A Mobile Platform for Non-invasive Diabetes Screening

by

Shivani Chauhan

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2019

Author . . . .
Department of Electrical Engineering and Computer Science
August 30, 2019

Certified by . . . .
Richard R. Fletcher
Research Scientist, D-Lab
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

A Mobile Platform for Non-invasive Diabetes Screening

by

Shivani Chauhan

Submitted to the Department of Electrical Engineering and Computer Science
on August 30, 2019, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

The global burden of diabetes is profound, with over 400 million cases worldwide. Middle-income countries, such as those in Southeast Asia, are especially affected as diabetes becomes a major public health issue. In particular, India has a record number of people diagnosed with diabetes, 8.8% of the adult population. Despite the benefits of early prevention and treatment, awareness of the disease remains low, with health workers unable to keep up with the demands of the population. In addition to simply screening for diabetes, it is also important to assess a person's severity of diabetes so that the proper intervention and therapy can be delivered.

Current screening techniques rely on blood tests, often fingerprick blood glucose tests, but even though portable glucose meters are fairly inexpensive ($10 USD), they require a reliable stock of glucose test strips which are relatively costly ($0.05 USD) and are often in short supply. As a possible alternative for diabetes screening, our group has designed a mobile platform that integrates various non-invasive tests for diabetes including clinical questionnaires, thermal imaging, iris imaging, retina imaging and photoplethysmography. For the purposes of evaluation, we have defined six stages of diabetes progression: stage 0 (no diabetes), stage 1-3 (prediabetes), stage 4 (diabetes), and stage 5 (advanced diabetes).

The data collection and assessment platform includes an Android mobile application and a server to store and process the measurements and return the results to the Android client and web client. In this thesis, I describe the development of the server API for this platform, as well as the development of a Bayesian network model that is used to process the data and predict the specific stage of diabetes.

As a sample real-world implementation of this platform, our team has begun a large-scale diabetes study in two different sites in India, one in Mumbai and one in the Bangalore area. Based on data from this field study, I developed models for three different stages of diabetes pathogenesis: stage 0-3 (no diabetes to prediabetes), stage 4 (diabetes) and stage 5 (advanced diabetes). The performance of each model was evaluated using the area under the ROC curve (AUC). The best performing model was a Bayesian network that integrated questionnaire and iris data. Preliminary results for this model show high differentiation for each stage, with AUC scores of

1.0, sensitivity scores for each stage above .67 and specificity scores for each stage above .68. While data collection is still ongoing, these early results are encouraging and show a promising path for future large-scale diabetes screening.

Thesis Supervisor: Richard R. Fletcher
Title: Research Scientist, D-Lab

Acknowledgments

I would first like to acknowledge my advisor, Rich Fletcher. Rich's vision to positively impact the world is truly inspiring, as is the number of visionary projects that he executes through the Mobile Technology Lab. His goal to bring the learnings of traditional holistic medicine to Western medicine inspired me from our first meeting, and I am thankful to him for helping me apply my passions and technical interests to this project. From the Mobile Technology Group, I'd also like to thank John Mofor for his technical guidance and Madeline Zhang and Kobbie Ofori-Atta for their contributions.

I would also like to acknowledge our wonderful partners in India at S-VYASA and AJFTLE. Specifically, I would like to thank Dr. Ragavendrasamy Balakrishnan and Dr. Sureshbabu Venkatasamy for all their guidance and help on the project, and for being such accommodating hosts during my stays in India. At AJFTLE, I would like to thank Dr. Radhika Krishnan, Dr. Astha Jain, Ashwini Rogye, Harinder Virdi, and Swapnali Desai, who provided valuable contributions to the project. Our partners do inspiring work in the field of diabetes, and it was a truly great experience to visit and collaborate with them.

I would also like to thank the MIT TATA Center and the staff for their help with advice on the project and support during my travels to India. Chintan Vaishnav, who taught 15.S17 and 15.779, provided many helpful lessons geared towards troubleshooting problems and pushing the project forward. Diane Rigos was also an invaluable mentor, and I appreciated her check-ups on the project work.

Finally, I would like to thank my supportive family for their unconditional love throughout this process.

Contents

1 Introduction 15

1.1 Type 2 Diabetes . . . 15

1.2 Global Burden of Diabetes . . . 15

1.3 Prevalence of Diabetes in India . . . 16

1.4 Prevention and Treatment . . . 16

1.5 Scope of Thesis . . . 17

2 Diabetes Pathogenesis 19

2.1 Type 2 Diabetes Progression . . . 19

2.1.1 Stage 1 (Compensation) . . . 19

2.1.2 Stage 2 (Stable Adaptation) . . . 19

2.1.3 Stage 3 (Unstable Early Decomposition) . . . 20

2.1.4 Stage 4 (Stable Decomposition) . . . 20

2.1.5 Stage 5 (Severe Decomposition) . . . 20

2.2 Late-Stage Complications . . . 21

3 Tools and Methods for Diabetes Screening 23

3.1 Need for Screening Tools . . . 23

3.2 Existing Diabetes Diagnostic and Screening Tools . . . 24

3.2.1 Standard Blood Tests . . . 24

3.2.2 Limitations of Traditional Tests . . . 25

3.3 Existing Non-invasive Methods . . . 27

3.3.1 Clinical Questionnaires . . . 27

3.3.2 Retina Imaging 27

3.4 Emerging Non-invasive Methods 28

3.4.1 Thermal Imaging 28

3.4.2 Skin Fluorescence 29

3.4.3 Breath Analysis 29

3.4.4 Iris Imaging 30

4 Proposed Solution for Diabetes Screening 33

4.1 Prior Work in Mobile Technology Group 33

4.2 Selected Measurements 35

4.2.1 Diabetes Questionnaire 35

4.2.2 Thermal Imaging 36

4.2.3 Iris Imaging 37

4.2.4 Photoplethysmography (PPG) 37

4.2.5 Psychology Questionnaires 37

4.3 Diabetes Screening Server Platform 38

5 Server Platform 41

5.1 Architecture Overview 41

5.1.1 Background 41

5.1.2 Server Database 42

5.1.3 User Management 42

5.1.4 General API Design 42

5.2 Creating New Measurements 43

5.2.1 Add Measurement Database Models 43

5.2.2 Add Measurement API Call and Implementation 45

5.3 Creating Diagnoses 46

5.3.1 Database Models for Clinician Diagnosis 46

5.3.2 API Call and Implementation 47

5.4 Server Analysis Implementation 47

5.4.1 Database Models for Server Analysis 48

5.4.2 API Call and Implementation 48
5.4.3 Asynchronous Tasks 50
5.5 Viewing Measurements 51
5.6 Editing Measurement Labels 52
5.6.1 Database Models 53
5.6.2 API Call and Implementation 53

6 Server Platform Front-end 55

6.1 Overview 55
6.2 Clinician Portal 56
6.3 Patient Measurements 57
6.4 Patient Diagnosis 59
6.4.1 Adding Diagnosis 60
6.5 Bulk Download 61
6.5.1 Api Call and Implementation 63

7 Mobile Application 67

7.1 Android Libraries 67
7.2 Mobile Applications for Measurements 68
7.3 Measurement Analysis on Mobile Application 68

8 Computational Models for Assessing Diabetes Stage and Predicting Risk 71

8.1 Bayesian Network Algorithm with Questionnaire 71
8.1.1 Background and Prior Work 71
8.1.2 Overview 72
8.1.3 General Network Structure 72
8.1.4 Model Simplification 75
8.1.5 Design and Implementation 75
8.2 Iris Analysis Algorithm 80
8.2.1 Background and Prior Work 80

8.2.2 Multiclass Classification 80
8.3 Bayesian Network with Diabetes Questionnaire and Iris Analysis . . . 81
8.3.1 Modified Symptoms Layer 82

9 Field Study Description and Preliminary Results 83
9.1 Study Design 83
9.2 Standards for Model Training and Validation 86
9.2.1 Blood Glucose Measurements 86
9.2.2 Retina Imaging 86
9.3 Study Protocol 87
9.4 Participant Demographics 88
9.5 Bayesian Algorithm with Diabetes Questionnaire 89
9.5.1 Preliminary Results 90
9.5.2 Sensitivity Analysis 90
9.6 Iris Algorithm Analysis 94
9.6.1 Initial Iris Analysis 94
9.6.2 Preliminary Results 94
9.6.3 Discussion 96
9.7 Bayesian Algorithm with Iris Images and Questionnaire 97
9.7.1 Preliminary Results 97
9.7.2 Sensitivity Analysis 99
9.7.3 Discussion 101

10 Conclusion 103
10.1 Contributions of Work 103
10.1.1 Data Collection and Assessment Platform 103
10.1.2 Algorithms to Classify Diabetes Stage Classification 103
10.2 Future Work 104
10.3 Larger Impact 105

List of Figures

2-1 Pathogenesis of diabetes from stage 1 to stage 5 [1] 21

4-1 The device used to capture iris images alongside hood attachment [2] 34

4-2 Thermal Imaging Measurement 36

4-3 Sample Iris Images 37

4-4 Diabetes Screening Platform Architecture 39

5-1 DiagnosticMeasurement with BloodTestQuestionnaire and DiagnosticMeasurementMetaData 44

5-2 ServerDiagnosticPrediction, AbstractDiagnosticPredictions and IrisScannerResult 49

6-1 Clinician Portal 57

6-2 Patient Measurements 58

6-3 View Diagnosis Page 60

6-4 Add Diagnosis Page 61

6-5 Add Diagnosis Modals 62

6-6 Download Measurements Form 64

6-7 Download Measurement Zip File Organization 65

7-1 Screen Captures of Measurement and Container Mobile Applications 69

7-2 Results Page of Questionnaire Mobile Apps 70

8-1 Three-Layer Generalized Bayesian Model [3] 73

8-2 Bayesian Model with Non-Diabetic Stage (Stage 0) and Fives Stages of Diabetes 74
8-3 Bayesian Network for Stage 0-3, 4, and 5 of Diabetes 76
8-4 Bayesian Network with Iris Classification for Stage 0-3, 4, and 5 of Diabetes 81

9-1 Data Collection in Mumbai 85
9-2 Retina Measurement 87
9-3 ROC curve for Bayesian Network 91
9-4 Region 18 Iris Features vs. RBS Values 95
9-5 Region 4 Iris Features vs. RBS Values 95
9-6 Mean ROC curve for Region 21 from Iteration of Randomly Selected Test Set 97
9-7 ROC curve for Bayesian Network with Iris Classification 98

List of Tables

5.1 Query Parameters for View APIs 52

8.1 Risk Factors for Bayesian Network 77

8.2 Prior Probabilities for Risk Factors 78

8.3 Prior Probabilities for Diabetes Stages 78

8.4 Symptoms for Bayesian Network 79

8.5 Additional Symptoms for Bayesian Network with Iris Classifier 82

9.1 Characteristics of subjects 89

9.2 Number of Subjects in Stages of Diabetes 89

9.3 Mean AUC, Sensitivity, and Specificity Values for Bayesian Network . 90

9.4 Cross-Entropy Values of Stage 0-3 92

9.5 Cross-Entropy Values of Stage 4 93

9.6 Cross-Entropy Values of Stage 5 93

9.7 Mean AUC, Sensitivity, and Specificity scores for Region 4 96

9.8 Mean AUC, Sensitivity, and Specificity scores for Region 18 96

9.9 Mean AUC, Sensitivity, and Specificity scores for Region 21 96

9.10 Mean AUC, Sensitivity, and Specificity Values for Bayesian Network with Iris Classification 98

9.11 Cross-Entropy Values of Stage 0-3 99

9.12 Cross-Entropy Values of Stage 4 100

9.13 Cross-Entropy Values of Stage 5 100

Chapter 1

Introduction

1.1 Type 2 Diabetes

Diabetes mellitus is a disease where the body is not able to properly use insulin, which is needed to transport glucose, the cell's source of energy, from the bloodstream into the cells [4]. This causes the cells to be starved for fuel. In addition, when glucose is not able to enter cells, it builds up in the bloodstream which can lead to problems with the heart, eyes, nerves or kidneys [4]. In type 2 diabetes, the most common type of diabetes, the pancreas produces insulin, but it either does not produce enough insulin or insulin resistance prevents normal function [5].

