presents the core contribution of our thesis. The proposed framework and algorithms in this section are published in our work [43]. The purpose of the framework is to select suitable MLA to build classifiers for multiple disease/disease risk prediction.

3.2.1 Proposed Framework

The Dynamic MLA Selection framework is a part of the AI inference block of the Fig. 3.1. The key underlying idea of the proposed method is to integrate and match the knowledge of Datasets (D), Algorithms (A), and Goals (G) dynamically to select a proper MLA to process a particular dataset for a specific goal. Fig. 3.2 illustrates the proposed framework for Dynamic MLA Selection. The framework incorporates several functional units: extractDataKnowledge, extactGoalKnowledge, extractAlgoKnowledge, and matchKnowledge. The methodology followed by the proposed approach is discussed below.

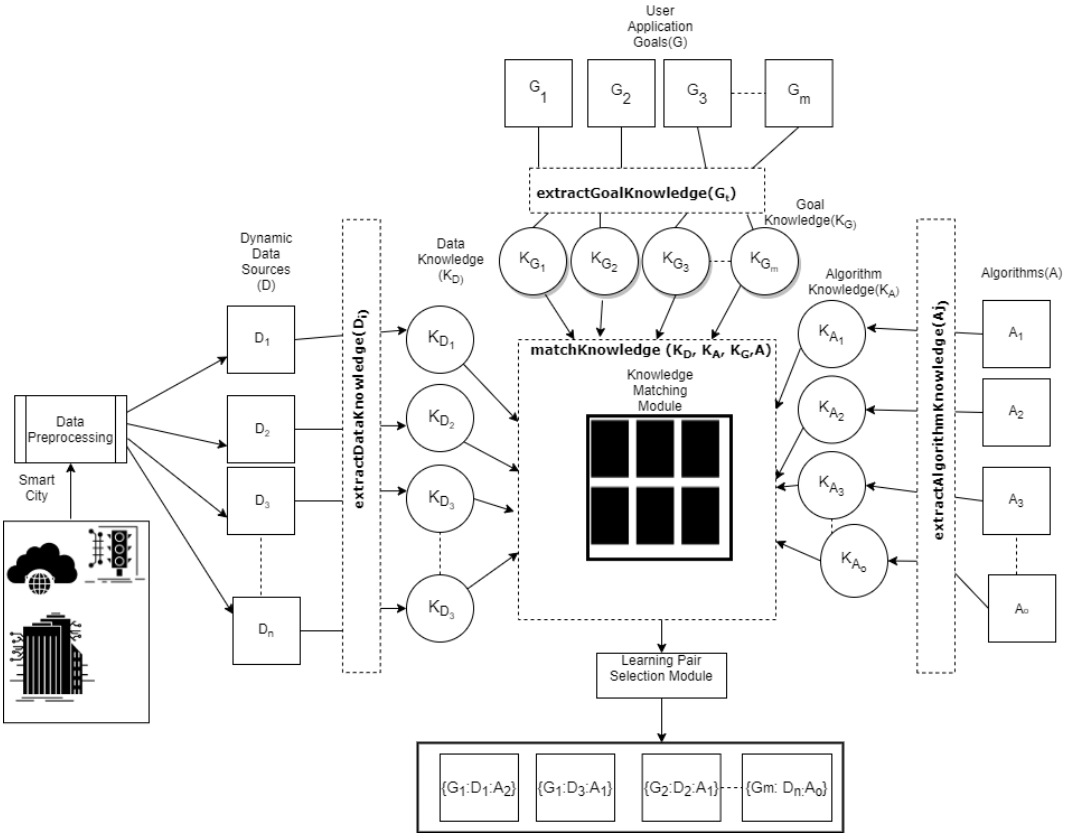


Figure 3.2: Proposed dynamic MLA selection framework for WDT.

3.2.2 Data Pre-processing

Our framework is designed to accept heterogeneous disease datasets and aims to pick any MLA suitable for predicting the disease. In case of multiple diagnosis, data become heterogeneous and contains a empty fields. Therefore, the data-prepossessing is designed with common steps for pre-processing each dataset. Fig. 3.3, illustrates a flow of multi-stage data pre-processing phase to clean raw data.

1. Attribute dropping: Usually, raw data contains multiple unnecessary attributes that

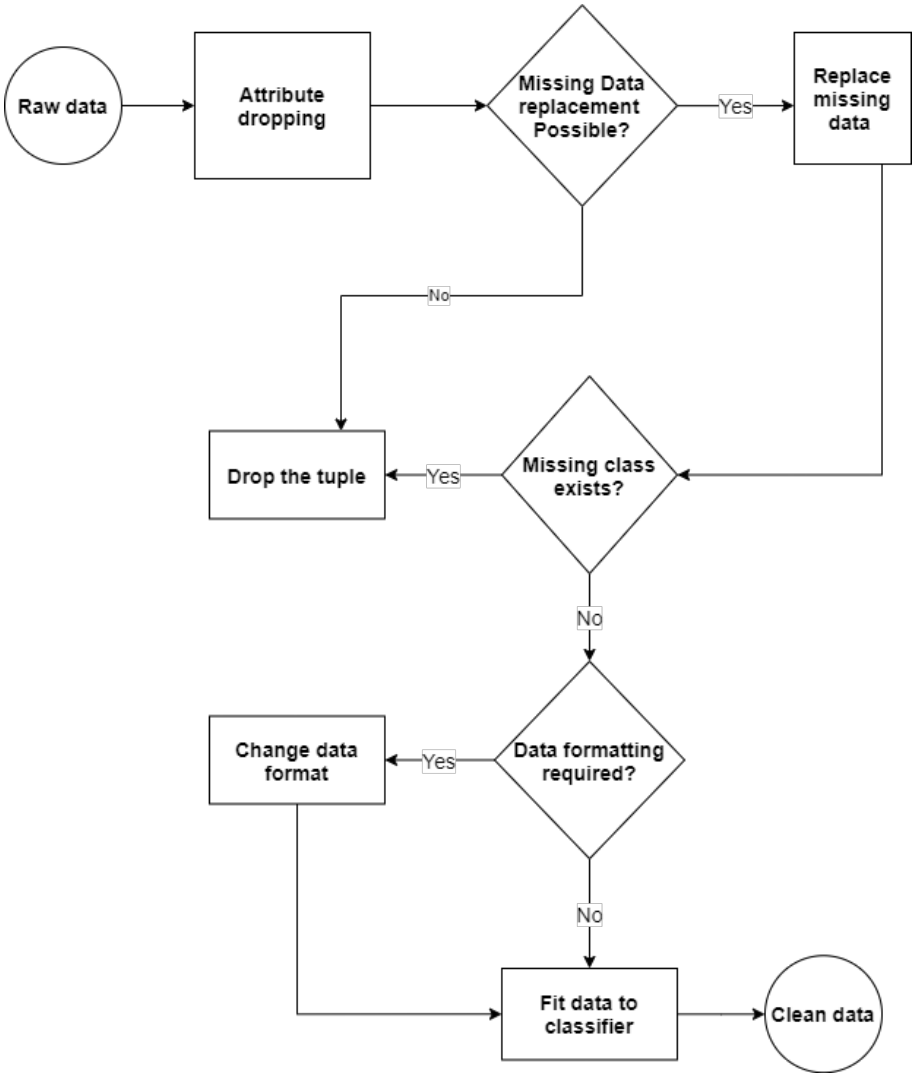


Figure 3.3: Flowchart of data pre-processing steps.

can be ignored during classification. At the initial stage of data pre-processing, these attributes are dropped. It cleans the data and reduces the used memory. For real-time processing memory-efficient approach makes the system suitable for deploying in the cloud environment.

For example, three attributes- city, country, and postal code-represent a patient's address in the Telemedicine healthcare record. Based on the application need, only the postal code can be kept, and the remaining two can be dropped.

2. Missing value handling: We propose missing value handling in two stages. The first stage is missing attribute value handling, and the second one is missing class handling.

   First we check if the missing data is replaceable or not. For example, if the field of disease type is empty, we can replace the value with *unknown*. If the data is not replaceable we drop the tuple.

   Next, we check the missing class. If the class labelling is missing for a tuple, we follow the ignore the tuple approach of data mining and drop the value.

3. Data normalization: We propose a two-stage data normalization process.

   In the first stage, the data value is transformed into a suitable format. For example, if the dataset contains ages between 1-100 years old, these data are categorized into different age groups. Again, gender attributes can be represented using boolean values.

   The second stage applies the Standard Scaler normalization or default classifier fit method to fit the dataset for the classification algorithm. This stage converts string and other data types into a numeric value. For example, seven cities are represented using seven different numbers. Such normalization fits the dataset to a muti-range ML classifier.

3.2.3 Extraction of Dataset Knowledge

The knowledge properties of any dataset are represented as K_D. The process of knowledge extraction from a dataset is given in Algorithm 3.1.

Algorithm 3.1 *extractDataKonwledge* (D_i)

Input: A realtime dataset D_i
Output: A list K_{D_i} containing knowledge of a dataset
Initialize an empty list K_{D_i}
`/*` $getDataType, ..., getDataKnowledge_x$ `are` s `number of abstract knowledge retrieval functions. */`
$K_1 = getDataType(D_i)$;
$K_3 = getLinearity(D_i)$;
$K_2 = getDataContext(D_i)$;
$K_4 = getLocation(D_i)$;
.....
$K_s = getDataKnowledge_x(D_i)$;
$K_{D_i} = add(K_1, K_2, K_3, K_4, ...K_s)$;
`/* add() intregrates the knwoledge attibutes */`
return (K_{D_i});

The specific knowledge properties considered are as followings.

- Data Type: This property refers to the type of data of a dataset. For instances, binary, nominal, numerical, ordinal and mixed .
- Linearity: Linearity represents whether a dataset is linear or non-linear.
- Context: This refers to the context of the dataset. For example, diabetes, thyroid, parking, activity.
- Dataset Location: This property carries information about the source of the dataset. For example, Ottawa, Montreal or any other cities.

3.2.4 Extraction of Goal Knowledge

In this work a goal is defined as a class having multiple attributes, which are collectively considered as goal knowledge K_G. The process of goal knowledge extraction is given in Algorithm 3.2.

Algorithm 3.2 *extractGoalKonwledge* (G_t)

Input: A particular goal G_t

/* G_t is defined as a class, attributes of which will be defined by the system developers, while the values of those attributes are given from the application interface. */

Output: A list K_{G_t} containing knowledge of the goal

Initialize an empty list K_{G_t}

/* $getGoalProcess, ..., getGoalKnowledge_y$ are u number of abstract knowledge retrieval functions. */

K_1= $getGoalProcess(G_t.goalName)$;
K_2= $getGoalDataType(G_t.goalDomainType)$;
K_3= $getGoalModelType(G_t.goalOutputType)$;
K_4= $getGoalTarget(G_t.goalContext)$;
K_5= $getGoalLocation(G_t.goalCoverage)$;
.....
K_u= $getGoalKnowledge_y(G_t.newKnowledge)$;

$K_{G_t} = add(K_1, K_2, ..., K_u)$;
return (K_{G_t});

Specific goal knowledge are in the following.

- *goalName*: The title of the goal (e.g. discovery or verification) represents knowledge about the objective of the Ml process (e.g. prediction or hypothesis).

- *goalDomainType*: It provides knowledge about data type of target dataset (nominal, ordinal, binary, mixed) to be mined in response to the goal.

- *goalOutputType*: The output type of the ML process (e.g. class, group, number, pattern) as expected by the goal that represents knowledge about model (e.g. classification, regression, clustering, hypothesis testing).

- *goalContext*: The context of the goal (diabetes, thyroid, parking, activity) that represents knowledge about target data domain of a dataset.

3.2.5 Extraction of Algorithm Knowledge

The knowledge of MLA is considered as K_A. It is extracted using Algorithm 3.3.

Algorithm 3.3 *extractAlgoKonwledge* (A_j)

Input: A DM algorithm A_j
Output: A list K_{A_j} to contain DM algorithm knowledge
Initialize an empty list K_{A_j}
`/*` $getSensitivity, ..., getAlgoKnowledge_z$ `are` v `number of abstract knowledge retrieval functions. */`
$K_1 = getSensitivity(A_j)$;
$K_3 = getExpectedDataType(A_j)$;
$K_4 = getExpectedOutputType(A_j)$;
$K_2 = getExpectedProcess(A_j)$;
.....
$K_v = getAlgoKnowledge_z(A_j)$;

$K_{A_j} = add(K_1, K_2, K_3, K_4, ...K_v)$;
return (K_{A_j});

Specific algorithm knowledge attributes are:

- Sensitivity: Represents the limitation information of a MLA. In particular, whether the algorithm can accept null value, non-linear data, etc. or not.

- Expected data type: Information of data types that a MLA can handle. For example, binary, nominal.
- Expected output type: This property refers to output type of a MLA. For example, class, group, and pattern.
- Expected process: Refers to the category of the MLA based on data mining task. For instances, clustering, classification, and regression.

Here, we highlight that the required knowledge properties are changeable in the WDT scenario due to the heterogeneity of devices and applications. Thus, Algorithm 3.1– Algorithm 3.3 include abstract functions to extract new knowledge, which in turn utilize several implemented functions in various machine learning libraries such NumPy [66].

3.2.6 Matching Knowledge

Matching among the data knowledge, goal knowledge and MLA knowledge are performed using Algorithm 3.4 to find similarity of knowledge based on the concept of similarity checking approach introduced in [68].

The Fig. 3.4 explains the walkthrough of the matching knowledge block in Fig. 3.2 approach as the following.

- Let there are several datasets like, D_1 = Diabetes Dataset D_2 = Mantal Stress Dataset, D_3 = COVID-19 Dataset available in the Diagnosis database connected to a WDT framework.The proposed model uses Algorithm 3.1 to obtain the knowledge of D_1, D_2, and D_3 as follows.

Algorithm 3.4 *matchKnowledge (K_D, K_A, K_G, D, A, G)*

Input: Knowledge of: Datasets(K_D), Goals(K_G), DM Algorithms(K_A); list: goals(G), datasets(D), DM algorithms(A)
Output: Return a set T_{match} with matched set of tuples.

```
for each G_t in G do
    maxSimD, maxSimA = –Infinity;/* initialize maxSimD with a least value                          */
    setD, setA ={};/* initialize empty set to hold matched datasets                                */
    for each D_i in D do
        S_{G_t.D_i} = Sim(K_{G_t}, K_{D_i}); /* similarity betw.  knowledge of Goal/Dataset          */
        If(S_{G_t.D_i} ¿ maxSimD)
            { setD.clear(); /* clear all previous elements from setD                               */
        setD.add(D_i); /* store datasets with highest similarity score                            */
        maxSimD =S_{G_t.D_i};
          }
          Else If(S_{G_t.D_i} == maxSimD)
          { setD.add(D_i); }

    for each A_j in A do
        S_{G_t.A_j} = Sim(K_{G_t}, K_{A_j}) /* similarity betw.  knowledge of Goal/DM Algorithm     */
        If(S_{G_t.A_j} ¿ maxSimA)
            { setA.clear();/* clear all previous elements from setD                               */
        setA.add(A_j); /* store DM algorithms with highest similarity score                       */
        maxSimA =S_{G_t.A_j};

        }
          Else If(S_{G_t.A_j} == maxSimA)
          { setA.add(A_j); }

    /* Now, a loop will run till e_1=|setD|, which is the cardinality of setD                     */
    for each D_{e_1} in setD do
        maxSimMerge =–Infinity;/* initialize maxSimMerge with a least value                       */
        setD_A={};/* initialize empty set to hold matched dataset and DM algorithm               */
        /* Now, a loop will run till e_2=|setA|, which is the cardinality of setA                 */
        for each A_{e_2} in setA do
            /* find similarity score betw.  knowledge of items in setD and setA.                  */
            S_{D_{e_1}, A_{e_2}} = Sim(K_{D_{e_1}}, K_{A_{e_2}});
                If(S_{D_{e_1}.A_{e_2}} ¿ maxSimMerge)
                { setD_A.clear();/* clear all previous elements from setD_A                       */
            setD_A.add({D_{e_1}, A_{e_2}}); /* store datasets/DM algorithms pair with highest similarity score */
            maxSimMerge =S_{D_{e_1}.A_{e_2}};
              }
        T_match= T_match ⋃ {G_t × setD_A};
          /* T_match contains selected tuples of Datasets, DM Algorithms and Goal G_t.           */
return T_match;
```