1.2 Global Burden of Diabetes

Diabetes is a worldwide problem, with diabetes prevalence rising more rapidly in low and middle income countries. According to the World Health Organization, 1.6 million deaths in 2016 were directly caused by diabetes and the global prevalence of diabetes in adults has risen from 4.7% in 1980 to 8.5% in 2014 [6].

The rising prevalence has worrying implications for the global cost of diabetes, which is calculated from the direct costs of medical care as well as indirect costs incurred through loss of productivity or earnings. By 2030, the global cost of diabetes is projected to substantially rise to a maximum of 2.2% of global GDP [7]. Low and middle income countries with the highest rates of prevalence of diabetes are the greatest contributors to the global cost. In particular, India is projected to see the largest increase in the absolute number of people with diabetes [8].

1.3 Prevalence of Diabetes in India

India has a record number of people suffering from type 2 diabetes, with over 73 million cases of diabetes in India in 2017, or 8.8% of the adult population, with higher rates of prevalence in South India [9]. Genetic predisposition, lifestyle factors, and an overburdened healthcare system all contribute to the high rate of diabetes in India.

Though there is strong evidence that genetic disposition and greater insulin resistance contribute to the rate of diabetes in India, lifestyle changes are also responsible [10]. Industrialization and migration to urban cities from rural areas has led to increased obesity rates due to physical inactivity and poor diets [10]. The rise of fast, relatively cheap foods across income brackets has meant that both poor and wealthy communities are increasingly prone to the disease [11].

Though diabetes prevalence has increased in all states across India over the last quarter century, an Indian multistate study reported that 47% of diabetes cases in the population went undiagnosed [12]. One contributing issue is that the number of people in India far outweighs the number of available health workers and health professionals [13]. The most vulnerable people are those with low incomes who are unable to receive the early detection and awareness that could help them make necessary lifestyle modifications in the face of this medical condition.

1.4 Prevention and Treatment

Appropriate diabetes treatment and management is determined based on an individual's degree of diabetes. Early stage preventions for prediabetes and diabetes are preferred because current methods of treating diabetes do not prevent all complications associated with the condition [14]. Eating well, exercising and maintaining a healthy diet are encouraged because they are cost-effective and generally safe lifestyle modifications that have been shown to have a positive impact on glycemic control [15].

In addition to lifestyle modifications, there also exist other pharmacologic and surgical treatments for diabetes, which are shown to decrease the incidence of diabetes but not as much as lifestyle modifications [14]. Though all these treatments prevent complications of diabetes, there is no cure for type 2 diabetes and blood glucose levels are not usually restored [14]. And if the condition is more severe or if complications arise, specific treatment, diabetes medications or insulin therapy may be required [5].

1.5 Scope of Thesis

The contents of this thesis are organized as follows. In Chapter 2, I present a detailed multi-stage description of diabetes. In Chapter 3, I motivate the need for a diabetes screening tool and describe current and emerging screening methods. Then, I describe our proposed solution and its implementation in Chapters 4 through 8. In Chapter 9, I present the preliminary results from our data collection and modeling platform based on an ongoing field study. In Chapter 10, I discuss contributions of this work and final conclusions.

Chapter 2

Diabetes Pathogenesis

2.1 Type 2 Diabetes Progression

The progression of diabetes is characterized by abnormal beta cell function [16]. Beta cells are unique pancreas cells that produce, store and release insulin [17]. The progression of diabetes can be classified into five stages [16]. The first three stages are known as prediabetes, the fourth stage is diabetes, and the fifth stage is advanced diabetes. Figure 2-1 gives a visual representation of the five stages of diabetes, which are described below.

2.1.1 Stage 1 (Compensation)

At this stage, the first signs of insulin resistance are present but beta cells are able to compensate for the resistance by producing higher levels of insulin, by increasing in either number or size [16]. Individuals in this stage exhibit symptoms of hyperinsulinemia and prediabetes, including weight gain, strong cravings for sugar, intense or frequent feelings of hunger, anxiety, fatigue, and lack of motivation [18].

2.1.2 Stage 2 (Stable Adaptation)

In this stage, blood glucose levels rise above normal levels as beta cell activity decreases, producing less glucose-stimulated insulin secretion (GSIS) [16]. This stage is stable with individuals usually evading progression to diabetes for years as long as the amount of insulin production is sufficient to prevent sharp rises in blood glucose levels [16]. In this stage, early symptoms of hyperglycemia may begin to appear, such as increased thirst or hunger, frequent urination, headache, blurred vision and fatigue [19].

2.1.3 Stage 3 (Unstable Early Decomposition)

This final stage of diabetes is marked by a rapid increase in blood glucose as the beta cell decline passes a critical point [16]. Though this stage is often transient, an individual may experience the same symptoms of hyperglycemia as described in stage 2, but to a stronger degree. The presence of long-term hyperglycemia may also be present, such as nausea and vomiting, shortness of breath, dry mouth, muscular weakness, and abdominal pain [19].

2.1.4 Stage 4 (Stable Decomposition)

In this stage, which marks the beginning of diabetes disease, an individual is nearing beta cell failure, but the beta cells produce enough insulin to avoid ketoacidosis [16], in which the body must rely on ketone bodies from fat for energy. The main symptoms of diabetes appear in this stage: tingling, numb or painful sensations in the hands or feet, slow healing of cuts and wounds, patches of darkened skin, itchy skin/yeast infections, and symptoms of hyperglycemia [20].

2.1.5 Stage 5 (Severe Decomposition)

This stage is a more advanced form of the disease in which various diabetes-related disorders may manifest. Individuals often become ketotic, meaning that their beta cell levels have decreased to a point where outside sources of insulin are necessary for glucose-based energy production [16]. Along with the symptoms of diabetes, an individual could experience advanced complications. The complications resulting from stage 5 of diabetes are explored more in the next section.

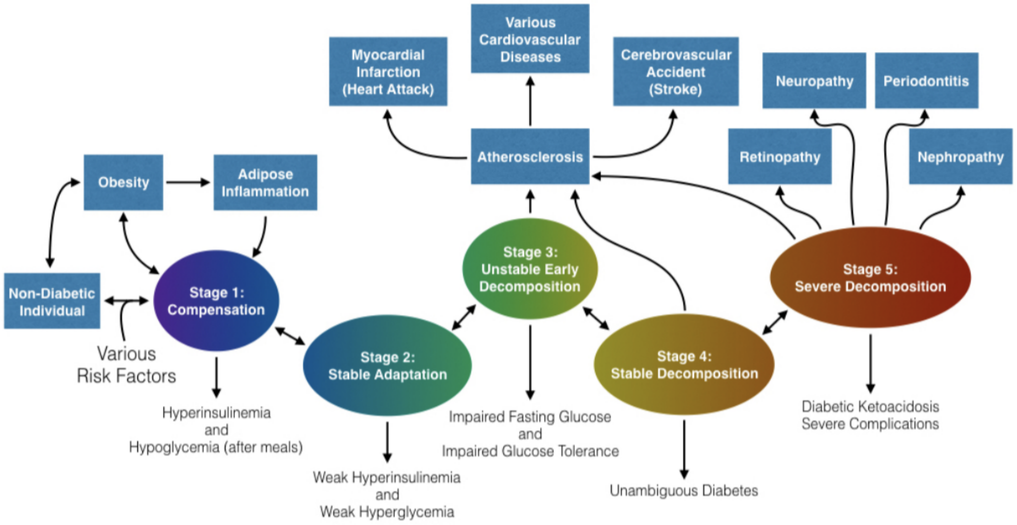

Myocardial Infarction (Heart Attack)
Various Cardiovascular Diseases
Cerebrovascular Accident (Stroke)
Neuropathy
Periodontitis
Atherosclerosis
Retinopathy
Nephropathy
Obesity
Adipose Inflammation
Non-Diabetic Individual
Stage 1: Compensation
Various Risk Factors
Stage 3: Unstable Early Decomposition
Stage 5: Severe Decomposition
Stage 2: Stable Adaptation
Stage 4: Stable Decomposition
Hyperinsulinemia and Hypoglycemia (after meals)
Impaired Fasting Glucose and Impaired Glucose Tolerance
Diabetic Ketoacidosis Severe Complications
Weak Hyperinsulinemia and Weak Hyperglycemia
Unambiguous Diabetes

Figure 2-1: Pathogenesis of diabetes from stage 1 to stage 5 [1]

2.2 Late-Stage Complications

Late-stage complications of diabetes include cardiovascular disease, nephropathy, retinopathy, neuropathy, and periodontitis [5]. Prolonged hyperglycemia is the primary factor associated with most of these complications. Macrovascular complications, such as atherosclerosis and stroke, are accelerated in the presence of diabetes, and are a growing epidemic in the type 2 diabetic population [21].

High blood sugar levels in diabetics can also cause microvascular damage to the retina, kidneys and nerves resulting in the complications of diabetic retinopathy, diabetic nephropathy and diabetic neuropathy respectively. Diabetic retinopathy is microvascular disease that can cause damage to the blood vessels and retina of the eyes [22]. The too-high levels of glucose in the blood damage the small blood vessels in the retina [23]. In diabetic nephropathy, the kidneys are damaged and fail to function, and in diabetic neuropathy, the somatic and autonomic peripheral nerves are damaged, causing reduced blood supply to the peripheries and nerve damage [22].

Chapter 3

Tools and Methods for Diabetes Screening

3.1 Need for Screening Tools

Screening tests are important in early detection of diabetes to prevent subsequent complications and healthcare costs associated with more advanced stages of the disease. Screening tests differ from diagnostic tests, which are offered to aid in the diagnosis of a suspected disease or condition, whereas screening tests are offered to large numbers of asymptomatic people who may be at high risk for a disease, in order to guide further diagnostic tests if needed [24]. Therefore the cutoffs tend towards high sensitivity because false positives are acceptable, particularly if the test is inexpensive and not harmful [24].

In the case of diabetes, determining which stage of the disease an individual is in is important in order to prescribe the appropriate treatments. At earlier stages of diabetes, lifestyle modifications such as healthier diet and exercise choices will be more effective. As the disease progresses, however, the treatments will have to be modified accordingly.

3.2 Existing Diabetes Diagnostic and Screening Tools

3.2.1 Standard Blood Tests

Several standard methods exist to diagnose diabetes. These tests primarily consist of measuring blood glucose levels. According to the Mayo Clinic, the most common tests include [25]:

- **Glycated hemoglobin test (A1C)**: This test does not measure the instantaneous glucose levels but rather measures the average level of glucose over a longer period of time: two or three months. It requires the results from two separate tests in order to determine whether a person has diabetes or prediabetes. An A1C between 5.7% to 6.5% indicates prediabetes, while an A1C value higher than 6.5% indicates diabetes [26].

- **Random blood sugar (RBS) test**: This test uses a blood sample taken at a random time to measure blood sugar level. A blood glucose level between 140-200 mg/dl indicates prediabetes and a result greater than or equal to 200 mg/dl indicates diabetes [27].

- **Fasting blood sugar (FBS) test**: This test uses a blood sample taken after an overnight fast and also looks at two separate tests to determine prediabetes or diabetes. A value between 100-125 mg/dl indicates prediabetes and a value of 126 mg/dl or higher indicates diabetes [26]. More specifically, stage 2 fasting glucose levels are around 89-130 mg/dl, stage 3 fasting levels rise rapidly from 130 mg/dl to the more stable stage 4 levels of 285-350 mg/dl and stage 5 glucose levels exceed 350 mg/dl [16].

- **Postprandial Glucose (PPG) test**: This test measures plasma glucose levels after eating. A two hour post-prandial value of 140-199 mg/dl indicates prediabetes, while a value of 200 mg/dl or more indicates diabetes [28].

When these tests are performed in a clinical setting, the blood drawn is from the vein in order to measure plasma glucose level for accuracy. However, in the past

30 years, low-cost finger prick methods have been developed that allow patients to frequently self-monitor their glucose levels. The devices, called glucometers, require a small amount of capillary blood usually from a finger, to be placed on a test strip. There it reacts with an enzyme and induces an electrical current and subsequent color change in the strip that is proportional to the amount of glucose in the sample; the color change can then measured using reflectance photometry [29].

Since this method of measuring blood glucose level can undertaken at any time of day, does not require blood from a vein, and can be performed by anyone, it is being explored as an alternative to blood tests to conduct mass screenings for diabetes in India [30]. Important to note is that these random capillary blood glucose (RCBG) tests are less accurate than lab tests performed on blood drawn from the vein. One reason is because RCBG tests measure whole blood glucose levels, which are usually 10-15% lower than plasma glucose alone [29]. We explore more challenges in the next section.

3.2.2 Limitations of Traditional Tests

As discussed in the previous section, although low-cost blood glucose testing methods consisting of glucometers and test strips are available, a major problem of access still exists in low-resource areas in developing countries. Four main problems are:

- **Lack of supply chain for test strips**: The cost of test strips continues to be problematic, with a box of 100 plastic or paper strips costing about 1,600 Indian rupees or $25 USD. Therefore most health clinics in low-resources areas do not have a ready supply of these strips and no reliable means of restocking such costly items [31].

- **Lack of health infrastructure**: Without ready access to test strips, some health organizations in developing countries opt for drawing blood in order to determine blood glucose levels. However, the Indian healthcare infrastructure has many problems—public health facilities are inefficient, inadequately managed and staffed and have poorly maintained medical equipment [32]. Therefore, a significant number of blood samples are mislabeled, lost or generally unable to be traced back to the original owner, even if results are eventually determined.

- **Risk of blood-borne infections**: A major drawback of diabetes screening associated with both venous blood tests and finger prick blood tests is the risk of transmitting blood-borne infections. In low-resource areas in low and middle income countries, conditions such as hepatitis and HIV are valid concerns [33].
- **Lack of diabetic stage specificity**: Although measuring RBS is efficient in terms of both time and cost, the measurement is not able to differentiate which stage of diabetes the patient is in since it is a less accurate test than a whole blood glucose test [29].

In light of these problems, an urgent need remains to create a diabetes screening solution that can meet the following requirements:

- **Low cost**: The solution is provided at little to no cost and does not require recurring supplies to be accessed in order to be effective.
- **Non-invasive**: The screening test should not require any blood to be drawn in order to avoid the risk of blood-borne infections.
- **Real-time results**: The patient should receive near immediate results before they leave the clinic, thus reducing the organizational overhead required to contact the patient at a later date.
- **EMR integration**: The test should allow integration into an existing electronic medical records system so that the results can be stored and shared with clinics and the health care system.
- **Stage placement**: The solution should be able to differentiate which stage of diabetes the individual is currently in. This is important because diabetes treatment will vary depending on the individual's severity of the disease.

3.3 Existing Non-invasive Methods

3.3.1 Clinical Questionnaires

Compared to other assessments of disease risk, questionnaires are usually low-cost, brief and easy to administer and score. However the normative and baseline values for these questionnaires often rely on population-specific information, such as average waist or hip circumference, in order to make recommendations. In ethnically diverse nations, such as the US, this can pose a challenge. Indeed, national guidelines within the US disagree on who should be screened for diabetes, and no existing diabetes risk score is highly generalizable or widely followed [34]. In India, however, a general questionnaire, the Indian Diabetes Risk Score (IDRS), was developed at the Madras Diabetes Research Foundation to assess diabetes risk within a more homogenous population. The questionnaire evaluates an individual in four categories: age, family history, abdominal obesity and physical activity, in order to issue a final risk score [35]. The study found that an IDRS value of greater than or equal to 60 detected undiagnosed diabetes with a positive predictive value of 17%, negative predictive value of 95.1% and accuracy of 61% [35].

The value of such a test is as a first, low-cost round of screening, where only the population that scores above a certain value, such as 60, will need to be screened a second time with more comprehensive tests. The test aims for a high negative predictive value, ensuring that subjects with a negative result truly do not have the disease. Despite these benefits, the IDRS is a relatively simplistic evaluation of information, with a straightforward scoring guideline. In Chapter 7, we attempt to extract more information from a subject's questionnaire answers by using a Bayesian algorithm.

3.3.2 Retina Imaging

For late stage diabetes, one of the conventional non-invasive tools is fundus or retinal imaging. Digital retina imaging uses high-resolution imaging to allow medical professionals to assess the health of the retina and detect eye and health conditions such as glaucoma, macular degeneration and diabetes. Advanced diabetes, in particular, can be detected through signs of diabetic retinopathy, though the disease only develops when an individual is in more advanced stages of diabetes. Mobile retina tests have many qualities that make them desirable for large-scale population screening: they are non-invasive, portable and provide immediate results. Companies already utilizing retina tests include Peek Vision, which provides smartphone-based vision screening to individuals in low-resource areas [36]. Another company, Eyenuk, harnesses deep learning to develop an automated cellular-based screening device for diabetic retinopathy, though its market is domestic [37]. Although retina scanning as a technology has been around for over 10 years, with many organizations like those previously mentioned developing portable retina scanners for mobile phones, the specialized hardware required has caused some hurdles in wider adoption and application of the technology.

3.4 Emerging Non-invasive Methods

To meet the need for non-invasive diabetes screening tools, a few technologies have been explored in academic research in the fields of skin fluorescence, thermal scanning, breath analysis, iris scanning and retinal scanning, with some of the research currently being used to create commercial products. These technologies are described in this section.

3.4.1 Thermal Imaging

A main advantage of thermal scanning is that it requires no contact with a patient. It captures the natural IR radiation from a body to view the distribution of heat on certain areas of the body. Thermal imaging studies have been undertaken to study a variety of diseases, such as inflammatory diseases, which can affect the surface temperature of the human body. In the case of diabetes mellitus, circulatory dysfunction can occur in the hands and feet. This phenomenon can be illustrated with a cold challenge test, in which a subject submerges a thinly-gloved hand into cold water for 1 minute or longer and then images of the hand are taken as it undergoes thermal recovery [38]. A delayed recovery, where the fingers may be colder than before immersion may indicate diabetes [38]. Several clinical studies have shown abnormalities in thermal patterns in the hands or feet of diabetes patients, but it has not been show how or if this method can be implemented in a global health setting to screen for diabetes.

3.4.2 Skin Fluorescence

Skin autofluorescence is a non-invasive technique that illuminates an area of the skin surface about 4 cm^2 on the volar side of the forearm with an excitation light source and measures the natural emitted light with a spectrometer with a glass fiber. Diabetes research is concerned with how skin autofluorescence is affected by tissue accumulation of advanced glycation end products (AGEs). AGEs are proteins and lipids that have become glycated due to the increased number of glucose sugar molecules in the bloodstream [39]. The AGE reader and Diab-spot are commercialized products from the company Diagnoptics that diagnose diabetes using this technology, specifically by shining UV light onto the forearm to calculate autofluorescence [40]. Once the device captures the measurement, it runs the data through an algorithm, such as a decision-tree, to provide a risk rating for each tested individual [40]. Currently, the technology developed by Diagnoptics is not portable, and is only intended for urban hospitals in its markets in the Netherlands, Canada and the US. However, there is an opportunity to create a low-cost version of this technology for use with mobile phones in low-resource areas [41].

3.4.3 Breath Analysis

Diabetic ketoacidosis (DKA) is a serious complication that occurs in stage 5 of diabetes when the the body does not produce enough insulin, forcing the liver to break down fat as fuel. Ketone bodies are produced when the liver metabolizes fatty acids, including acetone, β-hydroxybutyrate and acetoacetic acid [42]. Detection of ketone bodies can be done though analysis of urinary ketone concentrations and measuring the concentration of serum β-hydroxybutyrate in the blood, but there has been interest in creating a non-invasive tool to measure the concentration of acetone in the breath to guide therapeutic interventions [42]. The most common techniques to measure breath acetone include: gas chromatography techniques, mass spetrometry-based techniques, electrochemical sensors and laser spectroscopy techniques [43]. In particular, electrochemical sensors are both cost-effective and portable, and various sensors have been developed with the purpose of improving the sensitivity and selectivity of measurements [43].

However, the research in this area has shown mixed results. For example, a 2018 study measuring breath acetone concentrations found that the ratio of breath acetone at 1 hour to the initial breath acetone was found to be significantly different for subjects with type 2 diabetes as compared to the control and impaired glucose intolerance subjects [44]. But the study ultimately concluded that at least in their cohort, breath acetone could not be used as a potential screening diagnostic for type 2 diabetes due to individual variations in insulin deficiency and resistance in the diabetic and prediabetic patients [44].

A larger review on 41 independent breath acetone studies conducted in the last half century found that while elevated mean breath acetone concentration was a statistically common observation in type 1 diabetes, there was no such equivalent observation for type 2 diabetes [43]. In addition, there was no correlation between breath acetone and blood glucose level [43]. More research is needed in this field to tackle the challenging questions of finding clear quantitative correlations between type 2 diabetes and breath acetone concentrations.