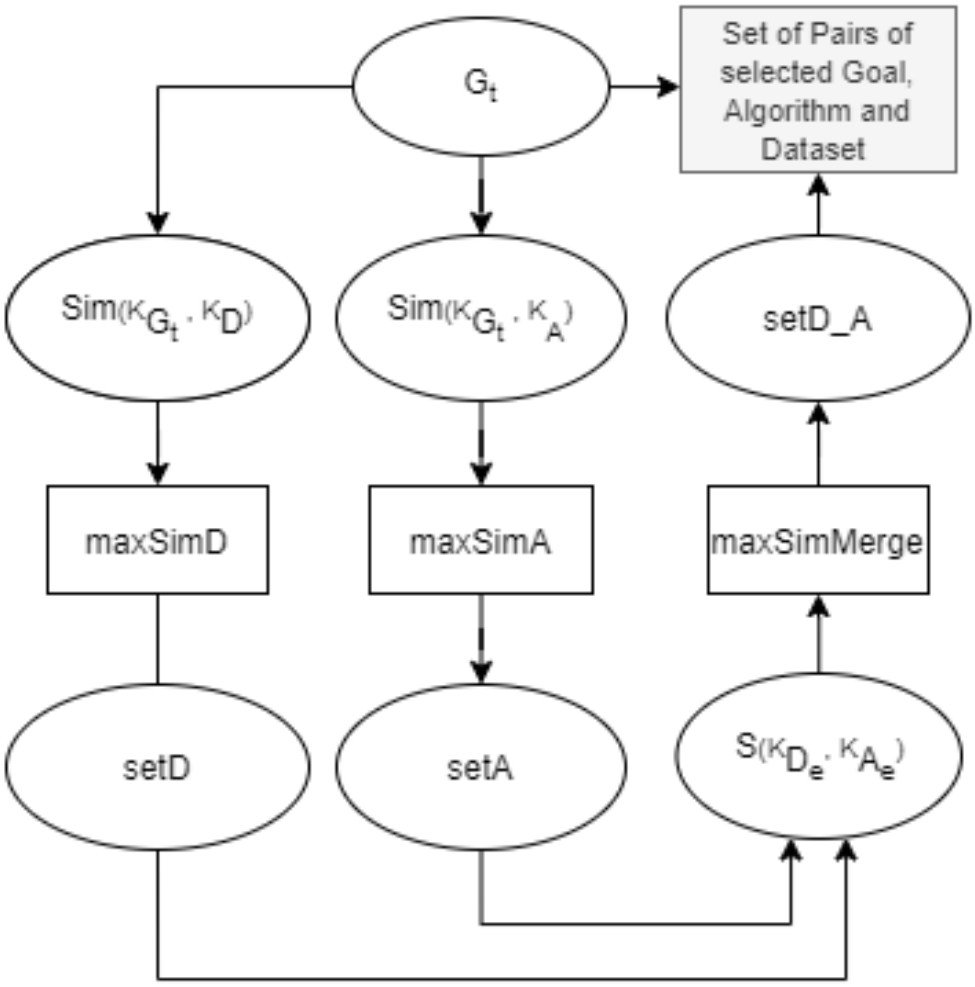


Figure 3.4: Workflow of matching knowledge block.

– K_{D_1} = {K_1=mixed, K_2= classification, K_3=diabetes risk, K_4= ottawa,...}

– K_{D_2} = {K_1=mixed, K_2= classification, K_3= mental stress, K_4= montreal,...}

– K_{D_3} = {K_1=nominal, K_2= classification, K_3= covid risk, K_4= toronto,...}

- Set of Goals G, where each goal is defined as a class. In particular, a goal refer to general tasks performed by MLA. For this example, we consider three goals: G_1 = diabetes risk prediction and G_2 = mental stress prediction and G_3 = COVID-19 risk prediction. Algorithm 3.2 is used to extract knowledge sets K_{G_1}, K_{G_2}, and K_{G_3} for G_1, G_2, G_3,and respectively as follows.
  - K_{G_1} ={K_1=prediction, K_2=mixed, K_3=classification, K_4 = diabetes risk,...}
  - K_{G_2} ={K_1=prediction, K_2=nominal, K_3=classification, K_4 = covid risk,...}
  - K_{G_3} ={K_1=prediction, K_2=mixed, K_3=classification, K_4 = mental stress,...}
- Set of MLA A, where an element of A can be any MLA. Let us consider several of such algorithms as, A_1= NB, A_2=RF, A_3 = HC, A_4=FC in list A. Now, Algorithm 3.3 is used to extract knowledge sets K_{A_1}, K_{A_2}, K_{A_3}, K_{A_4} of each of these MLA as in the following.
  - K_{A_1} = {K_1=classification, K_2=mixed, K_3=class,...}
  - K_{A_2} = {K_1=classification, K_2=nominal, K_3=class,...}
  - K_{A_3} = {K_1=clustering, K_2=mixed, K_3=group,...}
  - K_{A_4} = {K_1=clustering, K_2=mixed, K_3=pattern,...}

- Algorithm 3.5 is used to accumulate knowledge sets K_D, K_G and K_A. It can be observed that there are semantic matches among the values represented by K_D, K_G and K_A, which are accomplished by Algorithm 3.4. First for each goal, a set of potentially matching datasets ($setD$) and MLA ($setA$) are selected. Then a final matching is conducted among these two sets to pick the matching MLA for each selected dataset. Let, $Sim(K_{G_1}, K_{D_1})$, $Sim(K_{G_2}, K_{D_3})$, $Sim(K_{G_3}, K_{D_2})$ provides highest similarity, then $setD = \{D_1, D_2, D_3\}$ for G_1, G_3, G_2 respectively. In the same manner, assume A_1 is selected for D_1, D_2 and , A_2 is selected for D_3.

  So finally, for diabetes risk prediction and mental stress prediction, RF will be selected, and for covid risk prediction, NB will be selected.

3.2.7 Proposed DMLA Algorithm

This section presents the main Algorithm designed to implement the Dynamic MLA Selection process in the WDT model. The Algorithm 3.6, takes three inputs- a) set of disease datasets like- dataset for diabetes prediction, stroke risk prediction, COVID-19 prediction, mental stress prediction, b) Set of Goals like- classification/regression/clustering, and c) set of machine learning algorithms like- Random Forest(RF), Naive Bayes(NB), Decision Tree(DT), Neural Network (NN), etc. Then the Algorithm 3.1, Algorithm 3.3, and Algorithm 3.2, extracts knowledge of D, A and G respectively. After that, the Algorithm 3.4 finds the matching pair of MLA and Training Dataset based on the highest similarity between the extracted knowledge for the training prediction model of the WDT framework. The train function in Algorithm 3.6, trains the prediction model for each disease available in the Diagnosis Database.

Algorithm 3.5 *DMLA (D, G, A)*

Input: Set of Datasets D, Set of Goals G and Set of MLA A (supervised or unsupervised.)
Output: For each element in G for any element of D a potential element from A will be selected to obtain data mining result.

/* D, G and A will be updated at each execution time. */
for *each* D_i *in list* D **do**
    $K_{D_i} = extractDataKnowledge(D_i)$;
    $K_D = K_D \bigcup K_{D_i}$;

for *each* G_t *in list* G **do**
    $K_{G_t} = extractGoalKnowledge(G_t)$;
    $K_G = K_G \bigcup K_{G_t}$;

for *each* A_j *in list* A **do**
    $K_{A_j} = extractAlgoKnowledge(A_j)$;
    $K_A = K_A \bigcup K_{A_j}$;

T_{pairs}= $matchKnowledge(K_D, K_G, K_A, D, A, G)$;
    /* where, T_{pairs} stores the output of $matchKnowledge$ */
return $train(T_{pairs})$;
    /* $train$ is general ML training process with the selected tuple T_{pairs} */

3.2.8 Complexity Analysis

The worst-case computational cost of the proposed framework has been detailed here. The cost approximation is divided into two phases: knowledge retrieval phase and knowledge matching phase. In the Knowledge retrieval phase, $extractDataKnowledge()$, $extractGoalKnowledge()$, and $extractAlgoKnowledge()$ in Algorithm 3.5 are evaluated, whereas for the Knowledge matching phase, $matchKnowledge()$ algorithm is evaluated.

Complexity in Knowledge Retrieval

Taking a closer look at Algorithm 3.5 it can be observed that the upper bound of first for loop is n. For each call the $extractDataKnowledge()$ algorithm is executed to extract s knowledge items for each dataset. In the dataset, different data types may exist and so the algorithm needs to perform row-wise and column-wise search. Hence, in brute-Force case, the complexity of obtaining knowledge of n number of datasets, K_D, can be approximated as $O(s.n.1) \cong O(n^2)$ when considering $s = n$.

Again, for second loop in Algorithm 3.5, u number of goal knowledge items for each goal is extracted using the $extractGoalKnowledge()$ algorithm. So for a total m number of goals, the complexity of obtaining overall goal knowledge, K_G, can be approximated as $O(u.m.1)$ when each of K_1, K_2,...,K_u takes 1 CPU cycle. Now, considering $u = m$, the worst case complexity of goal knowledge extraction becomes $O(m^2)$.

Similarly, the complexity to extract the knowledge of data mining algorithms, K_A, is $O(v.o.1) = O(o^2)$ when $v = o$. Now, if n, m, and o are close proximity values, the overall complexity of knowledge retrieval phase approximates to $O(n^2) + O(m^2) + O(o^2) \cong O(n^2)$.

Complexity in Knowledge Matching

After the knowledge retrieval process, the knowledge matching process starts in Algorithm 3.5 by executing the $matchKnowledge()$ function, where the outer loop runs for m times. Then two matching operations are performed: matching between goal knowledge and dataset knowledge to get $setD$; matching between goal knowledge and algorithm knowledge to get $setA$. If we use a better search algorithm for matching, then the complexity to obtain $setD$ is $O(nlog\ n)$ for n number of datasets. Likewise, the complexity to obtain $setA$ is $O(olog\ o)$ for o number of algorithms. Together, for m number of goals, the complexity to obtain $setD$ and $setA$ becomes $O(mnlog\ n) + O(molog\ o)$.

Now, the remaining loops in Algorithm 3.5 are carried out to obtain $setD\_A$ by matching the knowledge between each of the e_1 number of selected datasets in $setD$ and e_2 number of selected algorithms in $setA$ for m goals; the complexity of which is approximated to $O(m.e_1.e_2log\ e_2)$.

Thus, the overall complexity of knowledge matching phase can be approximated to $O(mnlog\ n) + O(molog\ o) + O(m.e_1.e_2log\ e_2)$. As e_1 and e_2 are very small value, the portion $O(m.e_1.e_2log\ e_2)$ would be negligible. Now assuming, m, n, and o are close proximity values, the overall knowledge matching complexity results in $O(n^2log\ n)$.

3.2.9 Competitive Advantage

- The existing AI inference still needs manual analysis to select the proper MLA, which is costly and time-consuming. Whereas the proposed AI inference can do this dynamically using Algorithm 3.4.
- The current work is often limited to using a single type of prediction for weel-being. In

contrast the proposed dynamic framework is designed for numerous disease prediction fro health and well-being.

3.3 Dynamic Testing for Prediction

This section presents another contribution of our research for predicting several disease from a single WDT model. The contribution is published in [33].

3.3.1 Proposed Methods

In this section, we present the proposed Dynamic Testing process for prediction detailing the algorithms and method.

To design this method, we considered that the test data for disease prediction come from both the sensor data (from wearable network), and non-sensory data(from user's input). This method builds various disease classifiers using the training dataset and classifiers selected by the dynamically selected machine learning algorithm, discussed in the previous section. To map the sensory data as disease symptom status, the proposed testing method follows several steps as demonstrated in figure 3.5. The data flow is marked with numbers to demonstrate a closed-loop Machine Learning model. The detail of system processes appears in the following sections.

3.3.2 Detection of Health Monitoring Sensors

The wearable network includes sensors of diverse genres targeting different goals, such as health monitoring, disorder prediction, safety monitoring, home rehabilitation, activity

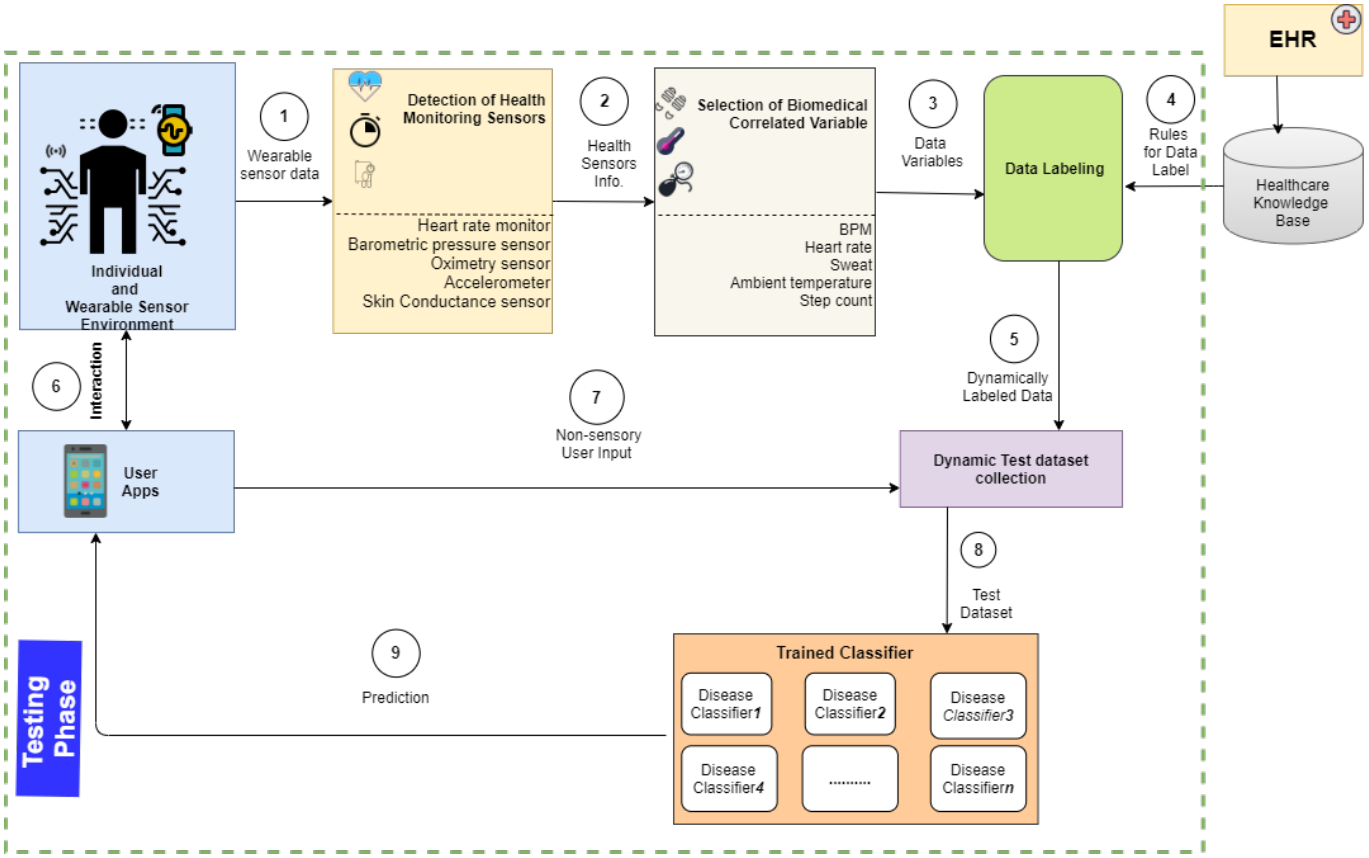


Figure 3.5: Flow of prediction using dynamic testing method.

monitoring, treatment assessment, etc. Information about these sensors includes sensor type, id, record type, manufacturer, and service information. This information can be extracted using Python command lines or dedicated tools. To detect whether the sensor is a health monitoring sensor or not, information about each wearable sensor in a specific network will be checked. If the model includes n wearable sensors then $W = w_1, w_2, w_3, ..., w_n$ is a list of sensors in an environment. For each wearable sensor w, and information list is extracted as $info\_list = i_1, i_2, .., i_m$. An instantiation of $info\_list$ can be seen in Figure 3.6.

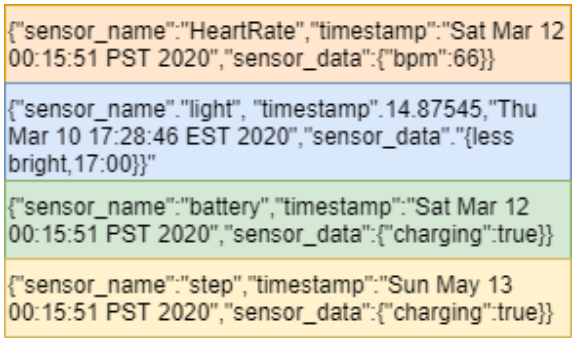


Figure 3.6: Example of information list of sensors.

3.3.3 Selection of Biomedical Correlated Data

The reading of the sensors will be used for selecting correlated biomedical variables. In particular, the correlation between sensor readings and the biomedical variable is assessed. Possible examples of correlated biomedical variables are- beat per minute (BPM), sweat, step count, etc. To conduct this step, a pre-defined list containing the biomedical variables is checked. For example, if the list contains {bpm, step count, sweat, sleeping time} and the reading from a wearable sensor w is {bpm:60} then the variable bpm will be selected as a correlated biomedical variable.