3.4.4 Iris Imaging

Outside of academic research, prior works in the field of alternative medicine has reported the ability to diagnose certain diseases using non-invasive methods. One of these methods, which our group at MIT discovered through our relationship with the

Indian yoga school Swami Vivekananda Yoga Anusandhana Samsthana (S-VYASA), is the use of the iris to screen for diabetes.

Iridology is the practice of mapping specific regions of the iris to various organs and tissues in the human body [45]. Although the use of iris analysis for health diagnostics has existed for over 100 years, the method has been highly controversial and attracted a great deal of skepticism from the medical community [2]. However, lack of data and large-scale experiments have prevented proper study and analysis of iridology. A review of iridology studies performed in 2000 found only four controlled and masked experiments performed, with some having too-small sample sizes [46]. Some recent studies have shown more promise in using this method of diagnosis for diabetes, including one performed by the researchers at S-VYASA [47].

Chapter 4

Proposed Solution for Diabetes Screening

In this chapter, I describe our proposed solution for robust mobile screening of diabetes. This screening and analysis platform is designed to be used by health workers and clinicians to screen for diabetes in low-resource and rural places in India. The mobile application employs non-invasive measurements, uploads them to a cloud server and then analyzes these measurements to predict diabetes and assess severity.

4.1 Prior Work in Mobile Technology Group

Our team at the Mobile Technology Group at MIT D-Lab has demonstrated encouraging results in the area of non-invasive methods by using iris imaging, thermal imaging and skin fluorescence to diagnose specific health conditions. This work is detailed below.

A previous MEng student in Dr. Fletcher's Mobile Technology Group, Tania Yu, implemented a system to analyze iris images and return a diagnostic for diabetes. The iris images were collected from participants in a study clinic in India. A webcam connected to a mobile device with an accompanying hood attachment, as illustrated in Figure 4-1, was used to take the images. Then the regions of the eye were segmented and non-linear and linear classifiers were used to classify the images and determine whether or not the patient was at risk of developing diabetes [2]. Although the dataset in this project was limited, and therefore strong conclusions could not be drawn, the results were promising, with the predictions performing well above the baseline of random guessing.

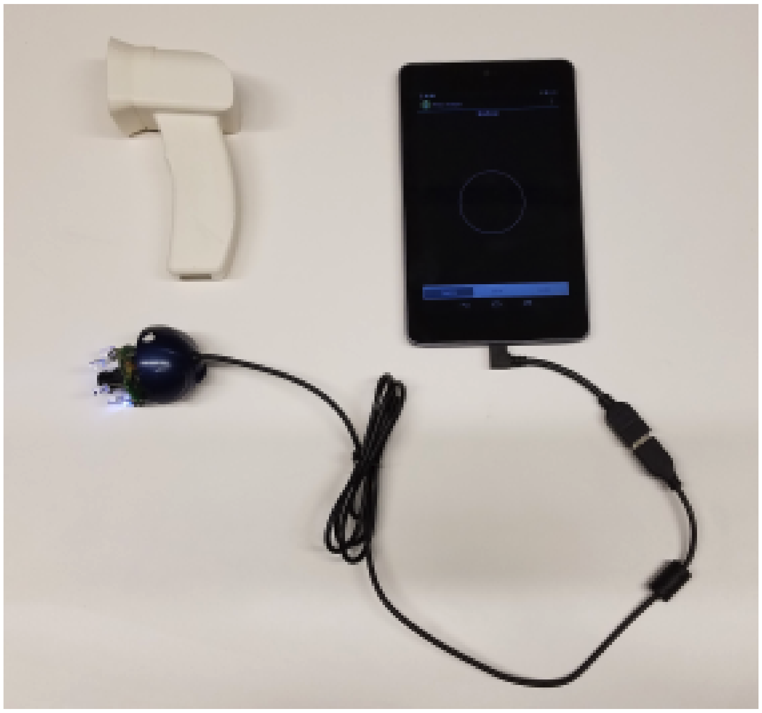


Figure 4-1: The device used to capture iris images alongside hood attachment [2]

Another student in our group, Lasya Thilagar, also conducted preliminary studies for diabetes screening; she used thermal imaging on a mobile phone. Images of the face were taken by a Seek Thermal Camera that connected to a mobile phone through a USB port. Thilagar plotted nose, cheek, forehead, nose-cheek, nose-forehead and cheek-forehead temperatures for individuals with and without diabetes. Again, although the dataset was limited in the number of individuals studied, the nose-cheek temperature plot seemed to show a clear difference in the temperature patterns between individuals with and without diabetes [48].

In the area of skin fluorescence, another student in our D-Lab group, Peniel Argaw, conducted early experiments using a simple attachment to a mobile phone. The low-cost 3D-printed device implemented the basic principles of a spectrometer in conjunction with a mobile phone to measure AGE fluorescence [41]. The device was tested in clinical trials in India on 32 patients with and without diabetes and showed promising results, though it acknowledged the role of skin tone and anti-diabetic medicine as factors of error [41].

4.2 Selected Measurements

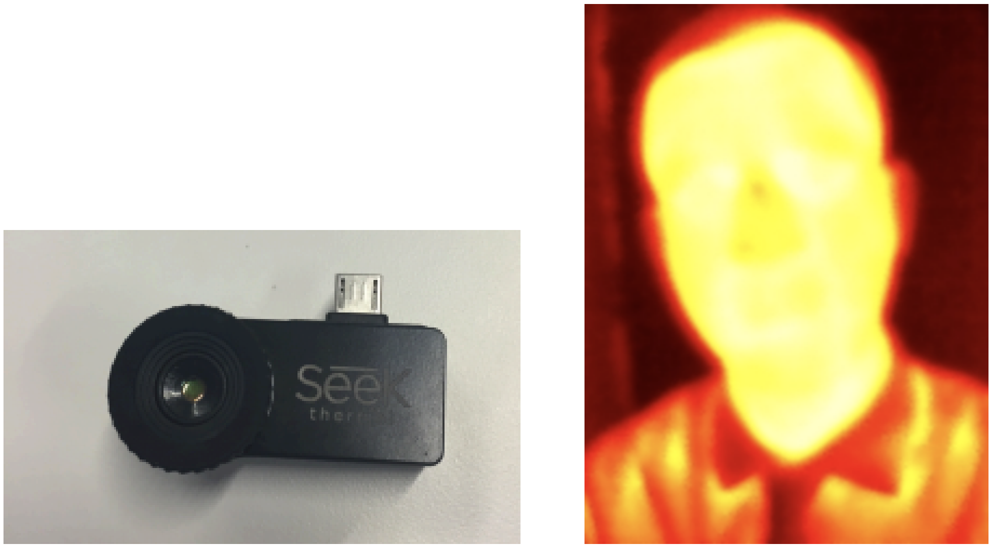
For our mobile platform, we have chosen specific non-invasive measurements, including questionnaires, images of the iris and retina, a thermal scan of the face, and a vision test. The full list of non-invasive measurements is detailed in the following sections.

4.2.1 Diabetes Questionnaire

The first non-invasive measurement is an extended version of the standard screening questionnaire known as the Indian Diabetes Risk Score (IDRS). As discussed in Section 3.3.1, the study uses four categories: age, family history, abdominal obesity and physical activity. Our questionnaire starts with these categories but has several more questions to assess additional diabetes risk factors, such as alcohol usage and stress level, and diabetes symptoms such as experiencing fatigue and numbness in limbs. We also take some standard measurements as part of the diabetes questionnaire such as height, weight and blood pressure. The diabetes questionnaire was developed in collaboration with our partners at S-VYASA.

4.2.2 Thermal Imaging

In order to capture thermal images, we are using the mobile phone-based thermal camera module (Seek compact) manufactured by Seek Thermal, which is relatively low-cost ($200 USD) and has a pixel size of 206x156 [49]. Our group has previously developed a custom Android mobile application for this module using the Seek Thermal Android SDK. This application not only captures a thermal image in the form of a JPEG, but also records a temperature array in CSV format. The Seek thermal camera and a sample thermal face image are shown in Figure 4-2.



(a) Seek Thermal Camera

(b) Seek Face Thermal Image

Figure 4-2: Thermal Imaging Measurement

4.2.3 Iris Imaging

The device in Figure 4-1 is used collect images of the iris. The webcam inside the hood is a Gear Head 5.0 MP webcam with a maximum resolution of 1280 x 1024. Infared LEDs are used to illuminate the iris and a low-pass visible light filter blocks visible wavelengths. Two sample iris images are displayed in Figure 4-3.

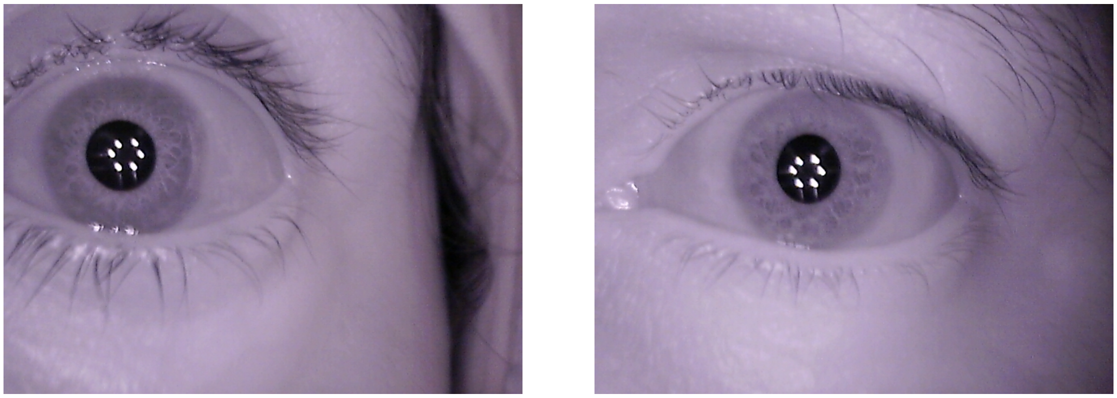


Figure 4-3: Sample Iris Images

4.2.4 Photoplethysmography (PPG)

Another non-invasive measurement that we are employing is photoplethysmography. Using the built-in camera and light of an Android phone, it is possible to capture a photoplethysmographic waveform without any external device. This waveform represents the blood volume pulse in the finger, which can then be analyzed using pulse wave analysis. The morphology of the pulse-waveform has been shown to be an early indicator of changes in the cardiovascular system, which may occur in the initial stages of diabetes pathogenesis [50].

4.2.5 Psychology Questionnaires

In order to explore pyschological risk factors that might contribute to diabetes, we have also included several validated psychology questionnaires to assess specific factors including anxiety, depression, sleep quality, and perseverative thinking. We added these questionnaires in consultation with the researchers at S-VYASA to do a more thorough evaluation of psychological parameters that have been identified as important risk factors of diabetes and cardiometabolic syndromes. The standard questionnaires we have selected are the following:

- **GAD-7 (General Anxiety Disorder-7)**: for assessment of anxiety
- **PHQ-9 (Patient Health Questionnaire-9)**: for assessment of depression
- **PTQ-15 (Perseverative Thinking Questionnaire-15)**: for assessment of perseverative thinking
- **PSQI (Pittsburgh Sleep Quality Index)**: for assessment of sleep

4.3 Diabetes Screening Server Platform

After the measurements are taken with the Android mobile applications and accompanying diagnostic devices, they are uploaded to a cloud server by calling one of several different Add Measurement APIs, described further in Section 5.2.1. The measurements are stored anonymously and securely in the server until a health worker requests to view or label the measurements. These actions are also performed by calling server APIs from either the mobile application or the website front-end. In addition, a health worker can request an analysis of the measurements from the mobile application which will then return back the results of applying algorithms on the individual measurements, and also an overall diabetes risk score that is analyzed from all the measurements. This architecture is shown in Figure 4-4.