3.3.4 Healthcare Knowledge Base

A Healthcare knowledge Base is includes medical practitioners approved EHRs such as prescriptions, clinical test reports etc. It contains health information to lable data. For example, risk, symptom and disease name, medical rules to determine attribute value, and other information. The use of an healthcare knowledge base can accelerate the performance of the classification system. The proposed system uses the knowledge base in both the training and the testing phase. In the training phase, the verified dataset from the knowledge base has been used to train the classifier with several ML classification algorithms. In contrast, in the testing phase, the knowledge base has been used to assign rules and data labels and to extract features for predicting diseases from sensor data.

3.3.5 Data Labeling

The data labelling is performed by applying rules from the healthcare knowledge base on the sensor data variables. The knowledge base includes the features of diseases and data from public healthcare records (marked as 3 and 4 on Figure 3.5). These records provide the selection of epidemiological factors by the users through the user interface in real-time. For example, the knowledge base will include epidemiological factors (e.g. age, sudden weight-loss, and palpitation) with rules. These epidemiological factors are the filtered features. The feature selection contains healthcare practitioner's identification. The Dynamic Testing has been described before. So, now we will move to the evaluation process of the ML process.

3.3.6 Training Classifiers

The labelled test data and user test data both will be received by the Trained disease classifiers.

The data source of training is the EHRs. Those EHR contains patient data collected from a direct pre-screening questionnaire or via other means, as well as approved and overseen by the healthcare practitioners. Those refined data are stored in the Database of Disease, serving as the knowledge base. The practitioners-approved data have the potential to increase the level of acceptance for risk prediction of diseases. Multiple electronic healthcare records (EHRs) train the model to predict multiple diseases through a single system. The suitable MLAs for the selected EHRs are provided by the MLA selection method described in the previous section 3.2. The WDT framework builds a classifier for each disease with these pairs of MLA and EHR.

The prediction from the classifiers are passed to the user again through an user interface. The user and user application will communicate with each other using multimodal interactions as we have seen in the high level view of the WDT model.

3.3.7 Dynamic Testing Algorithm

Algorithm 3.6 present an algorithm for dynamic testing. The defined class TESTING has two attributes, $feature\_name$ to contain a feature name of the training dataset and $feature\_value$ to contain the value of a particular feature. The other defined class RULE has five attributes, which represent a rule in the knowledge base. An example of instantiation of an object of this class is: r=RULE($match\_param\_name$= bpm, $operator = >$, $match\_param\_value = 80$, $decision\_param\_name$ = Irritability, $decision\_param\_value = 1$).

The algorithm takes the health sensor information list as an input and initializes list T for output. In Python programming, a list is used as an array. Therefore, we have used the term list and T as the list of TESTING instances. The contents of the information of sensors are presented in Figure 3.6. The sensor data of each sensor is then split into two parts with a delimiter. For example, sensor data = 'bpm:60' is split as bpm as the correlated biomedical variable, and 60 is its value.

Then the correlated biomedical variable is checked with the *match_param_name* in the list of type RULE by the *Compare* function. If the rule is satisfied then *decision_param_name* is stored as the feature name of a TESTING instance and *decision_param_value* is stored as the feature value. Finally, the list T is returned as the dynamically created training data record.

Algorithm 3.6 Algorithm for Dynamic Testing.

```
class TESTING
feature_name
feature_value
end class

class Rule
match_param_name
operator
match_param_value
decision_param_name
decision_param_value
end class

Initialize TESTING list T;
Input list health_sensor_info, RULE list rule;

for i=0 to health_sensor_info.length() do
    \ * Split sensor data which is in "data name: value" format.
    \ * find() is a Python function-returns index of a character in string.
    \ * ':' is the delimiter, splitting value before and after ':' .

    sensor_data = health_sensor_info[i].sensor_data;
    bio_cor_var = sensor_data[: sensor_data.find(':')];
    bio_cor_val = Int(sensor_data[sensor_data.find(':') + 1 :]);

    for j=0 to rule.length() do
        if rule[j].match_param_name == bio_cor_var then
            \ *Check the sensor data with knowledge base rule.
            Compare(bio_cor_val, rule[j].operator, rule[j].match_param_value)
            if TRUE then
                T[i].feature_name = rule[j].decision_param_name;
                T[i].feature_value = rule[j].decision_param_val;
Return T;
```

3.3.8 Competitive Advantage

- The existing WDT framework applies several MLA, checks the performance and finds the one with the best accuracy manually. In contrast, our proposed framework conducts each step for an ML prediction dynamically.
- The use of a medical practitioner's verified training dataset in the framework has reduced the massive pre-processing stage of ML.
- The multistage conversion of heterogeneous IoT sensor data into a meaningful dataset opens a new door to predict disease risks from the low-level sensor data in WDT.

Chapter 4

Dynamic MLA Selection Functions

We implemented a dynamic pre-processing function and a dynamic MLA selection function that fits multiple classifiers and detests. The following sections present the required Tools and Technology, Data Collection and function description.

4.1 Tools and Technology

In this work, we used several tools & technologies like libraries, simulators, platforms. This section describes the data analysis tool, Programming language, cloud environment, and essential libraries used to implement.

1. **Data Analysis Tool** For analyzing the characteristics of the various dataset and to extract the MLA knowledge, we used the *Waikato Environment for Knowledge(WEKA)* Analysis [37].

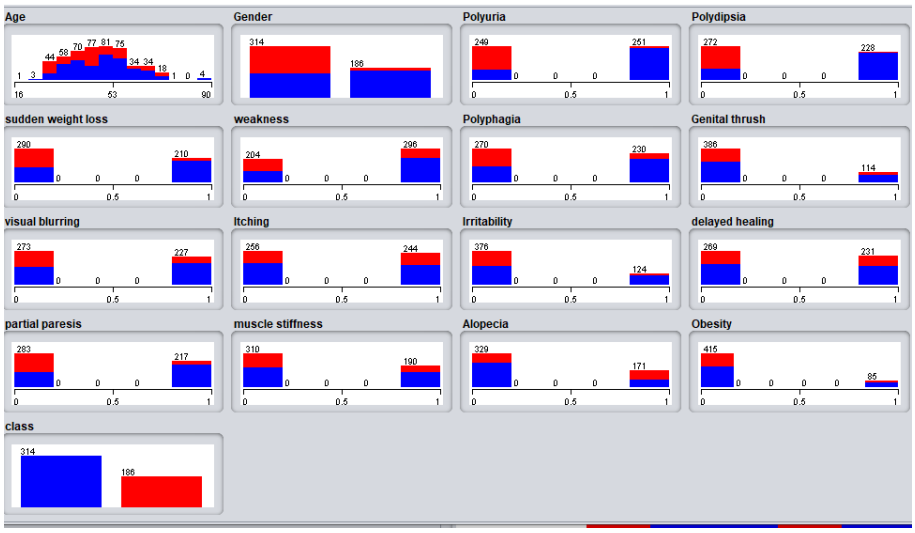


Figure 4.1: Attribute distribution in the diabetes risk prediction dataset using WEKA.

2. **Programming Language** The underlying programming language of the implementation is Python [47].The reason behind choosing Python was the plethora of advantages that make it particularly ideal for machine learning and deep learning applications.

3. **Cloud Environment** We implemented the functions on Cloud Jupyter Notebook environment Colab [63].The reason behind this selection is that Google Colab supports Python to working with several libraries, including PyTorch, Keras, TensorFlow, and OpenCV.

4. **Important Libraries** We employed several libraries for machine learning, and data visualization. Some notable of those are numpy, pandas, sklearn, seaborn, matplotlib [66].

5. **Simulator** The testing and verification of the framework on sensor-only data were complex and tedious without a simulator. Therefore we employed Cooja for creating

a simulation network of wearable sensors [17]. The reason behind selecting Cooja as the simulator is that. It can simulate heterogeneous nodes in the same network. Moreover, it gives a lot of details of the node's hardware.

4.2 Data Collection

In this section, we describe the data collection procedure for the implementation. We have categorized the data collection into two categories- a) Public Dataset Collection, b) Private Data Collection. The details are provided following.

4.2.1 Public Data

We collected publicly available data for Diabetes risk prediction, Liver disease prediction, Diabetes Prediction, and Stroke prediction. Some of these data were available on UCI Machine Learning Repository, and some on Kaggle. To collect data from UCI Machine Learning Repository, we used the open-source API known as UCI-ML-API. We cloned the git repository, executed the main python file to collect our desired data, and stored the data into Comma Separated Value (CSV) format. To collect other data from GitHub or Kaggle website, we used Python Request Module with REST API.

4.2.2 Private Data

We collected the data for Mental Stress Prediction, COVID risk prediction, and Thyroid Prediction in private mode from the hospital's medical record and from the individuals. Those data were solely accessed by us for the evaluation purpose of this research. All data

are collected with institution's consent and ethical approval. The Mental stress data were collected from a controlled setup, whereas the other two datasets were collected from a Telehealth service and hospitals in Bangladesh.

The data for Mental stress detection were collected from 27 volunteer subjects between 18-50 years old, who were monitored over one week from 20th June, 2020 to 30th June, 2020. Multimodal data were sensed using the sensor, app, and IoT technologies available on Redmi 8 Pro Android smartphone, Xiaomi Mi band 5 Pro version. These devices produced the data to model the digital twin of a stress survivor. The collected data include behavioural and psychological senses like heart rate, sleep time, activity time, step count; social senses like outgoing calls, incoming calls, messages, call duration, chatting time, online activity time; self-reported senses like age, appetite, preferences on social media activity, reaction to COVID-19 news and reports.

An original dataset from the Telemedicine Service of Bangladesh was collected. This dataset contains the record of one-month data from 1/04/2020 to 30/04/2020, while coronavirus started to spread in Bangladesh significantly. The record of the patient's call, system's input, and doctor's evaluation has been collected in the CSV data format through a secure API of the Telehealth service provider. Employee credential was created and used to access the patient's call record. The data collection for thyroid prediction from a clinic in Bangladesh follows the same approach.

4.3 Implementation of Function

We implemented several functions to implement the disease classifiers. The decription of the functions are followings.

4.3.1 MLA Selection Function

We implemented the Dynamic MLA selection function using python programming language. We checked the file extension of the input file using string matching function. For example whether $file_{extension}$ == csv or .jpg or .png, or .txt or unknown was checked to accept data stored in multiple format.

In the next stage, we implemented functions to retrieve the knowledge properties of the dataset, goal and ML Algorithm. The sub-function to extract data knowledge the function is for finding the type of data. Initially, we worked with numerical and nominal data like WEKA. However, the same function can be used for the binary, ordinal, continuous and discrete data.

We implemented more functions similar to this to retrieve context, location, feature_size and other knowledge properties. The data knowledge was possible to retrieve directly from the raw dataset. However, in Algorithm knowledge extraction, we are required to employ a list of MLA knowledge explicitly. For ease of implementation, we used the CSV format of the merged database.

For algorithm knowledge extraction, the CSV file is providing the knowledge properties of the machine learning algorithms. Unlike dataset, the knowledge properties is predefined. The algorithm knowledge was retrieved from the WEKA open-source tool. From Fig. **??**, it can be seen that the associated information about the Naive Bayes Algorithm is provided by the WEKA tool. Particularly, under the tag *Capabilities* the supported classes, attributes, and a Minimum number of instances are provided, which present the goal, data type and size supported by the specific MLA.

To implement the similarity function, we calculated the *cosine-distance* between two knowledge sets according to the Algorithm 3.4 proposed in Chapter 3. We implemented

the similarity function by calculating the similarity between two sets of knowledge. The knowledge were stored using string lists. We used the built-in python Counter() function to find the modalities of the knowledge. This function returned similarity in float format ranging from 0.0 to 100.0.

The goal of our target dataset is classifications. Therefore we implemented the knowledge matching by calculating the similarity between the knowledge of EHRs from and MLAs. The details of these datasets and algorithms are discussed in the next chapter. The matching function returns the matching pair of dataset and algorithm with the similarity score. It can be seen from the main output that pairs are selected at the maximum similarity.

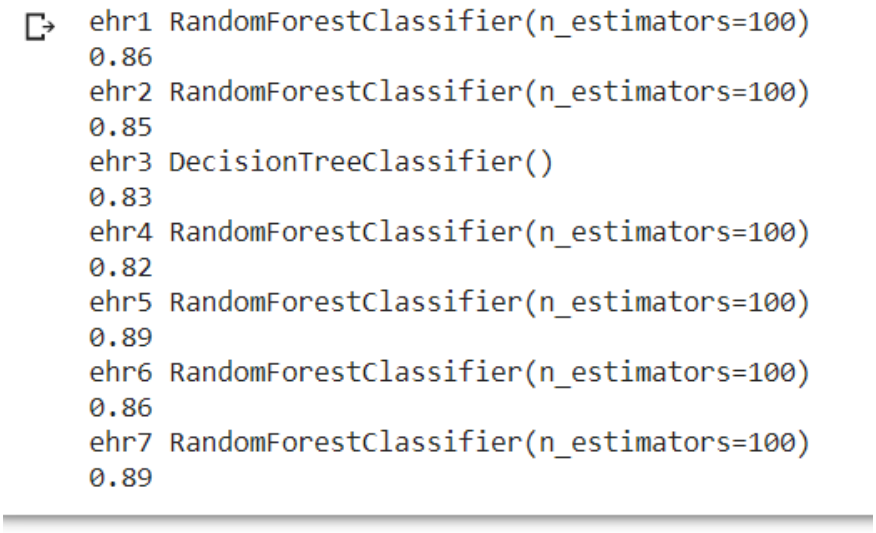


Figure 4.2: Output from DMLA function. The ehr1, ehr2, etc. are datasets, DecisionTreeClassifier(), RandomForestClassifier(n_estimator=100) are selected classifiers for specific datasets.The numeric values represent the similarity score.

4.3.2 Dynamic Pre-processing Function

As the training dataset are EHRs are heterogeneous. We implemented a pre-processing function that fits any training dataset and transforms each column value of a data frame into a standard format. Multiple machine learning classifiers can be trained with our pre-processing function. We used the SKlearn LableEncoder() function to transform each column value of the data frame to fit the algorithms. Also, this pre-processing function replaces the null value with a dummy value.

4.3.3 Dynamic Training Function

We implemented a dynamic training function. The function receives the matching pair of datasets and algorithms from the matching knowledge function described above. The training function then stores the datasets in a list named as selectedEHR list and algorithms in another list named as selectedML list. Then it uses the dynamic pre-processing function to each selected EHR and trains the classifier with that EHR, and its specific MLA.

4.3.4 Dynamic Testing Function

The non-sensory test data, like data from EHR was collected from the 80:20 percentage split. However,the IoT device data for mental stress were mapped to require feature values. These required features are the attributes from the training set used to build mental stress classifiers. In the following, we have described the outcomes of the testing data generation algorithm for the diabetes dataset.

– The heart rate sensor provides bpm record.

- The eating sensor provides information about the number of food intake time.
- The first light sensor is ON if the battery is charged and OFF otherwise.
- The Bluetooth sensors provide information about whether it is connected to the network or not. Another light sensor is on when the Bluetooth is connected.
- The step count sensor provides information about the number of steps completed by a person.
- Drink water sensors count the number of times water drunk by a person.
- The skin rub sensor provides information about the number of times a person rubs the skin.
- The blood pressure (bp) sensor provides information about systolic and diastolic pressure (systolic, diastolic) in mm (Hg). For detecting the mental stress, we collected smartphone, smartwatch data using REST API.
- In practice, the smartwatch and fitness band has embedded sensors to provide more meaningful information. The sensors are often made of basic sensors, which collect regular sensing data like pressure, temperature, movement, location, etc., and transform these data into more meaningful information like drink water time, food intake, sleep time, etc.

The payloads update the information of the sensor. There are multiple wearable sensors in the network for different purposes. Unlike basic sensors like gyroscopes or accelerometers, the modified sensors can provide more meaningful information. Table 4.1 provides detailed information of the wearable sensors simulated for this work. The sensor_name represents

the title of the sensors, which reflects the purpose of the sensors. The timestamps represent the time of capturing the sensor records. The sensor_data provides information about the reading of the sensors. Further elaboration of the sensors in Table 4.1 can be given as the following.

The sensor_name is matched with a pre-defined list of health sensors, and when a match is found, the flag is set to 1 in the program to represent it as a health sensor. Based on the information in Table 4.1, it can be seen that there are six sensors found as health sensors in the wearable sensor network as shown in Table 4.2.

4.3.5 Evaluation Function

To implement the evaluation function, first we retrieved the false positive (fp), true positive(tP) and threshold(th) from the scikit-learn python function. Then we computed the training accuracy, testing accuracy, precision and recall. We used the scikit-learn auc() function to get the ROC area value that takes the fp and tp value as input. We calculated all the parameters in percentage. This block of the loop runs for each ML prediction provided by the WDT model. The results from the evaluation are described in Section 5.2.4.

Table 4.1: Information list of sensors in the wearable sensor network.

Sensor ID	**Information List**
aaaa::212:7403:3:303	{"sensor_name":"Heartrate","timestamp":"Fri Mar 11 06:18:24 EST 2020","sensor_data":{"bpm":105}}
aaaa::212:740a:a:a0a	{"sensor_name":"Eating","timestamp":"Fri Mar 11 06:18:24 EST 2020","sensor_data":{"eat_count":10}}
aaaa::212:7402:2:202	{"sensor_name":"Light","timestamp":"Fri Mar 11 06:18:24 EST 2020","sensor_data":{"light_status":1}}
aaaa::212:7404:4:404	{"sensor_name":"Bluetooth","timestamp":"Fri Mar 11 06:18:24 EST 2020","sensor_data":{"btconnection":connected}}
aaaa::212:7405:5:505	{"sensor_name":"Stepcount","timestamp":"Fri Mar 11 06:18:24 EST 2020","sensor_data":{"step_count":2500}}
aaaa::212:7406:6:606	{"sensor_name":"Battery","timestamp":"Fri Mar 11 06:18:24 EST 2020","sensor_data":{"charging":70%}}
aaaa::212:7407:7:707	{"sensor_name":"Drinkwater","timestamp":"Fri Mar 11 06:18:24 EST 2020","sensor_data":{"drink_count":16}}
aaaa::212:7408:8:808	{"sensor_name":"Light","timestamp":"Fri Mar 11 06:18:24 EST 2020","sensor_data":{"light_status":1}}
aaaa::212:7409:9:909	{"sensor_name":"Skinrub","timestamp":"Fri Mar 11 06:18:24 EST 2020","sensor_data":{"skin_rub_count":35}}
aaaa::212:740b:b:b0b	{"sensor_name":"BP","timestamp":"Fri Mar 11 06:18:24 EST 2020","sensor_data":{"bp":90,50}}

Table 4.2: Detected health sensors.

Sensor ID	**Sensor Name**	**Health Sensor? (yes=1, no=0)**
aaaa::212:7403:3:303	{"sensor_name":"Heartrate"}	1
aaaa::212:740a:a:a0a	{"sensor_name":"Eating"}	1
aaaa::212:7402:2:202	{"sensor_name":"Light"}	0
aaaa::212:7404:4:404	{"sensor_name":"Bluetooth}	0
aaaa::212:7405:5:505	{"sensor_name":"Stepcount"}	1
aaaa::212:7406:6:606	{"sensor_name":"Battery"}	0
aaaa::212:7407:7:707	{"sensor_name":"Drinkwater"}	1
aaaa::212:7408:8:808	{"sensor_name":"Light"}	0
aaaa::212:7409:9:909	{"sensor_name":"Skinrub"}	1
aaaa::212:740b:b:b0b	{"sensor_name":"BP"}	1

Table 4.3: Biomedical correlated variable and sample rules from the epidemiological knowledge base to map sensor readings.

Sample sensor reading	**Biomedical correlated variables**	**Rules**	**Decision**
bpm:105	bpm	>100	Irritability=1
eat_count:10	eat_count	>5	Polyphagia=1
step_count:2500	step_count	<2000	Obesity=1
drink_count:16	drink_count	>15	Polydipsia=1
skin_rub_count:35	skin_rub_count	>30	Itching=1
bp:90,50	bp	<90,<60	Weakness=1

Chapter 5

Evaluation

This chapter presents three use cases, experiments, and findings of the proposed framework. We evaluated the performance of our framework from two different perspectives given following.

1. In the first evaluation, we obtained the performance of automatically selected MLAs and compared them with the performance of existing work.
2. In the second one, we verified the performance of the selected MLA and compared it with other existing works.

5.1 Use Cases

This section describes three possible use-cases for our proposed framework. We provide a brief description of the use cases describing the goal to be achieved by the framework.

5.1.1 WDT for Non-Communicable Disease

The NCDs are not transferable and non-contagious [90], although more life-threatening than contagious diseases. According to the World Health Organization (WHO), NCDs are responsible for 71% of all deaths globally. However, the risk factors and determinants of these diseases, which are commonly known as epidemiological factors, are adjustable and controllable [57]. For example, obesity is an epidemiological factor that can cause NCDs like diabetes, stroke, hypertension, and kidney disease [80]. Therefore, the incidence of NCDs can be minimized by controlling these factors. Epidemiological factors of NCDs generally stem from physical inactivity, alcoholic habit, diets, and other conditions. Hence, pre-screening and preventive measures are the keys to respond to NCDs [43]. The value of health transformation, the empowerment of wearable sensors, and ML must be broadly acknowledged in the fight against losses due to NCDs [15]. We considered the following NCDs for this research.

1. Diabetes Risk Prediction (at an early stage)
2. Diabetes Prediction (If it's a disease)
3. Liver Disease Prediction (If it's a disease)
4. Thyroid Risk Prediction (at an early stage)
5. Stroke Risk Prediction (at an early stage)

The flow of NCD prediction by our proposed WDT framework is following.

1. The proposed framework collects the epidemiology of NCDs from datasets. These datasets include information from various sources like- EHRs, smartphones, smartwatches,even from wearable sensors.

2. The datasets are then pre-processed to remove unnecessary attributes, and noisy instances following the multi-stage preprocessing steps in Fig. 3.3

3. Then proper MLA are selected automatically by the framework for each training dataset set.

4. Classifiers are built for each NCD with the algorithm selected by the framework.

Finally, the classifiers predicts the listed disease or risk of disease through the proposed dynamic testing method,as discussed in Section 5.3.

5.1.2 WDT for Mental Well-being

We consider Mental Stress as a case of Mental health issues. In general, stress is a natural response of the human body to stress-inducing factors (stressors) that lead to significant physiological and behavioural changes citeref70. The stressors are dynamic factors that can produce stress in one individual while sparing others [74]. For example, social media affecting one person's mental health can be proved relaxing for another person, Stress can be predicted from the real-life and social activity of a human.

The flow of mental stress prediction by our proposed framework is following.

1. The proposed framework collects the stressor from dataset containing the smart-watch exercise data (daily activities), social media stories, phone logs, etc. A dataset for training stress prediction model is the response to the questionnaire of stress assessment.

2. The datasets are then pre-processed to remove unnecessary attributes, and noisy instances following the multi-stage pre-processing steps in Fig. 3.3

3. Then proper MLA are selected automatically by the framework for the training set.

4. Classifier is built for stress prediction with the MLA selected by our framework.

5. Finally, the classifiers classify whether a person is stressed or not and report precautions and conditions to the real twin through a wearable interface like a smartwatch.

In this way, a mental well-being twin can be prepared with the capabilities of stress prediction, as discussed in Section 5.4.

5.1.3 WDT for COVID-19 Risk Prediction

The Coronavirus disease 2019 (COVID-19) pandemic has become a burden to the healthcare system around the world. Most of the countries around the world have been transformed their health sector from manual to digital version through telehealthcare service during the pandemic. Integration of Telehealthcare services with dynamic processing and ML to predict COVID risk is an urgent requirement for the current situation.

The flow of mental stress prediction by our proposed framework is following.

1. The proposed framework collects datasets containing daily public health status from public Telehealth services. The data from the telehealthcare service records the healthcare queries from patients. For instance, patient demographic information, health conditions, patient's profile through one service.

2. The datasets are then pre-processed to remove unnecessary attributes, and noisy instances following the multi-stage pre-processing steps in Fig. 3.3

3. Then proper MLA are selected automatically by the framework for the training sets.

4. Classifier is built for COVID-19 risk prediction with the MLA selected by our framework.

5. Finally, the classifier predicts COVID-19 risk from problems reported by patient's

In this way, the framework transforms manual identification of COVID-19 risk from the spreadsheet to a virtual process as discussed in Section 5.5.

5.2 Overall Experiment

This experiment evaluates the proposed framework to demonstrate whether it can select the suitable MLA among the many available MLAs algorithms without human intervention or not. For this purpose, some existing work has been considered, which applied different MLA for predicting various diseases. In our case, we considered ones either selected a similar target disease or same dataset. The goal of this experiment is listed in the following.

- To evaluate if the framework can select suitable algorithms automatically (in terms of similarity score)
- To evaluate if the model can handle heterogeneous data accurately (in terms of prediction accuracy)

5.2.1 Dataset

In order to experiment, we considered 7 different datasets (Table 5.1). The datasets are collected from different sources like Medical records and Prescriptions to satisfy the heterogeneity of data sources. It can be seen from the Table 5.1 is that we considered hetero-

geneous data source, type and disease goals. The reasons behind selecting these datasets are followings.

- These datasets contain the patient's symptoms and signs related to disease diagnosis. As our research focused on disease diagnosis, we found these datasets available and usable for disease diagnosis.
- These datasets are various EHRs containing healthcare practitioner's approved data labels. As the framework needs to deal with disease diagnosis, we relied on these datasets.

Table 5.1: Overview of Datasets

Disease Case	**Diabetes Risk**	**Diabetes**	**Liver Disease**	**Thyroid Risk**	**Stroke Risk**	**Mental Stress**	**COVID Risk**
Training Data Source	Digital Prescription	Clinical Report	Digital Prescription	Digital Prescription	Digital Health Record	Diagnosis Record	Telemedicine Record
Source Type	Public	Collected	Public	Collected	Public	Collected	Public
Data	Binary	Mixed	Neumeric	Mixed	Mixed	Binary	Mixed

5.2.2 Machine Learning Algorithms

There are many MLA as well as several distinct ways for implementing supervised algorithms [13]. As this research focuses explicitly on evaluating the MLA selection mechanism proposed in the framework itself, the existing available implementations of MLA from the Python Libraries [66] are considered here.

In Table 5.2, we demonstrate the settings and model type of the available algorithms in our proposed framework. It can be observed that we selected seven different model types,

including- Support Vector Network, Regression, Ensemble, Probabilistic, Tree, Neural Network, and Non-Parametric Model. The settings column shows different parameter value defined for the MLAs. The datasets have been pre-processed to eliminate the null and missing values by assigning mode values and ignoring missing tuples, respectively. The reasons behind selecting these MLAs and settings are followings.

- These algorithms have been proposed several times for disease classification in the literature.
- Various model types support datasets with various characteristics. Therefore we selected widely used classification model types.
- The proposed framework was intended to avoid rigorous modification of algorithm parameters. Therefore we used standard parameter values provided by sci-kit learn libraries. However, experiment with tuned parameters could be another option to analyze the results.

Table 5.2: Machine Learning Algorithms Available in Proposed WDT Framework

MLA Name	**Model Type**	**Setting**
Support Vector Machine (SVM)	Support Vector Network	SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)
Logistic Regression (LR)	Regression	LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100, multi_class='auto', n_jobs=None, penalty='l2', random_state=None, solver='lbfgs', tol=0.0001, verbose=0, warm_start=False)
Random Forest (RF)	Ensemble	RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, max_samples=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None, oob_score=False, random_state=None, verbose=0, warm_start=False)
Naïve Bayes (NB)	Probabilistic	GaussianNB(priors=None, var_smoothing=1e-09)
Decision Tree(DT)	Tree	DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini', max_depth=None, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort='deprecated', random_state=None, splitter='suitable')
Multilayer Perceptron (MLP)	Neural Network	Perceptron(alpha=0.0001, class_weight=None, early_stopping=False, eta0=1.0, fit_intercept=True, max_iter=1000, n_iter_no_change=5, n_jobs=None, penalty=None, random_state=0, shuffle=True, tol=0.001, validation_fraction=0.1, verbose=0, warm_start=False)
K-th Nearest Neighbour	KNN	KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=None, n_neighbors=3, p=2, weights='uniform')

5.2.3 Performance Metrics

To evaluate the performance of the classifiers, we considered the widely accepted ML evaluation measures against the disease datasets. The following are the Performance metrics considered for this experiment.

- **Prediction Accuracy** Most of the existing work has used accuracy with respective to the True Positive, False Positive, True Negative, False Negative value. Therefore we considered Accuracy one of the performance metrics. However, other MLA accuracy measures are detailed in the Individual Experiment section. **Accuracy=(TP + TN)/(TP + TN + FP + FN)**, where,
  - True Positive (TP) = Risk identified correctly for those who are at risk.
  - False Positive (FP) = Risk-free people identified incorrectly at risk.
  - True Negative (TN) = Risk-free people identified correctly as risk-free.
  - False Negative (FN) = Risk people are identified incorrectly as risk free.
  - Correctly Classified = Percentage of instances classified correctly.
  - Incorrectly Classified= Percentage of instances classified incorrectly.

  In percentage, the value ranges between 0% to 100%. The more the value closer to 100%, the better the accuracy.

- **Cosine Similarity** Another performance metric is the Cosine similarity value. The cosine similarity is calculated as: $Similarity(K_D, K_A) = \frac{K_D \cdot K_A}{\|K_D\| \times \|K_A\|} = \frac{\sqrt{\sum_{i=1}^{n}} K_{D_i} \times \sqrt{\sum_{i=1}^{n}} K_{A_i}}{\sqrt{\sum_{i=1}^{n}} K_{D_i}^2 \times \sqrt{\sum_{i=1}^{n}} K_{A_i}^2}$
  Where K_D, K_A represents the knowledge of Dataset and MLA, respectively. The value ranges between -1 and 1, where -1 represents perfectly dissimilar, and 1 represents perfectly similar.

For the individual experiments some more performance metrics were considered including the followings.

- **Kappa statistic (or kappa coefficient)** A kappa of 1 indicates perfect agreement, whereas a kappa of 0 indicates agreement equivalent to chance. The more the value close to 1 , the more better the performance.
- **Root Mean Square Error (RMSE)** Standard deviation of the prediction errors. It measures how concentrated the data is around the line of best fit.
- **TP Rate** The TP rate is calculated as TP/(TP + FN).
- **FP Rate** The FP rate is calculated as FP/(FP + TN).
- **Precision** The Precision represents the fraction of relevant instances among the retrieved instances. It is calculated as TP/(TP + FP).
- **Recall** represents the fraction of relevant instances that were retrieved. This is calculated as TP /(TP + FN).
- **F-measure** provides the harmonic mean of precision and recall. 2*(Precision*Recall)/(Precision + Recall)
- **ROC (Receiver Operating Characteristic) area** ROC curve is a plot of True positive rate‘and false positive rate. The closer the ROC area to 1.0, more accurate the classifier is.

5.2.4 Result & Findings

In Table 5.3, the second column shows the different existing works that considered different goals, such as diabetes prediction [48], as well as the MLA that they have used to process

the data for the target goal. The third column of the table lists the MLAs that are recommended by the proposed method. The fourth column shows the cosine similarity between the selected MLA, and its target dataset. It can be observed from Table 5.3 that the similarity score is higher in most cases and above 0.80. The highest similarity was found for the Liver Disease and Thyroid Risk Dataset at a score of 0.89. This indicates that the knowledge properties of that dataset and selected algorithms were more alike. By contrast, the COVID Risk Dataset and RF algorithm pair show the lowest similarity score at 0.82. However, this score is also above 0.80 as well as closer to 1.

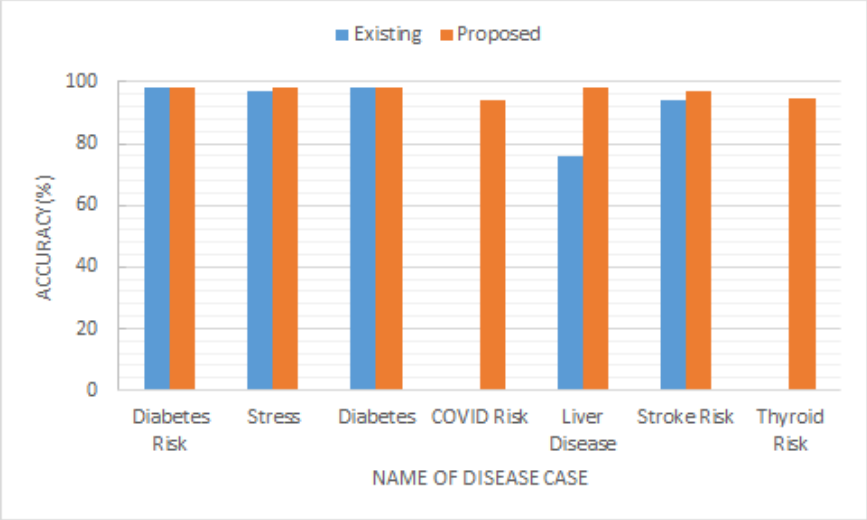


Figure 5.1: Comparative analysis between existing work and proposed framework. Existing Work for Diabetes Risk in [45], Stress in [50] , Diabetes in [77], Liver Disease in [69], Stroke Risk in [72]. Suitable existing work to compare COVID Risk prediction and Thyroid prediction was not available.