A detailed description of the mobile application, backend server APIs and website front-end is presented in Chapters 5-7. The algorithm used to assess the overall diabetes risk is discussed in Chapter 8.

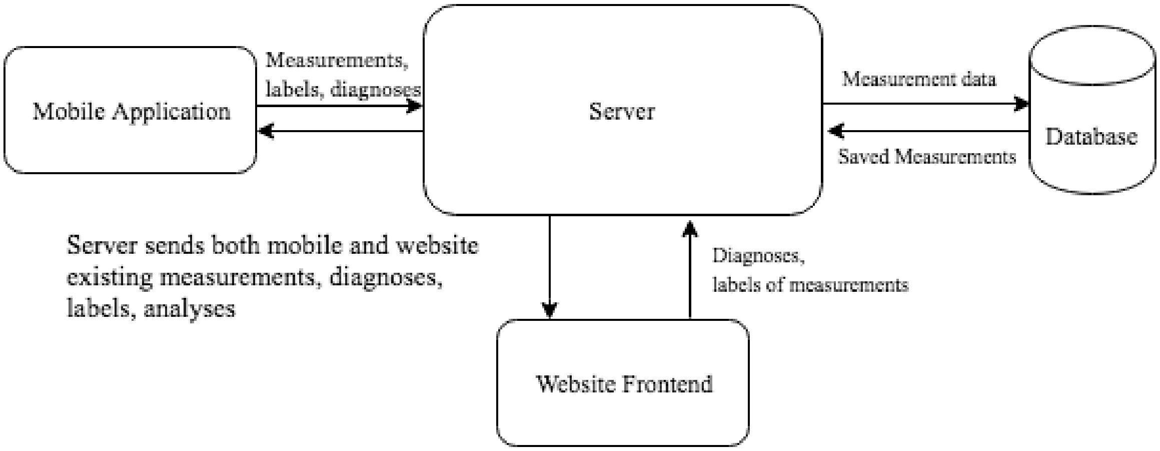


Figure 4-4: Diabetes Screening Platform Architecture

Chapter 5

Server Platform

This chapter describes the implementation of the diabetes server platform which stores all the recorded non-invasive measurements. It also provides functionality to: (1) add clinician diagnoses and (2) run server analyses to get back an analysis on specific measurements and an overall diabetes stage classification. The diabetes server provides cloud storage which makes it possible to record and upload measurements and diagnoses through the mobile application, and then view the data on any Android mobile application or tablet, or through the website front-end.

5.1 Architecture Overview

5.1.1 Background

The server architecture used in this project was developed by another student in the lab, John Mofor, as a separate master's thesis. He developed PyMedServer, an easy-to-use server framework that allows for the large scale collection of medical data. The framework provides User Authentication and Authorization, Permission Classes, support for Asynchronous Tasks, and other useful functionalities [51].

In this project, the PyMedServer framework was customized to meet the needs of the diabetes project by creating the specific APIs required for adding and viewing measurements, clinician diagnoses, and server analyses. We first describe the general database and API structure supported by Django and PyMedServer, and then explain specifically how we used it in this project to create the backend APIs called by the Android applications and website front-end.

5.1.2 Server Database

PyMedServer uses a PostgreSQL database and installs and configures the database automatically. It uses the default Django database features, which represent database objects as Python classes called models. A model contains class attributes called fields, which are the list of database fields that it defines [52]. In the diabetes server, we use fields of type `Integer`, `Float`, `String` and `FileField` to store the information from the measurements. Database fields consisting of other nested model classes are also used in order to organize the model structure. Whenever an object is saved to the database, a unique ID is created for that object.

5.1.3 User Management

Several different types of users may use the diabetes server such as health workers, clinicians, and patients. PyMedServer supports these different use cases though a user management system that starts with the creation of UserGroups. UserGroups contain all the users in the same study. A user can belong to multiple UserGroups. A user can be either a Patient, a Clinician, or a Clinician Admin. The difference between a Clinician and a Clinican Admin is that an Admin has an additional set of privileges. Primarily, they can add other clinicians to a UserGroup. PyMedServer provides permission classes to manage the different access levels of the different user types.

5.1.4 General API Design

When PyMedServer receives an HTTP or HTTPs request, it matches the URL to a class-based view that takes the request and returns a response [53]. Django provides several base view classes, but for our JSON APIs we use an APIView class from a Django library called Django Rest Framework [54].

In the ApiView class, an incoming request is dispatched to an appropriate handler method such as `.get()` or `.post()` [54]. The request passed to the handler is the REST framework Request instance, which has a useful data parsing attribute: `request.data`. The parsed data is then sent to a serializer. Serializers are used to convert between the data sent in a HTTP request and Django objects. We use a request serializer to convert the HTTP request to a Django object and a response serializer to convert a Django object to a HTTP response.

5.2 Creating New Measurements

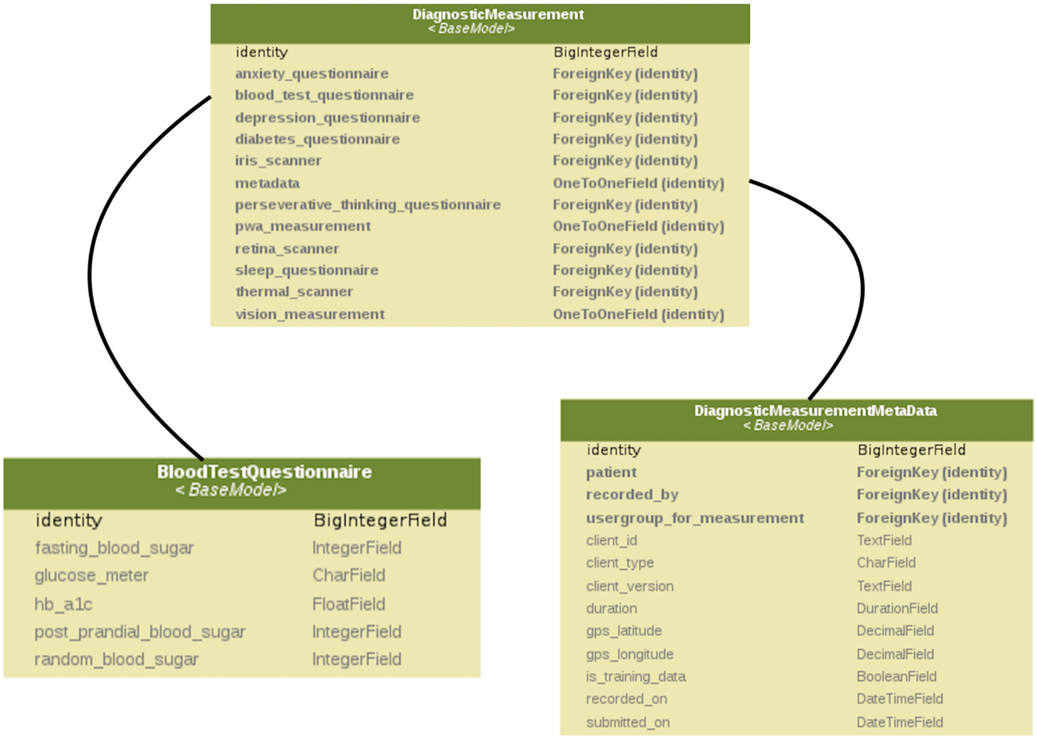
The measurements are added from Android mobile applications, which are discussed further in Chapter 7. The user records the measurement on the mobile application, uploads it to the server using a HTTP request and then stores the response back to the application. In the following section, we describe the implementation of the Add Measurements API used to upload a measurement to the server.

5.2.1 Add Measurement Database Models

When a measurement is uploaded to the server, it is stored in the database as a model containing the information from the measurement as database fields. For questionnaires, the database fields contain the answers to the questions. For the thermal measurement or iris measurement, the model contains a database `FileField` corresponding to an iris or face thermal image file.

In addition to the measurement object, a metadata model called `DiagnosticMeasurementMetaData` is also created and saved to the database. It stores information regarding the measurement itself, such as which patient the measurement was performed on, when the measurement was recorded, the GPS coordinates where the measurement was taken, etc.

The database objects corresponding to the measurement information and the metadata are then stored under one object called `DiagnosticMeasurement`. This object contains fields corresponding to all the different measurements that can be taken. However, only the metadata and the field corresponding to one measurement will ever be populated, and the other fields will remain null. The relationship between these model objects is shown in Figure 5-1, which shows an example blood test `DiagnosticMeasurement` containing a `DiagnosticMeasurementMetaData` and a `BloodTestQuestionnaire`. The rest of the fields in `DiagnosticMeasurement` are null since this is a blood test measurement.



DiagnosticMeasurement < BaseModel>	
identity	BigIntegerField
anxiety_questionnaire	ForeignKey (identity)
blood_test_questionnaire	ForeignKey (identity)
depression_questionnaire	ForeignKey (identity)
diabetes_questionnaire	ForeignKey (identity)
iris_scanner	ForeignKey (identity)
metadata	OneToOneField (identity)
perseverative_thinking_questionnaire	ForeignKey (identity)
pwa_measurement	OneToOneField (identity)
retina_scanner	ForeignKey (identity)
sleep_questionnaire	ForeignKey (identity)
thermal_scanner	ForeignKey (identity)
vision_measurement	OneToOneField (identity)

BloodTestQuestionnaire < BaseModel>	
identity	BigIntegerField
fasting_blood_sugar	IntegerField
glucose_meter	CharField
hb_a1c	FloatField
post_prandial_blood_sugar	IntegerField
random_blood_sugar	IntegerField

DiagnosticMeasurementMetaData < BaseModel>	
identity	BigIntegerField
patient	ForeignKey (identity)
recorded_by	ForeignKey (identity)
usergroup_for_measurement	ForeignKey (identity)
client_id	TextField
client_type	CharField
client_version	TextField
duration	DurationField
gps_latitude	DecimalField
gps_longitude	DecimalField
is_training_data	BooleanField
recorded_on	DateTimeField
submitted_on	DateTimeField

Figure 5-1: DiagnosticMeasurement with BloodTestQuestionnaire and DiagnosticMeasurementMetaData

5.2.2 Add Measurement API Call and Implementation

The Add Measurement APIs are exclusively called by the mobile applications, which send a HTTP request with JSON data formatted to look like the `DiagnosticMeasurement` object that will be stored in the PostgreSQL database. In particular, the JSON request contains the properties `metadata` and an additional property corresponding to a measurement that will become a field in the `DiagnosticMeasurement` model. For example, if the measurement being sent was a questionnaire, this property would be called `diabetes_questionnaire`.

Each measurement type has its own URL that must be called to add that type of measurement. For example, the URL https://diabetesServer.com/apis/diagnostics/add_measurement/questionnaire/clinician/ would be used to add a questionnaire measurement. Then the reverse lookup call finds the APIView associated with that URL. The request is dispatched to the post handler since all the add measurement requests are POST requests.

In the POST method, the request serializer converts the parsed `request.data` attribute to a Django object. To create the request serializer, we use a ModelSerializer, which automatically generates a set of fields based on a model [55]. In this case, the model is `DiagnosticMeasurement`. Then the model is validated to ensure that the request has all the necessary fields in order to be saved as a `DiagnosticMeasurement` object to the database. The serializer enforces that any required fields are present. If they are missing, the model will fail to validate.

If the request data validates, the database layer is accessed to store a `DiagnosticMeasurementMetaData` model object and a measurement model object to the database. Then a `DiagnosticMeasurement` with fields corresponding to these objects is also stored. A response serializer converts the final `DiagnosticMeasurement` object to an HTTP response, which is returned from the APIView.

5.3 Creating Diagnoses

The server platform allows for a clinician diagnosis of a patient to be annotated from either the Android application or the website front-end. To add a diagnosis, a clinician selects the measurements that contribute to his or her diagnosis, and assigns one of several different disease label options or provides a free-form text input called the open-ended clinical label. The clinician diagnosis has special considerations for retina grading.

5.3.1 Database Models for Clinician Diagnosis

The clinician diagnosis information is stored in a model class called `ClinicianDiagnosticPredictions`. The model class stores which clinician recorded the measurements, the disease diagnosis labels, and the open-ended clinical label. The disease label is a choice field, which means that the options for this field are choices predefined alongside the model—the user cannot enter free-form text. Disease labels exist both for retina grading, with options such as glaucoma and arterial occlusions, and for other diseases such as hypertension and anemia. Predefined disease labels are used to allow models to be trained using supervised machine learning. Though clinicians are encouraged to use these predefined labels, there is an additional label for entering in free-form text to give clinicians flexibility to enter uncommon diagnoses—this option is called the open-ended clinical label.

`ClinicianDiagnosticPredictions` inherits from a model class called `AbstractDiagnosticPredictions` which contains more general information about the prediction, such as when it was created and what measurements were used in the diagnosis. Another class, `ServerDiagnosticPredictions`, also inherits from the class because of the similarities between a diagnosis created by a clinician and a server. This will be further explored in Section 5.4.1.

5.3.2 API Call and Implementation

The JSON API request contains the measurement IDs of the selected measurements, as well as either a disease label or an open-ended clinical label, or both. In this case, all API requests go to the same URL and APIView. The request serializer converts `request.data` into a Django object called ApiRequest, which has three fields corresponding to the selected measurements, disease label, and open-ended clinical label.

In addition to validating the ApiRequest, a series of checks are performed to make sure that the measurements list is not empty and that either one of the disease label or the open-ended clinical label is provided. Once these checks pass, a new `ClinicianDiagnosticPrediction` object is created with the fields from the request, and saved to the database. Then a response serializer returns the `ClinicianDiagnosticPrediction` object as an HTTP Response.

5.4 Server Analysis Implementation

In order to integrate machine learning prediction algorithms into our screening platform, it is necessary to provide support for such algorithms on the server. This feature enables algorithms to be run in real-time by health workers in the field who are using the mobile application. In this section, we describe the Run Server Analysis API, which is used to generate measurement analyses on the server.

A user can launch a server analysis of selected measurements through the Android mobile application. This initiates two types of analyses. The first is an initial analysis that is only run on individual measurements. For the questionnaires, this initial analysis is computed on the Android mobile application before the measurement is uploaded to the server. But in the case of the iris measurement, the intermediate analysis happens on the server. We made the decision to have the iris analysis happen on the server instead of the phone because this algorithm is a time- and resource-expensive computation that can be performed faster with the computing abilities of the cloud server.

Then the other type of analysis is a secondary algorithm that combines these initial measurement analyses to return an overall diabetes stage classification. In Chapter 8, we explore using a Bayesian algorithm to return the diabetes stage, and in this section we detail how the Run Analysis API initiates the initial and secondary analyses.

5.4.1 Database Models for Server Analysis

The server analysis is stored in the model class `ServerDiagnosticPredictions` which as mentioned in Section 5.3.1 inherits from the `AbstractDiagnosticPredictions`. In addition to the inherited fields, such as measurements and created time, the `ServerDiagnosticPredictions` also has fields to store errors encountered by the server and the input parameters provided to the server algorithm. Two other important fields are diabetes risk, which stores the final diabetes stage classification, and iris scanner results. This field is a `ManyToManyField`, which creates an invisible "through" model that relates the source model `ServerDiagnosticPredictions` to the target model, an `IrisScannerResult` object.

An `IrisScannerResult` stores the information produced from calling the code that a previous student in the lab, Tania Yu, developed to process texture features in the iris. The code she developed outputs `Float` values for 24 regions in the eye, and the `IrisScannerResult` has 24 fields corresponding to the values of those regions. It also has a field `measurement` which contains the `DiagnosticMeasurement` object from which the result was created. These models are shown in Figure 5-2. The one-sided arrow indicates that `ServerDiagnosticPredictions` inherits from `AbstractDiagnosticPredictions` and the double sided arrow shows the many-to-many relationship between `ServerDiagnosticPredictions` and `IrisScannerResult`.

5.4.2 API Call and Implementation

Currently, the Run Analysis API is only called by the Android mobile phone. We made the design decision to have only one Run Analysis API that handles server

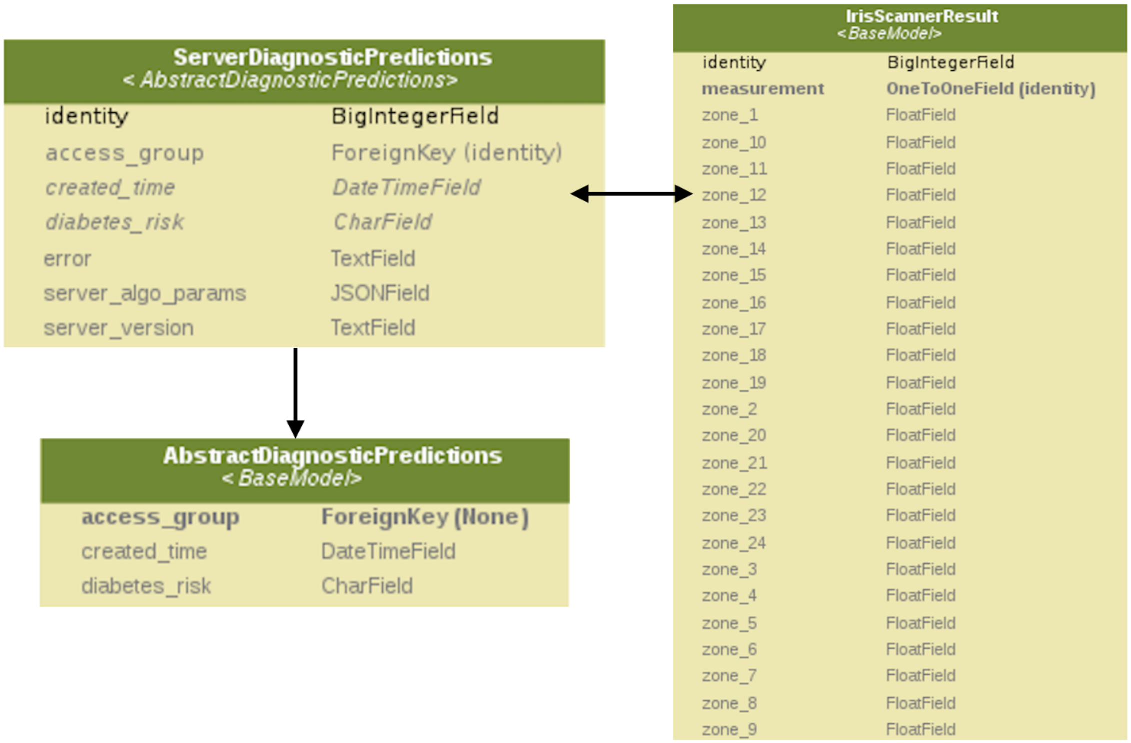


Figure 5-2: ServerDiagnosticPrediction, AbstractDiagnosticPredictions and IrisScannerResult

analysis requests for all types of measurements as opposed to having separate APIs for different measurements. This simpler design has the benefit of more easily combining measurement analyses to calculate a final diabetes stage classification.

The JSON API request contains the properties `measurement_ids`, which is a list of the different measurements that the user wants the server to analyze. Once the request serializer has created and validated a ApiRequest Django object, a new `ServerDiagnosticPrediction` object is created with measurements corresponding to the IDs in `measurement_ids`. Then the database layer is accessed to filter iris measurements from all other measurements provided by the request. If the iris measurement already has an associated `IrisScannerResult` (from a previous run analysis), it is added to the `iris_scanner_results` field of the `ServerDiagnosticPrediction`. If all of the iris measurements have been calculated, an asynchronous task is kicked off to calculate the overall diabetes risk. Otherwise each unanalyzed iris measurement launches an asynchronous task to calculate an iris scanner result. The input to the task is the unanalyzed iris measurement and the ServerDiagnosticPrediction that was created in the POST handler. But before it is passed to any asynchronous task, the ServerDiagnosticPrediction is saved to the database. Then the task saves the iris analysis results as an IrisScannerResult object and also adds it to the iris_scanner_results field of the ServerDiagnosticPrediction. Lastly, a response serializer returns the ServerDiagnosticPrediction as an HTTP response. The asynchronous tasks are discussed more in the next section.

5.4.3 Asynchronous Tasks

The asynchronous tasks are managed by a Python library called Celery. The PyMedServer provides support for this library and running asynchronous background tasks. Asynchronous tasks are used to run time-intensive processes like iris analysis and bayesian network algorithms in order to speed up computation time since each process can be run independently. However, we made the design decision to wait to call the overall diabetes classification algorithm until all of the iris results of the iris measurements being analyzed had been calculated. This ensures that the results of all the iris analyses can be incorporated into the overall risk assessment.

The iris analysis algorithm is launched in the POST handler of the Run Analysis API, which passes along the iris measurement and the server prediction object to an asynchronous task. Then in the asynchronous task, a Python subprocess is created to run the iris analysis algorithm. A subprocess runs the algorithm in a new, contained process, but still permits access to the input, output and error pipes. The input to the iris analysis algorithm is the path where the iris image is stored, which is obtained from the iris measurement. Then the output is loaded from the subprocess as JSON, stored to a newly created IrisScannerResult object and added to the iris_scanner_results field of ServerDiagnosticPrediction. Finally the IrisScannerResult is saved and ServerDiagnosticPrediction is updated in the database.

At the end of the asynchronous task, a check is performed to see if the number of

IrisScannerResults in ServerDiagnosticPrediction matches the number of iris measurements in the measurements field of ServerDiagnosticPrediction. If yes, an asynchronous task is initiated to calculate the overall diabetes classification. The check must be present at the end of all asynchronous tasks for iris analyses because the tasks happen simultaneously so it is unknown which one will finish last.