In most of the occasions except diabetes risk, the selection of MLA algorithm by the proposed approach is different from the existing one. Therefore in the next stage, we obtained the accuracy of the machine-selected algorithms, and the literature reported the suitable algorithms. The accuracy of the ML algorithms in both approaches for the target

tasks is reported in Fig. A.1. It shows that the accuracy obtained in both existing and proposed cases is comparable.

The existing work proposes NB as the suitable classifier with 75% accuracy for thyroid prediction, whereas the proposed framework achieves 94.5% accuracy with RF (ensemble technique) algorithm. Again, for the Liver Disease prediction, the existing work proposes SVM as the suitable MLA with an accuracy of 76.5%, while the proposed framework recommends an RF algorithm that achieves 98% accuracy. The accuracy of other framework selected alogorithms are also above 94.5%. Stroke risk prediction is also improved by the framework selected algorithm RF from 94% to 97%. The diabetes risk prediction accuracy are at the same level of existing works at 98%. This experiment indicates that the proposed framework can select proper MLA dynamically for a WDT model.

Table 5.3: Comparative result of existing and the proposed work

Goals	Proposed algorithm in existing work	Selected MLA by the proposed framework	Similarity Score
Diabetes Risk	RF	RF	0.86
Mental Stress	RT	RF	0.85
Diabetes	KNN, RF	DT	0.83
COVID Risk	-	RF	0.82
Liver Disease	SVM	RF	0.89
Stroke Risk	DT	RF	0.86
Thyroid Risk	NB	RF	0.89

In the next sections, we present three thorough evaluations for the three use cases. From the non-communicable diseases, we select the Diabetes risk, and the other two selections remain the same. The goals of this experiment are to find the following.

- The comparison of various MLA performance with existing work.
- Comparison of performance of multiple MLAs to verify that the automatically selected MLA by our framework performs with the suitable accuracy.

5.3 Individual Experiment: NCD Risk Prediction

From the seven NCDs used in the above experiment, we picked the diabetes risk prediction to evaluate the use case of NCD risk prediction. Following the method of this research, the experimental procedure is divided into two phases. In the first phase, the training procedure is performed using a verified dataset of diabetes [1]. This dataset has been

collected with ethical approval and informed consent from actual patients from a diabetic hospital. All data are collected from the patient's prescription, where a medical officer identified a patient as diabetes potential. More specifically, patients who are recommended for the clinical test are classified as positive. This dataset is used to predict the likelihood of diabetes at early-stage from common signs and symptoms such that potential loss of valuable life from diabetes can be minimized.

In the second phase, the test data has been produced from simulation and prototype to evaluate the performance of different classification algorithms. For the simulation network of wearable sensors, a sensor network was constructed by updating examples of cooja simulator using Contiki Operating System [17]. The sensors have been modified using the Python programming language. Finally, the results have been compared with other existing works focusing on diabetes risk prediction using ML.

5.3.1 Experiment Procedure

At this stage, a verified training dataset [1] is used as ground truth for the training purpose. There are total 17 attributes, including one class attribute. The distribution of positive (clinical diabetes test prescribed) and negative (clinical diabetes test not prescribed) classes in the training dataset is depicted in Figure 4.1. It can be observed that the dataset includes a class variation for all 16 attributes.

Following the attribute distribution, the classifier was trained with training datasets applying 11 classification algorithms: RF, DT, NB, BN, MLP, SVM-Polykernel and SVM-RBFKernel, LR, RT, AdaBoost, Bagging, and KNN. The time to build model for each classification algorithm has been shown in Figure 5.2.

It can be observed from Figure 5.2 that the minimum time taken to build a classifier is

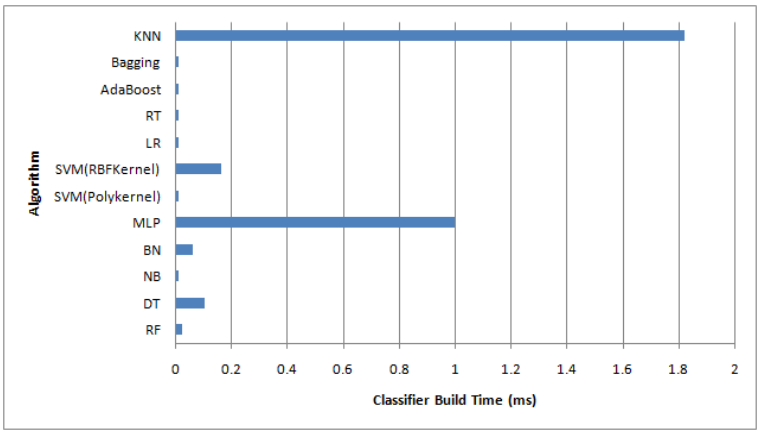


Figure 5.2: Time taken to build classifier with different algorithms during training.

0.01s for NB, SVM (PolyKernel), LR, RT, AdaBoost, and Bagging. The KNN algorithm required the maximum time at 1.82s. Time to build model was recorded by using the whole training dataset. The

The tree from the DT classification is depicted in Figure 5.4. The tree's root is polydipsia, which then branches to polyuria and afterwards reach the class attribute.

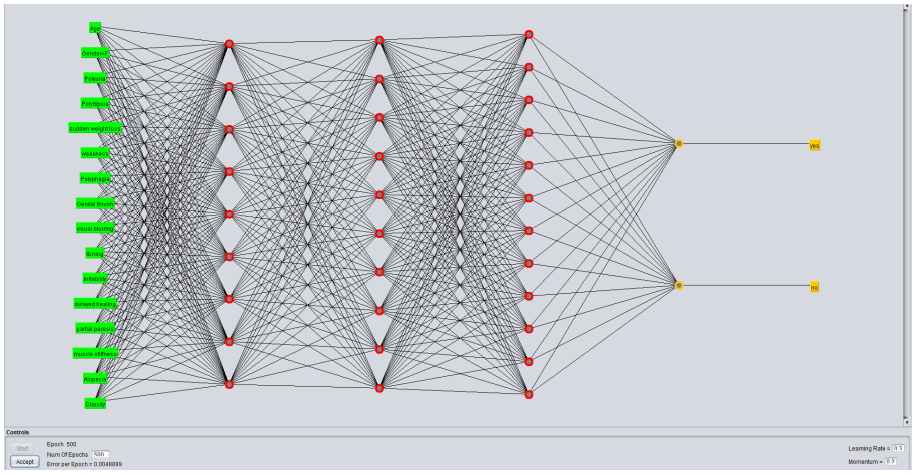


Figure 5.3: Neural network of MLP classification.

Table 5.4: Confusion matrix for the classification algorithms.

(a) Confusion matrix of RF

Positive	Negative	Actual Class
69	12	Positive
0	134	Negative

(b) Confusion matrix of DT

Positive	Negative	Actual Class
64	17	Positive
3	131	Negative

(c) Confusion matrix of NB

Positive	Negative	Actual Class
58	23	Positive
17	117	Negative

(d) Confusion matrix of BN

Positive	Negative	Actual Class
54	27	Positive
13	121	Negative

(e) Confusion matrix of MLP

Positive	Negative	Actual Class
65	16	Positive
3	131	Negative

(f) Confusion matrix of SVM

Positive	Negative	Actual Class
63	20	Positive
13	21	Negative

(g) Confusion matrix of LR

Positive	Negative	Actual Class
62	19	Positive
15	119	Negative

(h) Confusion matrix of RT

Positive	Negative	Actual Class
66	15	Positive
0	134	Negative

(i) Confusion matrix of AdaBoost

Positive	Negative	Actual Class
60	21	Positive
17	117	Negative

(j) Confusion matrix of Bagging

Positive	Negative	Actual Class
63	18	Positive
4	130	Negative

(k) Confusion matrix of KNN

Positive	Negative	Actual Class
66	15	Positive
0	134	Negative

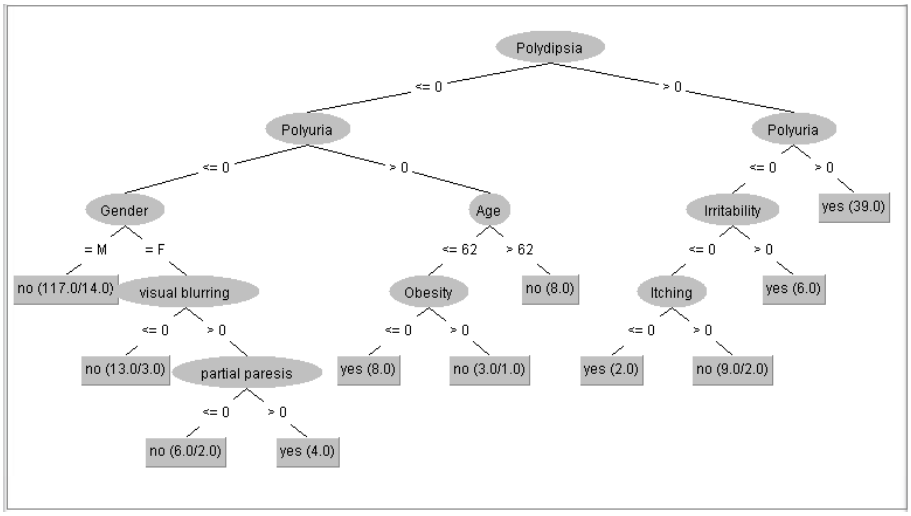


Figure 5.4: The tree from decision tree classification.

The Kappa statistics and RMSE comparison is provided in Figure 5.5. These two statistical measures are considered widely for ML performance evaluation. The kappa statistics value of RF, RT, AdaBoost, and KNN is mostly closer to 1, indicating these classification algorithms' efficiency for this problem. On the other hand, the RMSE value of KNN, RT, MLP, and RF is the least, which proves the efficiency of those algorithms for the target prediction task.

To get a more detailed view of the classifier performance, other accuracy measures like TP rate, FP rate, Precision, Recall, ROC area, and F-measure are illustrated in Figure 5.6. The highest value of these measures is approximately 0.93, 0.12, 0.94, 0.93, 0.93 and 0.93, respectively, for multiple algorithms like RF and RT. The SVM performs worst among these algorithms. The SVM (RBFkernel) accounts for the highest FP rate at 0.23 and the subsequent ones are for NB and BN at 0.22, and so on.

Based on all the analysis above, it is evident that RF performed suitable in terms of model build time during training and accuracy measures during testing.

5.3.2 Comparison with Existing Work

Existing work mainly considers clinical datasets for diabetic prediction, not for early-stage risk prediction of diabetes. Different works consider different datasets with a diverse set of attributes. However, we took the context of diabetes and ML to compare our work that considers dataset and corresponding attributes for early-stage diabetes risk prediction. A comparison of the proposed work with the existing work has been outlined in Table 5.5.

The suitable accuracy for each algorithm is highlighted in Table 5.5. It is evident from the table that our proposed work provides the suitable accuracy in most cases. However, the work in [83] comes next, providing the suitable accuracy for three algorithms DT, NB, and SVM (Polykernal) and then in [29] for LR algorithm. Also, the - sign in the table cell represents that the cited work does not use the corresponding algorithm. It can be observed from the table that each of the existing works individually has used 3-4 classification techniques. In contrast, we analyzed our data with 11 classification techniques that have been used for diabetes prediction in the literature.

Table 5.5: Comparison of accuracy of our work with existing work.

Algorithms	RF	DT	NB	BN	MLP	SVM (Polykernel)	SVM (RBFKernel)	LR	RT	AdaBoost	Bagging	KNN
This work	94.02	91.02	81.03	81.02	92.3	85.14	82.28	84.67	94	83	89	93
[77]	-	-	73.82	N/A	71	-	-	69	-	-	-	83
[81]	76.3	73.82	-	-	N/A	N/A	65.1	-	-	-	-	-
[29]	N/A	70	-	75	77	74	-	98	-	-	-	-
[83]	N/A	93	92	N/A	91	91	82	N/A	-	-	-	-

5.3.3 Result & Findings

Overall, it is evident from the experimental results that although multiple algorithms built the model in minimum time within 0.01s, their accuracy varied significantly. For instance, the model is built in 0.01s by applying Bagging, AdaBoost, RT, LR, SVM(PolyKernel),

and NB.

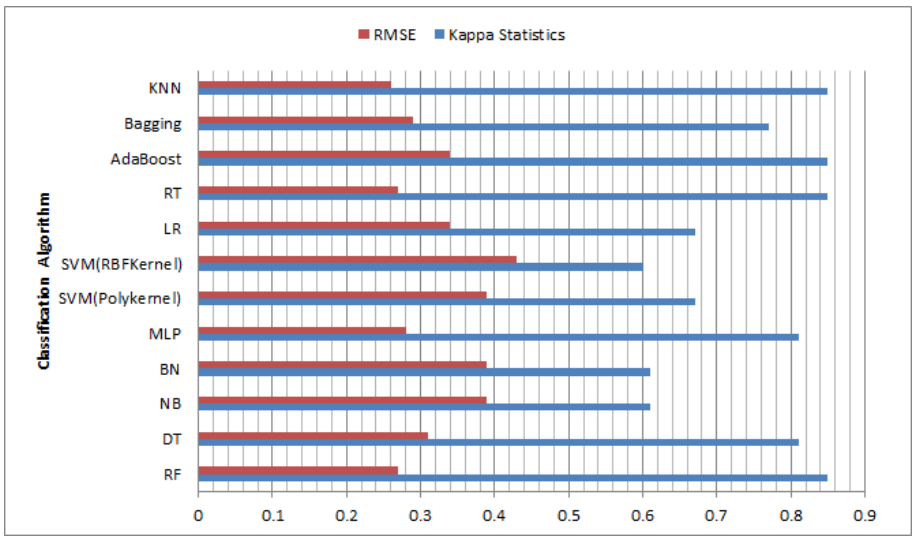


Figure 5.5: Statistical measures of classification algorithm.

However, the accuracy obtained by RF and RT is nearly 10% higher at 94% than SVM at 85.14%. Again, the RF provides the highest accuracy at 94.02% and ROC area at 0.97,

Though both RF and RT obtain the same value for TP rate, FP rate, Precision, Recall, F-measure, and ROC accuracy measures, the better accuracy supports RF as the suitable algorithm for this experiment. This diversity of results provides interesting insights, such as a) Although several algorithms provide almost similar accuracy, the classification algorithms may require variable training time and b) For the prediction of NCDs, performance should be evaluated in both training and testing phase.

Form this experiment we found that the framework selected algorithm RF is suitable for diabetes risk prediction in terms of accuracy. However, RT

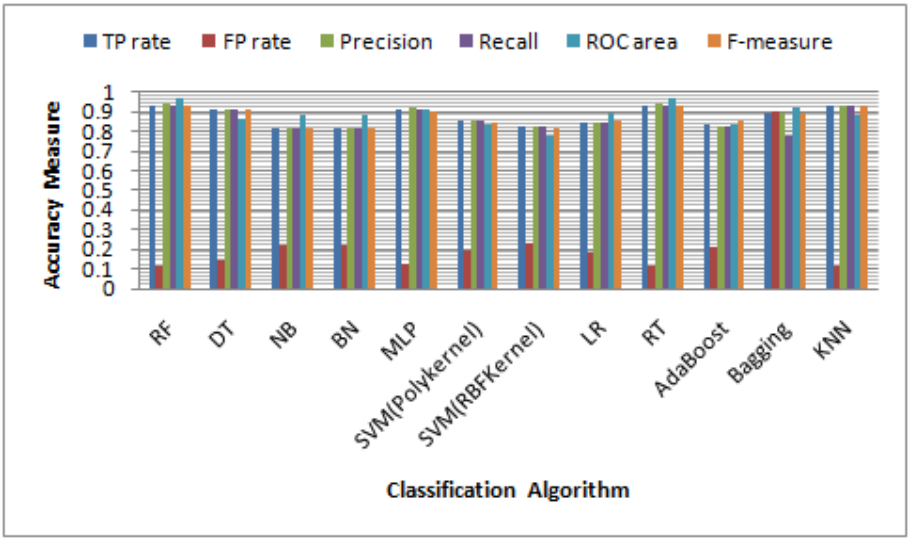


Figure 5.6: Accuracy measures of classification algorithm.

5.4 Individual Experiment: Mental Stress Prediction

This study involves 27 volunteer subjects between 18-50 years old, monitored over one week from 20th June 2020 to 30th June 2020. The details of the source features are provided in Table B.1.

5.4.1 Experimental Procedure

The average number of *outgoing calls, incoming calls, missed calls, and call duration* are combined to a derived attribute *communication pattern*, walking duration, exercise duration, inactive duration are combined to the activity pattern. The *sleep duration* was transformed into the *sleep pattern*, and *social media active time and chat time* were combined with the *social media activity pattern*. Due to the unavailability of the required knowledgebase, we acquired the threshold based on generic rules approved by a medical

practitioner. For an example, the sleeping hour less than 8 hours was labelled as *Disturbed* sleep pattern.

For ML classification, the following steps are followed.

- In this work, we employed ML classification algorithms that have been used for stress detection in the contemporary literature. The selection includes SVM, KNN, RF, NB, MLP, and DT. The classifiers were built using the default settings of the python scikit-learn library.
- Different senses of a person have been recorded at 0.5-second intervals using a smartwatch, smartphone, and smart band sensors. Then these samples were summarized at 1-week interval to construct the training dataset. The training dataset consists of features DTF and labels L1=(stress), 0=(non-stress).
- The ground truth for labeling the training data was obtained from the domain experts. The classifiers were trained with this training dataset where supervised ML algorithms have been used for achieving better precision.
- We eliminated the null value and implemented a dynamic pre-processing function to standardize attributes so that the dataset fits multiple classifiers using python sklearn pre-processing functions.
- We conducted 80:20 percentage split to obtain the performance metrics.

5.4.2 Comparison with Existing Work

The comparison of the proposed work with existing work is provided in Table 5.6. The proposed work considers heterogeneous data to get the digital status of stressors. Table 5.6,

Table 5.6: Comparison of accuracy with existing mental stress prediction

Work	Selected Algorithm	Obtained Accuracy	Number of Subjects	Number of Feature	Sensing
Proposed	RF	98%	27	20	Wearable, Social, Software
[75]	SVM	80.9%	10	5	Wearable
[50]	RT	97%	17	8	Wearable
[73]	KNN	87.5%	46	112	External
[84]	DT	80.9%	20	6	Wearable

shows the comparison in terms of selected algorithms, accuracy, number of subjects involved in the experiment, number of feature, and type of sensing.

Overall, the SVM mental stress classifier in [75] obtained 80.9% accuracy on wearable sensed data with five features. The DT classifier in [84] obtained the same level of accuracy using six features of a wearable sensed dataset. The KNN model in [73], obtained a better accuracy than [75] and [84]. The authors in this work involved the highest number of subjects and features. However, they utilized 112 features providing various information on keyboard typing. In contrast, the RF classifier in our proposed work obtained 98% accuracy, with 27 subjects and 20 features, including information about daily life activity, social media activity of an individual. We used physiological, behavioural, environmental data about the subjects for mental stress prediction. The number of subjects and the features in our experiment were moderate compared to the other existing work. In addition, we employed heterogeneous data and obtained comparatively good accuracy.

5.4.3 Results & Findings

It is evident from the accuracy results in Fig.5.7a that RF classified mental stress at 98% accuracy, whereas SVM and KNN classified it with lower accuracy at 83.33%. The lower accuracy of SVM and KNN is because of a higher rate of true negative value in the dataset.

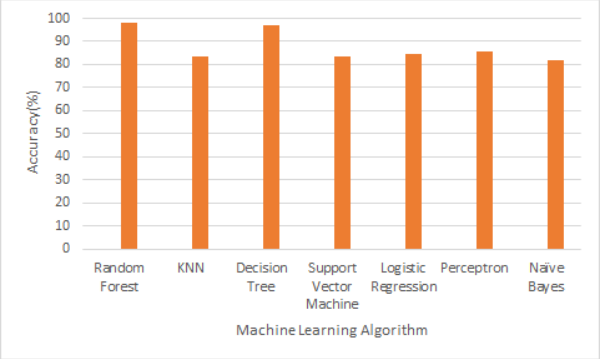
Both the test accuracy and the ROC equation use the TN value. Another reason is that SVM and KNN are not suitable for small datasets like NB or DT.

Particularly, this experiment was conducted on a very small dataset. Although in the case of accuracy, we obtained 83.33% for SVM and KNN. However, the ROC area value (5.7b) of those classifiers was only at 0.5. In contrast, the ROC of the RF algorithm was almost at 1.0.

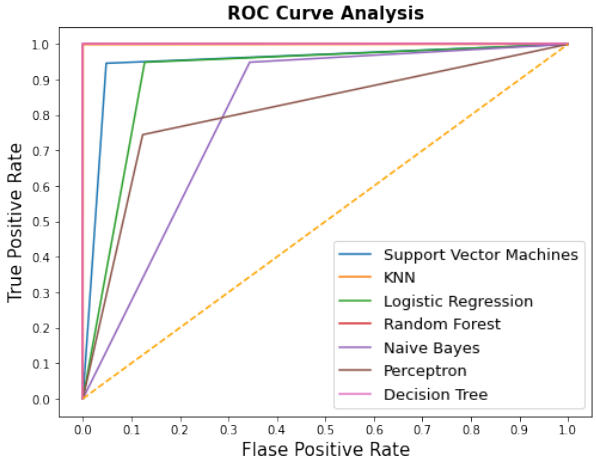
We agree that RF is a good fit for the mental stress prediction dataset. Because the precision and Recall value of RF was also at 0.98 and higher than the other algorithms. Even though the values of results are fluctuating, the lower performance of other MLAs shows that the selection by our proposed framework is suitable for the mental stress dataset.

5.5 Individual Experiment: COVID-19 Risk Prediction

To experiment, an original dataset from the Telemedicine Service of Bangladesh was collected. This dataset contains the record of one-month data from 1/04/2020 to 30/04/2020, while corona-virus started to spread in Bangladesh significantly. The record of the patient's call, system's input, and doctor's evaluation have been collected in the CSV data format. The attribute details of the dataset are provided in Table B.2. The MLA and it's setup



(a) Accuracy comparison of machine learning classifiers



(b) ROC area comparison of machine learning classifiers

Figure 5.7: Performance of ML classifiers in stress prediction

discussed in Table 5.2.

	Gender	Division	Diseses	Category	Symptom	ageGroup	class
0	female	Rajshahi	NaN	NaN	Tonsillitis	20-30	unknowr
1	male	Dhaka	NaN	NaN	Dry cough	31-50	unknowr
2	male	Dhaka	Medicine	Respiratory	Common cold	0	high risk
3	female	Dhaka	Surgery	ENT	Sore throat	20-30	low risk
4	male	Dhaka	NaN	NaN	Viral fever	20-30	unknowr

Figure 5.8: Clean data after pre-processing.

5.5.1 Experimental procedure

The Data Mapping and Pre-Processing was done in followings ways. After selecting the training data, the datasets undergo multiple-stage as described in the system architecture. For assigning class first, the attribute symptom was checked with the condition that indicates dry cough, sore throat, viral fever, breathlessness, or direct corona symptom to be labelled as high risk. Then, the attribute category was checked to see whether it is respiratory, respiratory system, or cardiovascular for assigning high-risk class. If the symptoms were unknown, the class was null, but if the category was unknown, the class was labeled as unknown. Finally, the data were categorized into i) high risk - COVID high risk, ii)low risk- COVID low risk.The age attribute was transformed to categorical from numerical. The data tuples with null class value had been dropped. After following these stages the data was cleaned and the data frame looked as illustrated in Fig. 5.8.

From Table 5.7, it is evident that after the pre-processing stage, the memory usage by

Table 5.7: Memory used by training and test dataset before and after cleaning.

Dataset	**Memory used before pre-processing**	**Memory used after pre-processing**
Training	1228.8+KB	579.5+KB
Test	92.9+KB	73.7+KB

both the training and test dataset dropped significantly. For the training set, the memory used was reduced to almost half. This indicates that proposed data pre-possessing can save memory while the classification from a large dataset.

After this stage, the datasets were fit to the classifiers with library functions. The class was transformed to binary 0 and 1 after final null value elimination. LogisticRegression classifier imported as a linear regression model for classification. SVC was imported to implement SVM. For neural network model implementation Perceptron model was imported. To implement a tree-type model, DecisionTreeClassifier was imported, and for ensemble model implementation, RandomForestClassifier was used. All these classifiers were imported from Sklearn libraries. The models had been implemented and compared using the same python notebook.

Suitable existing work to compare with this result was not available for this experiment. Therefore, the comparison section is excluded from this section.

subsectionResults & Findings

It can be observed from Fig. 5.9 that RF provided highest accuracy at 98%. The KNN provides 69.95%, Support Vector Machines provided 95.03% accuracy, Logistic Regression obtained 95.02% accuracy, Perceptron showed 89.44% and, Naive Bayes provided 83.92% accuracy. However, except for the KNN the other algorithms provided decent accuracy above 80%.

We observed that comparatively, the RF performed suitable in terms of the mentioned

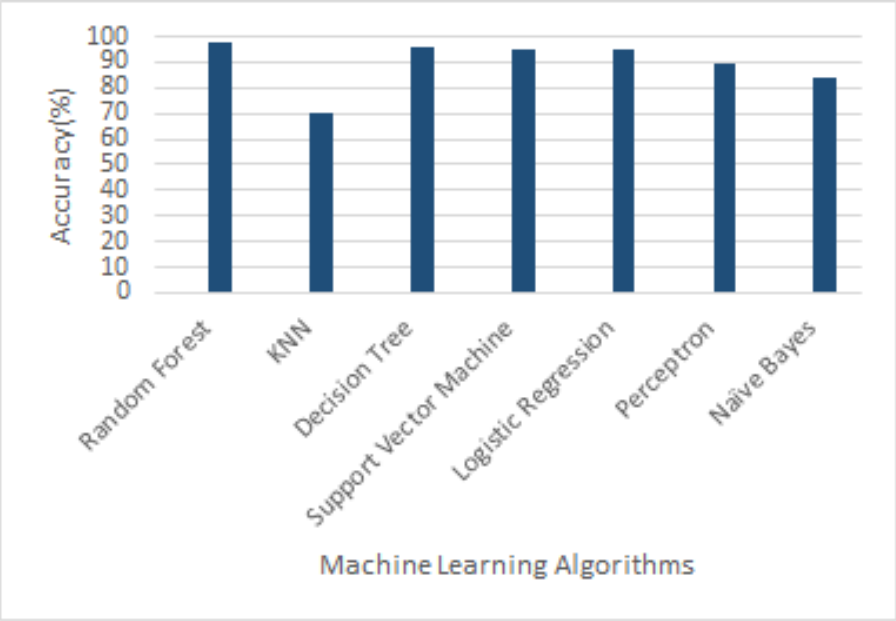


Figure 5.9: Accuracy measures of classification algorithm.

evaluation metrics. Our framework also selected DT for the COVID risk prediction. It indicates that the framework selected algorithm for this use case is suitable for the target disease prediction.

We implemented a visualization function to visualize statistics of different attributes from the test dataset in a meaningful representation. The predicted class was transformed into high risk = 1 and low risk = 0 and plotted into a bar diagram.

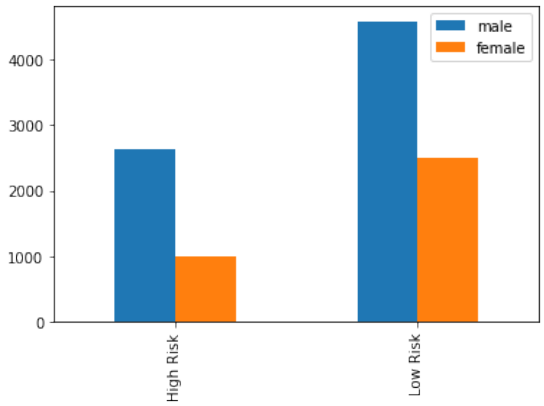
From Fig.5.10, it can be observed that males mainly were at risk. Fig.5.10a, it can be observed that males mainly were at risk. Fig.5.10b, shows that the Dhaka division is at more risk and the age group of 20-30 is vulnerable to risk than other age groups. This visualization provides essential insights that can be used to improve medical preparedness where risk is high. The type of special arrangements can also be prepared for risky age groups and gender. After comparing this to the original data, we obtained that statistically,

the same majority was depicted for the test data tuples.

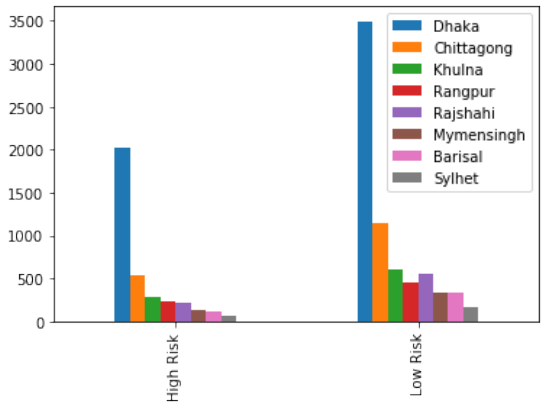
5.6 Summary of Findings

The following are the Summary of the Experiments found to form the above Experiments.

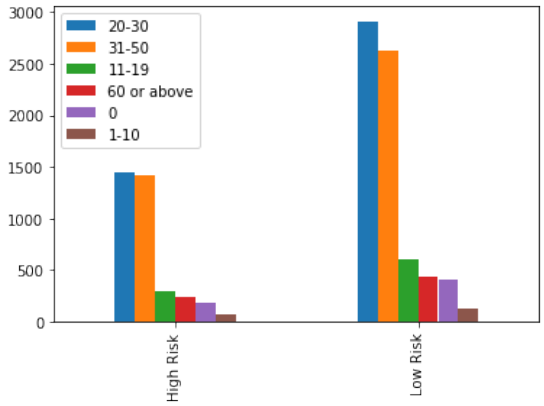
- The first experiment shows that the proposed framework can handle heterogeneous disease data and classify multiple Diseases at or above 94.5% accuracy. This satisfies our first objective- to provide a multi-disease prediction model.
- In addition, the first experiment shows that the framework selects MLA at a high similarity score ranging from 0.82-0.89. This confirms the reliability of the framework as well as the compatibility of the framework with precision healthcare
- From the second experiment, we found that the framework's selected algorithm are originally provides better accuracy than other classification algorithms. This verifies the selected models by our framework can be utilized for health and well-being related prediction.
- The selection of datasets contributed in improving prediction accuracy. Most of the datsets for the experiment are EHR, which includes healthcare practitioner's labelling.
- All the prediction problem was converted as binary classification problem. Therefore, ensemble technique RF was selected six times and Decision Tree was selected once. However, his framework will also work for other types of classification or clustering problem (Fig. A.1).



(a) Visualization of gender at high and low risk.



(b) Visualization of division at high and low risk.



(c) Visualization of age group at high and low risk.

Figure 5.10: Covid risk group visualization

Chapter 6

Conclusion and Future Work

In this research, we proposed a framework for multiple disease diagnoses. This work aims at predicting multiple diseases or risk of disease of individuals dynamically. Therefore, we designed a dynamic framework to selects dataset and machine learning classifiers for each type of disease prediction. This data is then prepared for the disease prediction classifier through a dynamic testing method. The benefits of research extend to multiple disease diagnosis through a single system.

We designed algorithms and implemented associated functions to evaluate the framework. We considered various use cases and evaluated the performance of the framework in terms of numerous performance metrics. These experiments helped to explore potentialities where digital twin can support the disease diagnosis at early stage. In the following sections, we detail issues, limitations, and possible improvements of this thesis.

6.1 Addressed Issues

Digital Twin in healthcare is still at its incubation period. We faced several issues while conducting this research. Precisely, during the implementation and evaluation of the thesis, we found a significant scarcity of datasets, existing implementation, etc. Some notable issues are the followings.

6.1.1 Restriction to Access Data

One of the biggest issues of this thesis was data collection for verifying the framework. The whole research has been conducted during the COVID-19 pandemic situation and remotely. Therefore, it was complex to get approval to access a vast amount of hospital data.

To resolve this challenge, we utilized existing public datasets. In addition, we attempted to evaluate the performance of the framework we employed samll to vast datasets from available data sources.

We believe stakeholders like Government and Healthcare Authorities can play a vital role by approving access to healthcare data. This will be helpful to ensure each healthcare at the right time.

6.1.2 Scarcity of Appropriate Dataset

There exist some publicly medical dataset can provide data presenting sign symptom and risk factors. However, the ground truth are not usually from the medical practitioner. We found it very challenging to get such dataset while conducting this research.

We used the four EHRs and Telehealth records to resolve this challenge of the healthcare practitioner's approval. Those are limited in size but consists real class label from the medical practitioners.

The collaboration of hospitals, clinics and diagnostic systems can accelerate digital healthcare and transform it to automated healthcare by sharing huge EHRs. This will be beneficial to understand patterns of various causes of a particular disease.

6.1.3 MLA Knowledge Extraction

The knowledge of goal for this research is same for the seven datasets. Also, the python libraries were helpful to extract data type, dataset size, etc. information. However, we struggled to extarct MLA knowledge.

We utilized the *capabilities* information of the selected algorithms from the WEKA classifiers (see Fig. A.2). However, as we designed our similarity function for matching string similarity all these knowledge required to modify into the same format of the dataset and goal knowledge.