The task to calculate diabetes risk happens similarly to the one that calculates the iris result. The one difference is that only the ServerDiagnosticPrediction is passed along to the task. Again, the asynchronous task here creates a subprocess that runs the algorithm. The output is a JSON object containing the diabetes classification, which is then added to the ServerDiagnosticPrediction and updated in the database.

5.5 Viewing Measurements

To view the measurements, clinician diagnoses, and server analyses on the mobile application or website front-end, there are three separate APIs: View Measurements, View Clinician Analyses and View Analysis. The structure of these APIs follows much the same format as the previous APIs. The JSON request contains properties such as patient or measurement IDs that allow the user to specify which objects are returned. Then the database is queried to find and return objects matching those specifications, and the response returns all matching objects to the caller to be displayed. Table 5.1 shows the different fields in the API that can be used to filter the HTTP response.

The view analysis response has an additional field report_html because of the way the server analysis is displayed on the Android mobile application. The Android application simply opens a WebView that can properly format HTML, so the report_html contains the results of the server analysis as an HTML String. The String is generated in the response serializer by iterating through all the measurements in ServerDiagnosticPrediction and appending the results of the analyses calculated on the mobile application (for the questionnaires) and the analyses generated on the server, such as the iris analysis and the diabetes stage classification, to

API	Request Properties	Response
View Measurements	Patient IDs Measurement IDs UserGroup IDs	`DiagnosticMeasurement`
View Diagnoses	Patient ID UserGroup IDs	`ClinicianDiagnosticPrediction`
View Server Analysis	Patient IDs Measurement IDs Server Diagnosis IDs	`ServerDiagnosticPredictions`

Table 5.1: Query Parameters for View APIs

the `report_html String`. HTML elements can be added to the `report_html String` so that the analysis results appear in a table or in bold font in the WebView. By pursuing this implementation, we chose to move much of the functionality of displaying analysis results to the server rather than the Android application in order to facilitate future maintenance of the APIs.

5.6 Editing Measurement Labels

For the purpose of developing labels for supervised data, it is necessary to provide a means for labeling data. A unique feature of our mobile platform is that it also provides the ability to label specific measurements using the mobile app itself. In particular, the platform supports individual labeling of retina, iris, thermal, and diabetes questionnaire measurements for clinicians. All of the labeling options are choice fields with predefined options. Clinicians are encouraged to use one of the predefined options for supervised machine learning purposes, but to allow flexibility in grading, there is also a free-form text input for a retina clinical label called open-ended clinical label.

5.6.1 Database Models

Each of these four measurements: retina, thermal, iris and questionnaire, has a field `clinical_label`, while the retina measurement also has a field `open_clinical_label`. When the Edit Label API is called, these clinical label fields are modified.

5.6.2 API Call and Implementation

The JSON HTTP request contains the measurement ID of the measurement being edited, and another property saying which label is being edited. This property corresponds to the measurement type, i.e. `diabetes_questionnaire` if the label being edited is for a diabetes questionnaire measurement. Then the request serializer first checks that only one measurement type property is provided. The one exception is retina, in which case both a clinical label and an open-ended clinical label may be provided. The request serializer also checks that the clinical label provided matches the measurement type, i.e. a diabetes questionnaire clinical label is provided for a diabetes questionnaire measurement type.

Finally, inside the POST method, a check is performed to ensure that the measurement is of the same type as the measurement type provided in the request, and if so, the measurement's clinical label is updated and the measurement itself is also updated in the database. The response serializer returns the measurement object as an HTTP response.

Chapter 6

Server Platform Front-end

The website front-end is built for the needs of clinicians interacting with the platform. Our data collection partners wanted to have a website where they could monitor existing patients, assign labels to individual measurements, add clinician diagnoses for multiple measurements, bulk download data to a CSV, approve clinician registration requests, and more. The website front-end is particularly important to encourage adoption and optimal use of our platform. For example, our clinician partners were more comfortable grading on desktop computers rather than on mobile phones—they wanted a large screen to properly view measurements and assign accurate measurement labels. With considerations like these, we created the website front-end.

6.1 Overview

When a browser requests the content at a particular URL from our diabetes server, the server resolves the URL pattern and returns a View, which in turn returns an HTML file that serves the content on the webpage. We use a shortcut to this process by using a render function that combines a given HTML template with a given content dictionary and returns a response [56].

For some of our templates, the patient ID is needed in order to customize the HTML page. So in the URL definition, a regex is used to create a pattern that must be matched: `([0-9]{14})`. This regex defines a format for an 14-digit-long patient

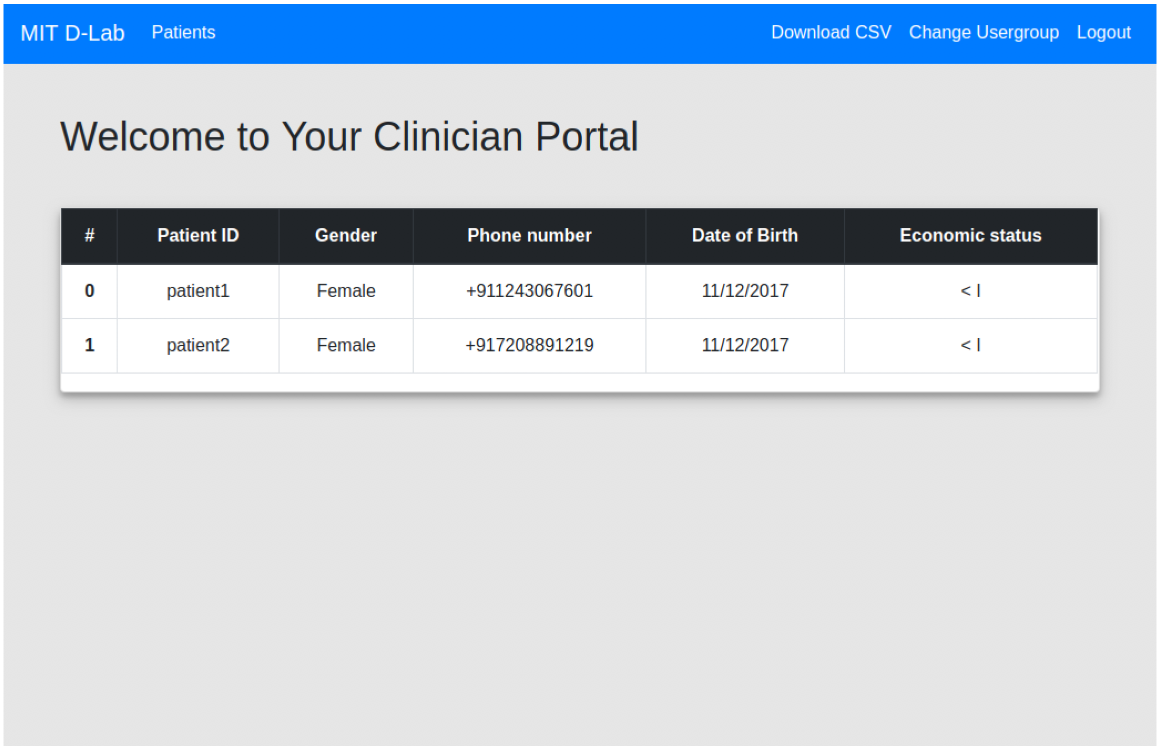
ID that should be part of the URL. Then in the render function, this ID is passed through the context to the template. The HTML templates use Javascript and CSS files that are served from a static folder configured by PyMedServer.

The first thing that a patient sees is a login screen. Once they enter the correct login credentials, the server issues an authorization token, which is used to make API calls. Once the login is successful, the user will see a modal asking them to pick a UserGroup. The chosen UserGroup is stored in local storage and the user will only see patients, measurements, diagnoses, and analyses in that group, but they have the option to switch UserGroups from the Clinician Portal page.

6.2 Clinician Portal

The Clinician Portal shows a table with all of the clinician's patients that are currently in a UserGroup together. The table is defined in the HTML file, but it is populated in the client-side Javascript file. Axios, a Javascript library, is used to make an HTTP request to a View Patients API (provided by PyMedServer) on the server with one parameter: the ID of the current UserGroup. The API returns the UserGroup patients accessible to the caller. Then the script iterates through the patients to populate the table with the patient name, gender, phone number, birthdate, and economic status. The table rows are also encoded with the ID of the patient, and when clicked, the row shows a modal that can be used to go to either the Clinician Diagnosis or the View Measurements page. Depending on what is clicked, the window location is replaced with the URL to view measurements or view diagnoses, and the patient ID from the row is passed along to the new URL.

From this page, there are also some additional features. The user can switch UserGroup or they can bulk download the measurements as CSV. The latter feature is just a redirect to the download measurement URL; the implementation of this page is discussed in Section 6.5. The Clinician Portal is shown in Figure 6-1.



Welcome to Your Clinician Portal

#	Patient ID	Gender	Phone number	Date of Birth	Economic status
0	patient1	Female	+911243067601	11/12/2017	< l
1	patient2	Female	+917208891219	11/12/2017	< l

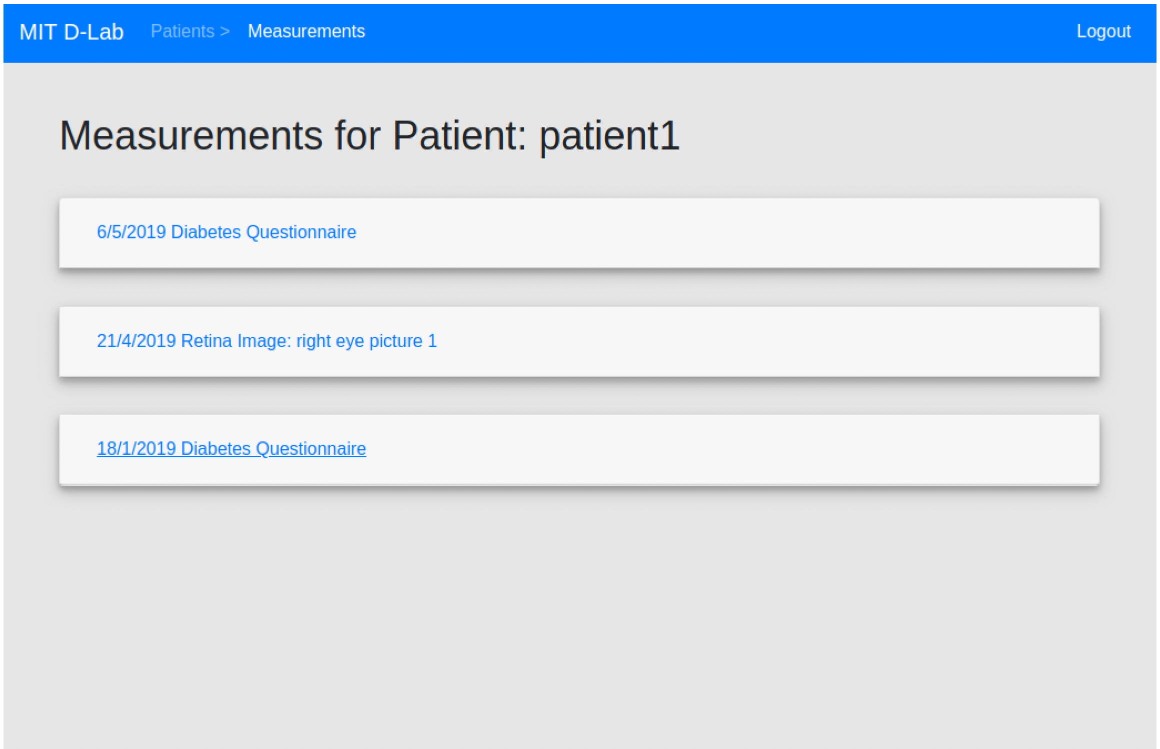
Figure 6-1: Clinician Portal

6.3 Patient Measurements

If the user clicks on a patient, and then selects view measurements, they are taken to the Patient Measurements page, which shows all of the measurements for a particular patient. The measurements are shown using the Bootstrap Accordion component with cards that can be expanded or collapsed. The Patient Measurements page is shown in 6-2.