A dataset of MLA knowledge can be beneficial to reduce the overhead of MLA knowledge extraction. Similar to the dataset knowledge, built-in libraries to extract MLA knowledge, can be another candidate solution.

6.2 Limitations

The limitations of this research are listed followings.

- The data collection and experiments for two predictions were conducted on a very small dataset. However, all the data are original and collected with the consent of the real users.
- The data for ML selection are predefined due to the unavailability of specific knowledge database. However, the data were collected from the algorithm capabilities information available int the Weka tool.
- The comparison of two disease cases- thyroid and covid risk were not possible. Because the DT in healthcare is at its primary stage of adoption. The existing works for these two use cases were not a good fit for the comparison.

6.3 Possible Improvements

As stated above, the main goal of our proposed WDT framework is to support multiple healthcare; future work may focus on extending or expanding the proposed framework in one or more of the following ways.

6.3.1 Improving Data Acquisition

Collect real-time data like image, video, hepatic, some of which we are already collecting but not making full use of yet. The more data we collect will add more detail of human health. In addition, more complex health issues will be predicted by the system. Fig. 6.2

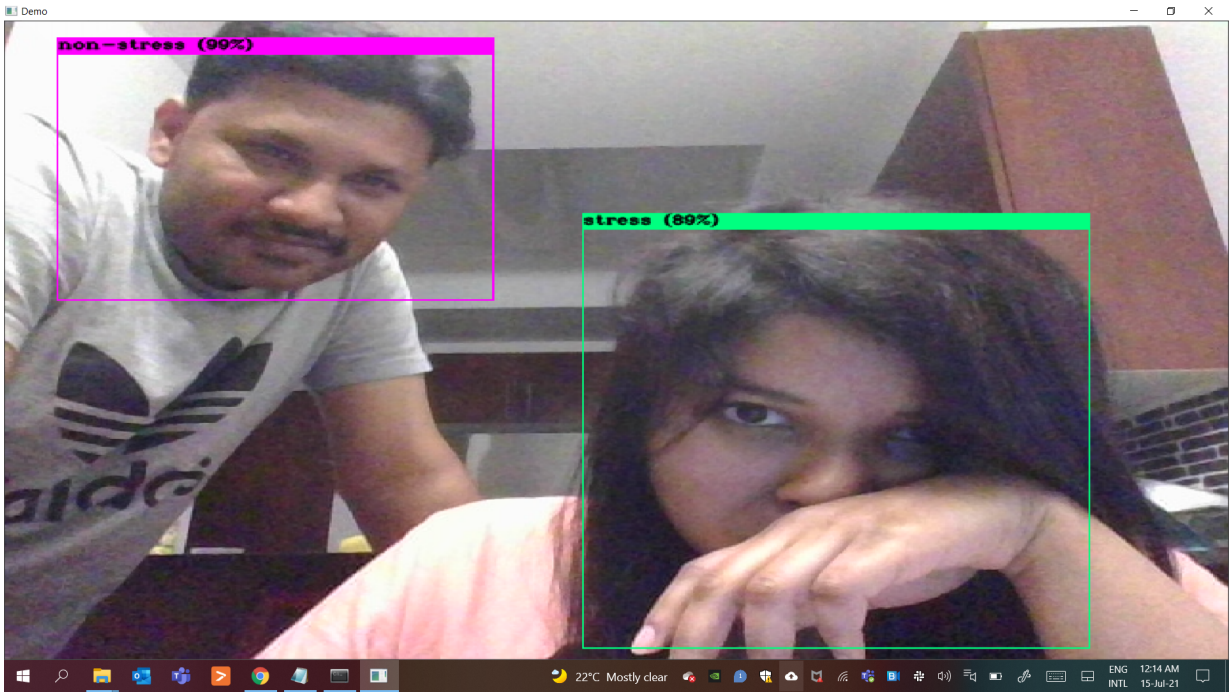


Figure 6.1: Mental stress detection from webcam data.

depicts an example of using real-time webcam data for stress prediction from an on-progress extension of this research.

6.3.2 Improving Prediction Mechanism

Explain the prediction. The prediction explanation will be revolutionary for individual healthcare prediction by providing the right recommendation for the right risk factors. More specifically, which risk factors affect an individual will provide insights to determine appropriate recommendations for the patient. Thanks to Explainable AI (XAI), which has made it possible to retrieve explanations for individual predictions. In Fig. 6.2 We demonstrate what difference the integration of XAI to healthcare can bring.

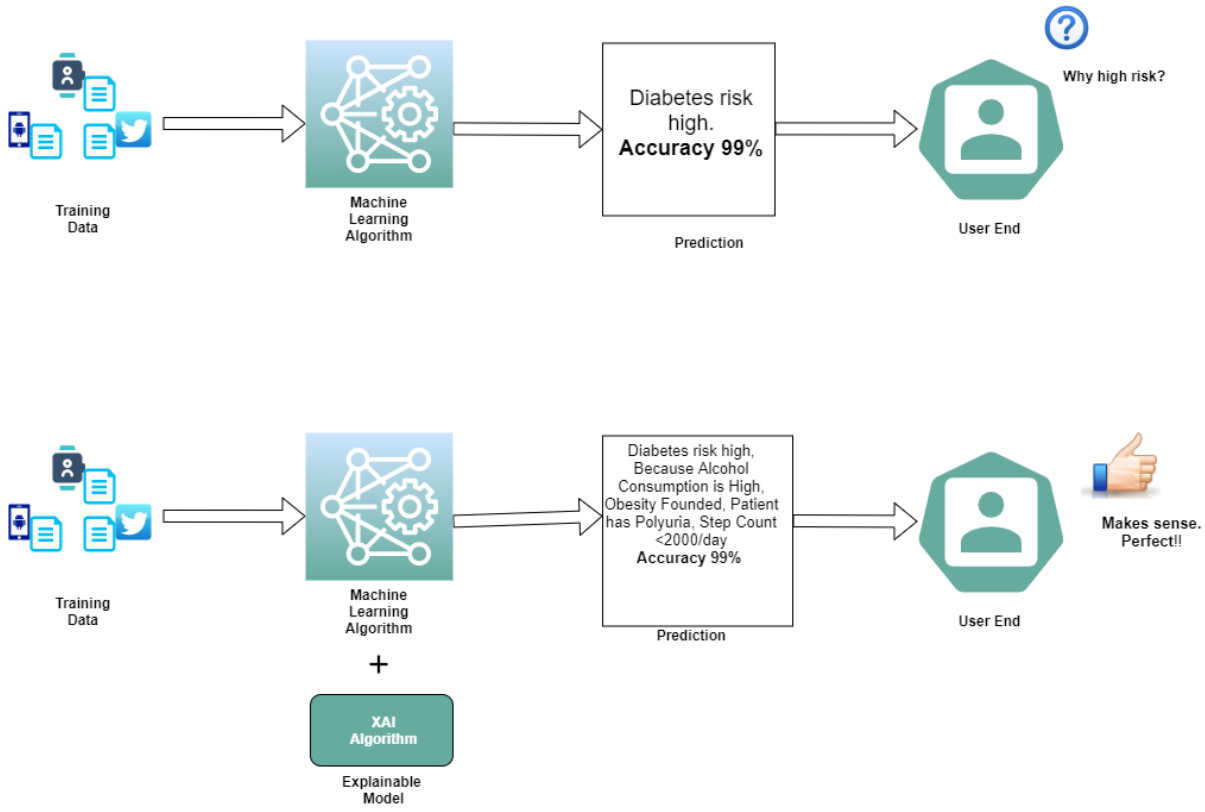


Figure 6.2: Difference between prediction with and without Explainable AI.

6.3.3 Improving Implementation

Implement an application feature for the smartwatch fitness apps to monitor an individual's health, predict disease risks and recommend specific measures. Nowadays, the apps related to fitness bands like Fitbit, Mi band or smartwatch like Apple watch are offering activity monitoring services to their users. The implementation of our proposed WDT framework may add a new feature for multiple disease prediction features in the future.

References

[1] *UCI machine learning repository*, accessed on 3 July 2021. http://archive.ics.uci.edu/ml/datasets/Early+stage+diabetes+risk+prediction+dataset.

[2] *Dassault Systems*, accessed on 7 July 2021. https://discover.3ds.com/digital-twin.

[3] *Digitwin*, accessed on 7 July 2021. https://www.mai.ai/digitwin/.

[4] *GE healthcare*, accessed on 7 July 2021. https://www.gehccommandcenter.com/digital-twin.

[5] *Hospital blue codes*, accessed on 7 July 2021. https://blog.thoughtwire.com/imagine-a-hospital-with-zero-code-blues.

[6] *IBM Digital Twin*, accessed on 7 July 2021. https://www.ibm.com/ca-en/products/digital-twin-exchange.

[7] *NHS info*, accessed on 7 July 2021. https://www.challenge.org/wp-content/uploads/2019/03/Digital_Era_02.pdf.

[8] *Phillips*, accessed on 7 July 2021. https://www.philips.com/a-w/about/news/archive/blogs/innovation-matters/20180830-the-rise-of-the-digital-twin-how-healthcare-can-benefit.html.

[9] *Simens*, accessed on 7 July 2021. https://www.siemens-healthineers.com/services/value-partnerships/asset-center/white-papers-articles/value-of-digital-twin-technology.

[10] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

[11] Shohin Aheleroff, Xun Xu, Ray Y Zhong, and Yuqian Lu. Digital twin as a service (dtaas) in industry 4.0: an architecture reference model. *Advanced Engineering Informatics*, 47:101225, 2021.

[12] Kazi Masudul Alam and Abdulmotaleb El Saddik. C2ps: A digital twin architecture reference model for the cloud-based cyber-physical systems. *IEEE access*, 5:2050–2062, 2017.

[13] Rahman Ali, Sungyoung Lee, and Tae Choong Chung. Accurate multi-criteria decision making methodology for recommending machine learning algorithm. *Expert Systems with Applications*, 71:257–278, 2017.

[14] Shawkat Ali and Kate A Smith. On learning algorithm selection for classification. *Applied Soft Computing*, 6(2):119–138, 2006.

[15] M Ambika and K Latha. Non-communicable diseases: an approach for prediction using machine learning technique. *Int J Appl Eng Res*, 10(55):806–810, 2015.

[16] Namrata Bagaria, Fedwa Laamarti, Hawazin Faiz Badawi, Amani Albraikan, Roberto Alejandro Martinez Velazquez, and Abdulmotaleb El Saddik. Health 4.0: Digital twins for health and well-being. In *Connected Health in Smart Cities*, pages 143–152. Springer, 2020.

[17] BA Bagula and Zenville Erasmus. IoT emulation with cooja. In *ICTP-IoT workshop*, 2015.

[18] Barbara Rita Barricelli, Elena Casiraghi, and Daniela Fogli. A survey on digital twin: definitions, characteristics, applications, and design implications. *IEEE access*, 7:167653–167671, 2019.

[19] Barbara Rita Barricelli, Elena Casiraghi, Jessica Gliozzo, Alessandro Petrini, and Stefano Valtolina. Human digital twin for fitness management. *Ieee Access*, 8:26637–26664, 2020.

[20] Bergthor Björnsson, Carl Borrebaeck, Nils Elander, Thomas Gasslander, Danuta R Gawel, Mika Gustafsson, Rebecka Jörnsten, Eun Jung Lee, Xinxiu Li, Sandra Lilja, et al. Digital twins to personalize medicine. *Genome medicine*, 12(1):1–4, 2020.

[21] Avrim L Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997.

[22] Matthias Braun. Represent me: please! towards an ethics of digital twins in medicine. *Journal of Medical Ethics*, 47(6):394–400, 2021.

[23] Koen Bruynseels, Filippo Santoni de Sio, and Jeroen van den Hoven. Digital twins in health care: ethical implications of an emerging engineering paradigm. *Frontiers in genetics*, 9:31, 2018.

[24] Neeraj Kavan Chakshu, Igor Sazonov, and Perumal Nithiarasu. Towards enabling a cardiovascular digital twin for human systemic circulation using inverse analysis. *Biomechanics and Modeling in Mechanobiology*, 20(2):449–465, 2021.

[25] Min Chen et al. Urban healthcare big data system based on crowdsourced and cloud-based air quality indicators. *IEEE Communications Magazine*, 56(11):14–20, 2018.

[26] Angelo Croatti, Matteo Gabellini, Sara Montagna, and Alessandro Ricci. On the integration of agents and digital twins in healthcare. *Journal of Medical Systems*, 44(9):1–8, 2020.

[27] Laizhong Cui, Shu Yang, Fei Chen, Zhong Ming, Nan Lu, and Jing Qin. A survey on application of machine learning for internet of things. *International Journal of Machine Learning and Cybernetics*, 9(8):1399–1417, 2018.

[28] Jürgen Dieber and Sabrina Kirrane. Why model why? assessing the strengths and limitations of lime. *arXiv preprint arXiv:2012.00093*, 2020.

[29] Ashok Kumar Dwivedi. Analysis of computational intelligence techniques for diabetes mellitus prediction. *Neural Computing and Applications*, 30(12):3837–3845, 2018.

[30] Abdulmotaleb El Saddik. Digital twins: The convergence of multimedia technologies. *IEEE multimedia*, 25(2):87–92, 2018.

[31] Abdulmotaleb El Saddik, Hawazin Badawi, Roberto Alejandro Martinez Velazquez, Fedwa Laamarti, Rogelio Gámez Diaz, Namrata Bagaria, and Juan Sebastian Arteaga-Falconi. Dtwins: a digital twins ecosystem for health and well-being. *IEEE COMSOC MMTC Commun. Front*, 14:39–43, 2019.

[32] Haya Elayan, Moayad Aloqaily, and Mohsen Guizani. Digital twin for intelligent context-aware iot healthcare systems. *IEEE Internet of Things Journal*, 2021.

[33] Rahatara Ferdousi, M Anwar Hossain, and Abdulmotaleb EL Saddik. Early-stage risk prediction of non-communicable disease using machine learning in health cps. *IEEE Access*, 2021.

[34] Aidan Fuller, Zhong Fan, Charles Day, and Chris Barlow. Digital twin: Enabling technologies, challenges and open research. *IEEE access*, 8:108952–108971, 2020.

[35] Mohamed Medhat Gaber, Adel Aneiba, Shadi Basurra, Oliver Batty, Ahmed M Elmisery, Yevgeniya Kovalchuk, and Muhammad Habib Ur Rehman. Internet of things and data mining: From applications to techniques and systems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):e1292, 2019.

[36] Rogelio Gámez Díaz, Qingtian Yu, Yezhe Ding, Fedwa Laamarti, and Abdulmotaleb El Saddik. Digital twin coaching for physical activities: A survey. *Sensors*, 20(20):5936, 2020.

[37] Stephen R Garner et al. Weka: The waikato environment for knowledge analysis. In *Proceedings of the New Zealand computer science research students conference*, volume 1995, pages 57–64, 1995.

[38] Saikat Gochhait and Aashish Bende. Leveraging digital twin technology in the healthcare industry–a machine learning based approach. *European Journal of Molecular & Clinical Medicine*, 7(6):2547–2557, 2020.

[39] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. Xai—explainable artificial intelligence. *Science Robotics*, 4(37), 2019.

[40] Shah Ahsanul Haque, Syed Mahfuzul Aziz, and Mustafizur Rahman. Review of cyber-physical system in healthcare. *international journal of distributed sensor networks*, 10(4):217415, 2014.

[41] Konrad Höffner, Sebastian Walter, Edgard Marx, Ricardo Usbeck, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. Survey on challenges of question answering in the semantic web. *Semantic Web*, 8(6):895–920, 2017.

[42] M Anwar Hossain, Rahatara Ferdousi, and Mohammed F Alhamid. Knowledge-driven machine learning based framework for early-stage disease risk prediction in edge environment. *Journal of Parallel and Distributed Computing*, 146:25–34, 2020.

[43] M Anwar Hossain, Rahatara Ferdousi, Sk Alamgir Hossain, Mohammed F Alhamid, and Abdulmotaleb El Saddik. A novel framework for recommending data mining algorithm in dynamic iot environment. *IEEE Access*, 8:157333–157345, 2020.

[44] M Shamim Hossain. Cloud-supported cyber–physical localization framework for patients monitoring. *IEEE Systems Journal*, 11(1):118–127, 2015.

[45] MM Faniqul Islam, Rahatara Ferdousi, Sadikur Rahman, and Humayra Yasmin Bushra. Likelihood prediction of diabetes at early stage using data mining techniques. In *Computer Vision and Machine Intelligence in Medical Image Analysis*, pages 113–125. Springer, 2020.

[46] Jaime Ibarra Jimenez, Hamid Jahankhani, and Stefan Kendzierskyj. Health care in the cyberspace: Medical cyber-physical system and digital twin challenges. In *Digital twin technologies and smart cities*, pages 79–92. Springer, 2020.

[47] Prateek Joshi. *Artificial intelligence with python*. Packt Publishing Ltd, 2017.

[48] J Pradeep Kandhasamy and SJPCS Balamurali. Performance analysis of classifier models to predict diabetes mellitus. *Procedia Computer Science*, 47:45–51, 2015.

[49] Abdallah Karakra, Franck Fontanili, Elyes Lamine, and Jacques Lamothe. Hospit'win: a predictive simulation-based digital twin for patients pathways in hospital. In *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 1–4. IEEE, 2019.

[50] N Keshan, PV Parimi, and Isabelle Bichindaritz. Machine learning for stress detection from ecg signals in automobile drivers. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2661–2669. IEEE, 2015.

[51] Fedwa Laamarti, Hawazin Faiz Badawi, Yezhe Ding, Faisal Arafsha, Basim Hafidh, and Abdulmotaleb El Saddik. An iso/ieee 11073 standardized digital twin framework for health and well-being in smart cities. *IEEE Access*, 8:105950–105961, 2020.

[52] Asadullah Laghari, Zulfiqar Ali Memon, Sadiq Ullah, and Intesab Hussain. Cyber physical system for stroke detection. *IEEE Access*, 6:37444–37453, 2018.

[53] Li Li and Amir Ghasemi. Iot-enabled machine learning for an algorithmic spectrum decision process. *IEEE Internet of Things Journal*, 6(2):1911–1919, 2018.

[54] Ying Liu, Lin Zhang, Yuan Yang, Longfei Zhou, Lei Ren, Fei Wang, Rong Liu, Zhibo Pang, and M Jamal Deen. A novel cloud-based framework for the elderly healthcare services using digital twin. *IEEE Access*, 7:49088–49101, 2019.

[55] Roberto Martinez-Velazquez, Rogelio Gamez, and Abdulmotaleb El Saddik. Cardio twin: A digital twin of the human heart running on the edge. In *2019 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–6. IEEE, 2019.

[56] W Michael. Grieves digital twin: Manufacturing excellence through virtual factory replication-llc. 2014.

[57] J Jaime Miranda, Sanjay Kinra, Juan P Casas, G Davey Smith, and Shah Ebrahim. Non-communicable diseases in low-and middle-income countries: context, determinants and health policy. *Tropical Medicine & International Health*, 13(10):1225–1234, 2008.

[58] K Monisha and M Rajasekhara Babu. A novel framework for healthcare monitoring system through cyber-physical system. In *Internet of Things and Personalized Healthcare Systems*, pages 21–36. Springer, 2019.

[59] Mario A Muñoz, Laura Villanova, Davaatseren Baatar, and Kate Smith-Miles. Instance spaces for machine learning classification. *Machine Learning*, 107(1):109–147, 2018.

[60] Amril Nazir. Seamless automation and integration of machine learning capabilities for big data analytics. *International Journal of Distributed and Parallel Systems*, 8(3):1–18, 2017.

[61] Elisa Negri, H Davari Ardakani, Laura Cattaneo, Jaskaran Singh, Marco Macchi, and Jay Lee. A digital twin-based scheduling framework including equipment health index and genetic algorithms. *IFAC-PapersOnLine*, 52(10):43–48, 2019.

[62] Elisa Negri, Luca Fumagalli, and Marco Macchi. A review of the roles of digital twin in cps-based production systems. *Procedia Manufacturing*, 11:939–948, 2017.

[63] Mark J Nelson and Amy K Hoover. Notes on using google colaboratory in ai education. In *Proceedings of the 2020 ACM conference on innovation and Technology in Computer Science Education*, pages 533–534, 2020.

[64] Dijana Oreski, Stjepan Oreski, and Bozidar Klicek. Effects of dataset characteristics on the performance of feature selection techniques. *Applied Soft Computing*, 52:109–119, 2017.

[65] Carlotta Patrone, Gabriele Galli, and Roberto Revetria. A state of the art of digital twin and simulation supported by data mining in the healthcare sector. In *Advancing Technology Industrialization Through Intelligent Software Methodologies, Tools and Techniques*, pages 605–615. IOS Press, 2019.

[66] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[67] Nitin Pise and Parag Kulkarni. Algorithm selection for classification problems. In *2016 SAI Computing Conference (SAI)*, pages 203–211. IEEE, 2016.

[68] T Rachad, J Boutahar, et al. A new efficient method for calculating similarity between web services. *arXiv preprint arXiv:1501.05940*, 2015.

[69] Dattaraj Jagdish Rao and Shraddha Mane. Digital twin approach to clinical dss with explainable ai. *arXiv preprint arXiv:1910.13520*, 2019.

[70] Adil Rasheed, Omer San, and Trond Kvamsdal. Digital twin: Values, challenges and enablers. *arXiv preprint arXiv:1910.01719*, 2019.

[71] Luis F Rivera, Miguel Jiménez, Prashanti Angara, Norha M Villegas, Gabriel Tamura, and Hausi A Müller. Towards continuous monitoring in personalized healthcare through digital twins. In *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, pages 329–335, 2019.

[72] Aishwarya Roy, Anwesh Kumar, Navin Kumar Singh, and D Shashank. Stroke prediction using decision trees in artificial intelligence. 2018.

[73] Ensar Arif Sağbaş, Serdar Korukoglu, and Serkan Balli. Stress detection via keyboard typing behaviors by using smartphone sensors and machine learning techniques. *Journal of medical systems*, 44(4):1–12, 2020.

[74] Virginia Sandulescu, Sally Andrews, David Ellis, Nicola Bellotto, and Oscar Martinez Mozos. Stress detection using wearable physiological sensors. In *International work-conference on the interplay between natural and artificial computation*, pages 526–532. Springer, 2015.

[75] Virginia Sandulescu, Sally Andrews, David Ellis, Nicola Bellotto, and Oscar Martinez Mozos. Stress detection using wearable physiological sensors. In *International work-conference on the interplay between natural and artificial computation*, pages 526–532. Springer, 2015.

[76] Steven M Schwartz, Kevin Wildenhaus, Amy Bucher, and Brigid Byrd. Digital twins and the emerging science of self: Implications for digital health experience design and "small" data. *Frontiers in Computer Science*, 2:31, 2020.

[77] S Selvakumar, K Senthamarai Kannan, and S GothaiNachiyar. Prediction of diabetes diagnosis using classification based data mining techniques. *International Journal of Statistics and Systems*, 12(2):183–188, 2017.

[78] Wei Shengli. Is human digital twin possible? *Computer Methods and Programs in Biomedicine Update*, 1:100014, 2021.

[79] Omid Rajabi Shishvan, Daphney-Stavroula Zois, and Tolga Soyata. Incorporating artificial intelligence into medical cyber physical systems: A survey. In *Connected Health in Smart Cities*, pages 153–178. Springer, 2020.

[80] Orlando Simpson and Sergio G Camorlinga. A framework to study the emergence of non-communicable diseases. *Procedia computer science*, 114:116–125, 2017.

[81] Deepti Sisodia and Dilip Singh Sisodia. Prediction of diabetes using classification algorithms. *Procedia computer science*, 132:1578–1585, 2018.

[82] Sandeep K Sood and Isha Mahajan. A fog assisted cyber-physical framework for identifying and preventing coronary heart disease. *Wireless Personal Communications*, 101(1):143–165, 2018.

[83] K Sowjanya, Ayush Singhal, and Chaitali Choudhary. Mobdbtest: A machine learning based system for predicting diabetes risk using mobile devices. In *2015 IEEE International Advance Computing Conference (IACC)*, pages 397–402. IEEE, 2015.

[84] Feng-Tso Sun, Cynthia Kuo, Heng-Tze Cheng, Senaka Buthpitiya, Patricia Collins, and Martin Griss. Activity-aware mental stress detection using physiological sensors. In *International conference on Mobile computing, applications, and services*, pages 282–301. Springer, 2010.

[85] Lu-An Tang, Jiawei Han, and Guofei Jiang. Mining sensor data in cyber-physical systems. *Tsinghua Science and Technology*, 19(3):225–234, 2014.

[86] Fei Tao, Meng Zhang, Jiangfeng Cheng, and Qinglin Qi. Digital twin workshop: a new paradigm for future workshop. *Computer Integrated Manufacturing Systems*, 23(1):1–9, 2017.

[87] Isabel Voigt, Hernan Inojosa, Anja Dillenseger, Rocco Haase, Katja Akgün, and Tjalf Ziemssen. Digital twins for multiple sclerosis. *Frontiers in Immunology*, 12:1556, 2021.

[88] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[89] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[90] WHO. Non communicable diseases, mar 2020.

[91] Jun Zhang, Lin Li, Guanjun Lin, Da Fang, Yonghang Tai, and Jiechun Huang. Cyber resilience in healthcare digital twin on lung cancer. *IEEE Access*, 8:201900–201913, 2020.

[92] Xi Zheng, Min Fu, and Mohit Chugh. Big data storage and management in saas applications. *Journal of Communications and Information Networks*, 2(3):18–29, 2017.

Appendix A

Figures

A.1 Experiment Result from Published Contribution

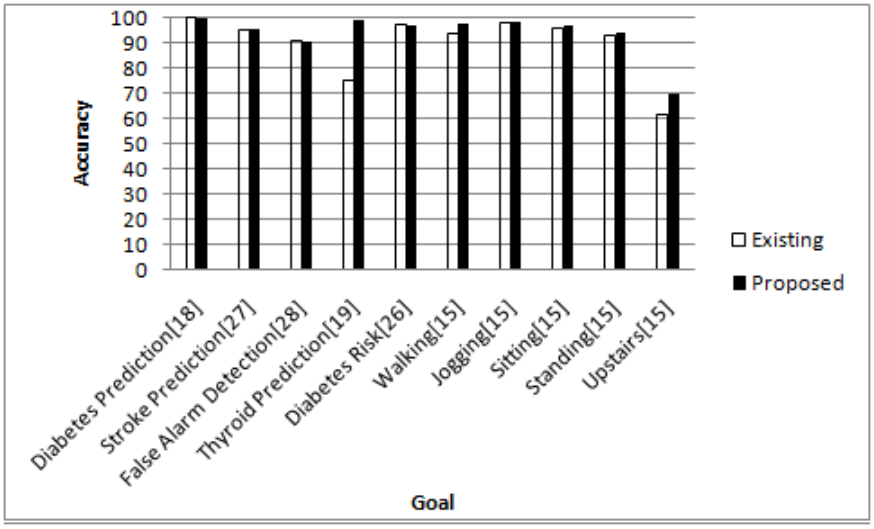


Figure A.1: Comparative analysis between existing work and proposed framework [43].

A.2 MLA Capabilities

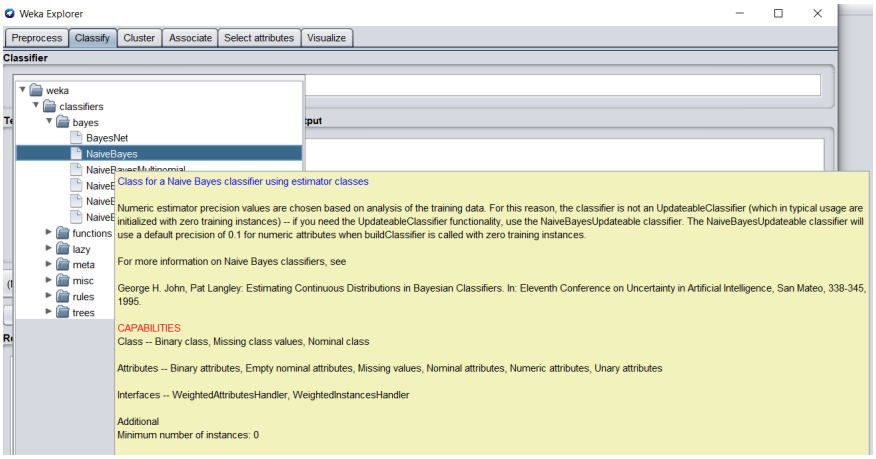


Figure A.2: Screenshot from WEKA.

Appendix B

Tables

B.1 Mental Stress Dataset Information

Table B.1: Collected Feature Details of the Mental Stress Dataset

No.	Feature Name	Data Type	Source
1	Time-stamps	Date-time	System Clock
2	Age	Categorical	Self-reported
3	Address	Categorical	GPS
4	Appetite	Categorical	Self-reported
5	Panic	neumeric	Smartwatch
6	Daily Phone usage	numeric	Phone
7	Incoming Calls	numeric	Phone
8	Outgoing Calls	numeric	Phone
9	Missed Calls	numeric	Phone
10	Longest Call duration	numeric	Phone
11	Social Media Active Time	Time	Social Media
12	Social Media Chat time	Time	Social Media
13	Favorite Social Media	String	Social Media
14	Favorite activity on social Media	String	Social Media
15	Heart rate	numeric	Smartwatch
16	Walking Time	numeric	Smartwatch
17	Exercise Time	numeric	Smartwatch
18	Inactive Time	numeric	Smartwatch
19	Sleeping Hour	numeric	Smartwatch
20	Stress	Binary	Mental healthcare provider

B.2 COVID Dataset information

Table B.2: Variable details of the whole dataset.

Variable Name	Short Description	Distinct	Type	Continuous/Categorical	Category
Date	Calling date	30	Date	Continuous	System
District	District in address of a patient	64	Nominal	Categorical	Details on Patient
Division	Division in address of a patient	8	Nominal	Categorical	Details on Patient
Upazilla	Upazilla in the address of a patient	406	Nominal	Categorical	Details on Patient
Disease	Specialized sector for a disease in medical.	5	Nominal	Categorical	Determined by Doctors
Category	Specialized sub-sector of a disease.	20	Nominal	Categorical	Determined by Doctors
Symptom	Diagnosis from the symptoms described by a patient.	773	Nominal	Continuous	Determined by Doctors
Gender	Gender of a patient	2	Nominal	Categorical	Details on Patient
Age	Age of a patient	100	Neumerical	Continuous	Details on Patient

B.3 Relationship between epidemiology factors of mental stress and IoT Technology

Table B.3: Summary of COVID-19 stressors and associated information

COVID-19 Stressor	**epidemiological fcators**	**Type of health data**	**Digital data to be collected**	**IoT Technology**	**IoT sensing type**
Fear of infection	Panic, sleep disturbance, anxiety	Behavioral, psychological	Sleep duration, heart rate variability (HRV)	Smartwatch	Wearable
Lockdown/ quarantine	Frustration, boredom	Social, behavioral, psychological	Communication pattern, activity, HRV	Smartphone, accelerometer, and gyroscope on smartphone	External, Social, wearable
Overloaded Information and Misinformation	Confusion, anxiety	Social, psychological	News, online content, HRV	API, software, HRV	External, Wearable, social and software.
Stigma	Unbalanced lifestyle	Behavioral	News, online content, activity	News, accelerometer, and gyroscope smartphone	Social, wearable
Unavailability of supply	Anxiety	Psychological	HRV	Smartwatch	Wearable
Loss of work or being jobless	Sleep disturbance, anxiety, irritability	Behavioral, psychological	Sleep duration, sleep pattern, HRV	Smartwatch	Wearable
Duration of quarantine	Anxiety	Psychological	HRV	Smartwatch	Wearable
Location of quarantine	Location	Behavioral	Location	Smartwatch or Smartphone	External

B.4 Relationship between epidemiology factors of diabetes and IoT Technology

Table B.4: Example of collecting epidemiological factors for early stage diabetes risk prediction

epidemiological factors		**IoT Technology**	**epidemiological factors**		**IoT Technology**
1.	Age	Apps	9.	Visual blurring	Apps
2.	Gender	Apps	10.	Itching	Sensor
3.	Polyuria	Apps	11.	Irritability	Sensor
4.	Polydipsia	Sensor	12.	Delayed healing	Apps
5.	Sudden weight loss	Sensor	13.	Partial paresis	Apps
6.	Weakness	Sensor	14.	Muscle stiffness	Apps
7.	Polyphagia	Sensor	15.	Alopecia	Apps
8.	Genital thrush	Apps	16.	Obesity	Sensor

B.5 Description of Knowledge Variables

Table B.5: List of variables

Definition of variable
Set of datasets, $D = \bigcup D_i$, where i = 1,2,3,...,n is an integer
Set of goals, $G = \bigcup G_t$, where t = 1,2,3,...,m is an integer and a goal G_t is an object of class *Goal*
Set of MLA, $A = \bigcup A_j$, where j = 1,2,3,...,o is an integer
Set of knowledge attributes of a particular dataset, K_{D_i}
Set of knowledge attributes of a particular goal, K_{G_t}
Set of knowledge attributes of a particular MLA, K_{A_j}
Collection of data knowledge, $K_D = \bigcup K_{D_i}$
Collection of goal knowledge, $K_G = \bigcup K_{G_t}$
Collection of MLA knowledge, $K_A = \bigcup K_{A_j}$