Again Axios is used to make an HTTP request, but this time to the View Measurements API with one parameter: the ID of the current patient. The response from the server returns all the measurements associated with that patient. Then each measurement is displayed in its own card in the accordion. In addition to the measurement, the card also displays the current clinical label of the measurement and the clinician who recorded that measurement. The clinician ID is retrieved from the measurement metadata.

The cards for iris, retina, thermal, and diabetes questionnaire also have an addi-



Measurements for Patient: patient1

6/5/2019 Diabetes Questionnaire

21/4/2019 Retina Image: right eye picture 1

18/1/2019 Diabetes Questionnaire

Figure 6-2: Patient Measurements

tional Add Label button to allow the clinician to add a clinical measurement label for that measurement. We went through several design iterations of how to add this functionality. When the button is clicked, jQuery is used to show a modal from which the clinician can select the appropriate label option. In the first design, we created unique modals for each measurement. We did this because the clinical label modal has to have the measurement ID of whichever measurement is being labeled, and the most straightforward solution was to create a unique modal for each measurement. However, the time needed to create all the necessary modals slowed down the loading time of the page.

Therefore, in the modified design, only four different modals are created, each corresponding to the different measurement types. When the Add Label button is clicked, jQuery is used to add the measurement ID from that measurement to the modal being called, thus eliminating the need for unique modals for each measurement.

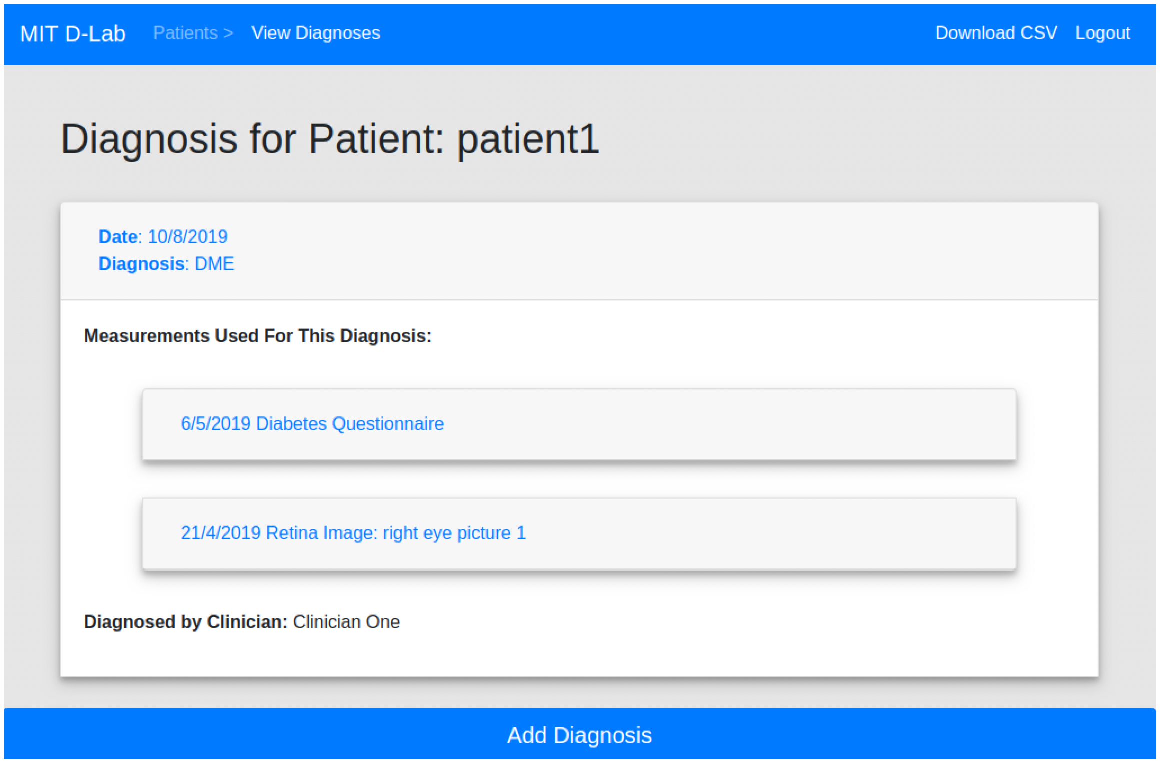
Here we could have made the decision to simplify even further and only have one modal for all measurements, but significant differences in the different measurement types would have complicated the design. For example, each measurement type has a different HTTP request parameter and different clinical label options. And while thermal, diabetes questionnaire, and iris measurements use a Bootstrap radio button to limit the user's selection to one clinical label, the retina measurement uses checkboxes and also has an open text form to enter in the open clinical label. Therefore, in order to accommodate the differences between the measurement types, and to allow for easier future modifications, we decided to keep separate modals for the different measurement types. With the new design, the number of created modals stays constant even as the number of measurements increases, allowing for scalability.

Finally, in the modal itself, once the submit button is clicked, indicating that the user is done selecting labels, Axios is used to call the Edit Label API using the measurement ID sent from the measurement card and the selected labels from the modal.

6.4 Patient Diagnosis

From the Clinical Portal page, if the user decides to click the option View/Add Diagnoses, the user is taken to the Patient Diagnosis page, which displays all the previous diagnoses entered for that patient. The View Diagnoses API is called to get all the previous diagnoses, and the diagnoses are shown in a Bootstrap Accordion format similar to the Patient Measurements page. In each card, the actual diagnosis is shown in the header of the card, and the associated measurements are shown as nested cards in the body of the outer diagnosis card. Also shown is the clinician who recorded the diagnosis, again by using the clinician ID in the metadata. The View Diagnoses page is shown in Figure 6-3.

The Patient Diagnosis differs from the Patient Measurements page in that a measurement may be shown twice on this page, which can cause unwanted behavior. If two measurement cards have the same ID in different diagnosis cards, toggling one's



Diagnosis for Patient: patient1

Date: 10/8/2019
Diagnosis: DME

Measurements Used For This Diagnosis:

6/5/2019 Diabetes Questionnaire

21/4/2019 Retina Image: right eye picture 1

Diagnosed by Clinician: Clinician One

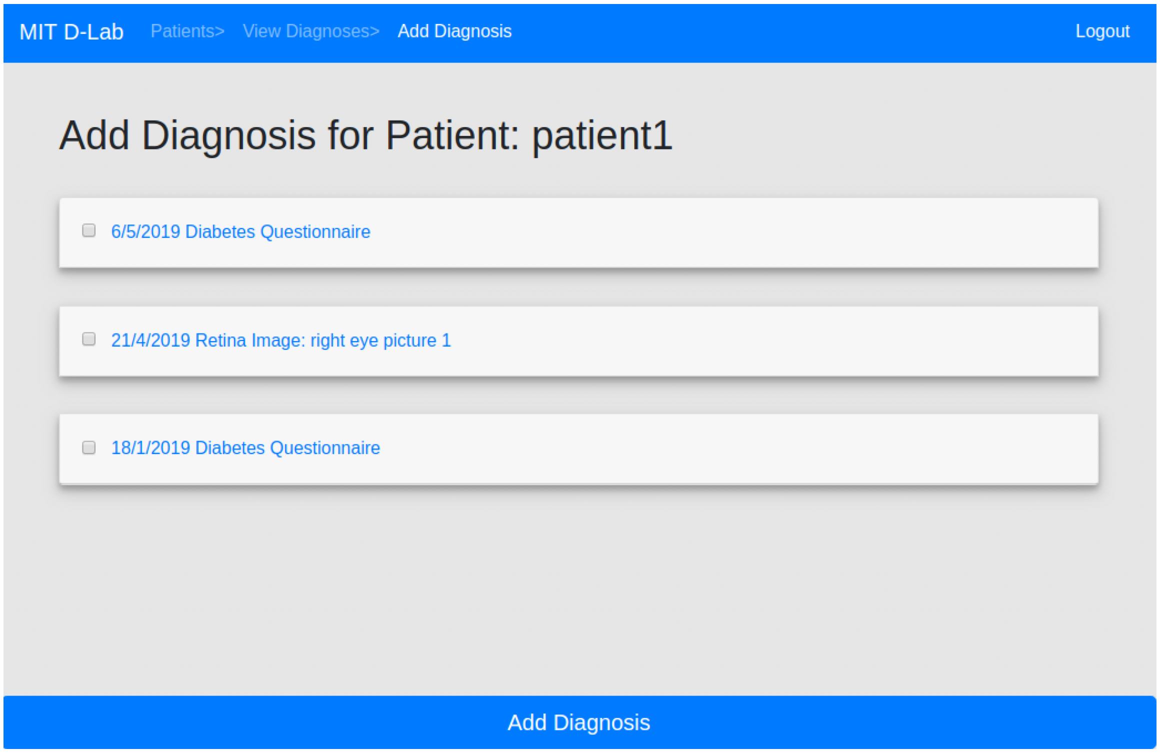
Figure 6-3: View Diagnosis Page

collapse behavior will affect both cards. In order to remedy this problem, the cards were issued unique IDs by combining the diagnosis ID with the measurement ID.

6.4.1 Adding Diagnosis

To add a diagnosis, a clinician clicks the Add Diagnosis button at the bottom of the View Diagnoses page, which redirects to another URL (while passing along the current patient's ID as a parameter). The Add Diagnosis page is shown in Figure 6-4.

The Add Diagnosis page is very similar to the View Measurements page, except that each card is selectable through a checkbox. Once the user selects checkboxes, they can click the Add Diagnosis button on the page, which toggles a modal. The modal has three options: patient diagnosis, eye diagnosis, or healthy. We chose this first modal in order to better organize the diagnosis options and create a better user experience for clinicians using our website. If eye diagnosis or patient diagnosis is



Add Diagnosis for Patient: patient1

□ 6/5/2019 Diabetes Questionnaire

□ 21/4/2019 Retina Image: right eye picture 1

□ 18/1/2019 Diabetes Questionnaire

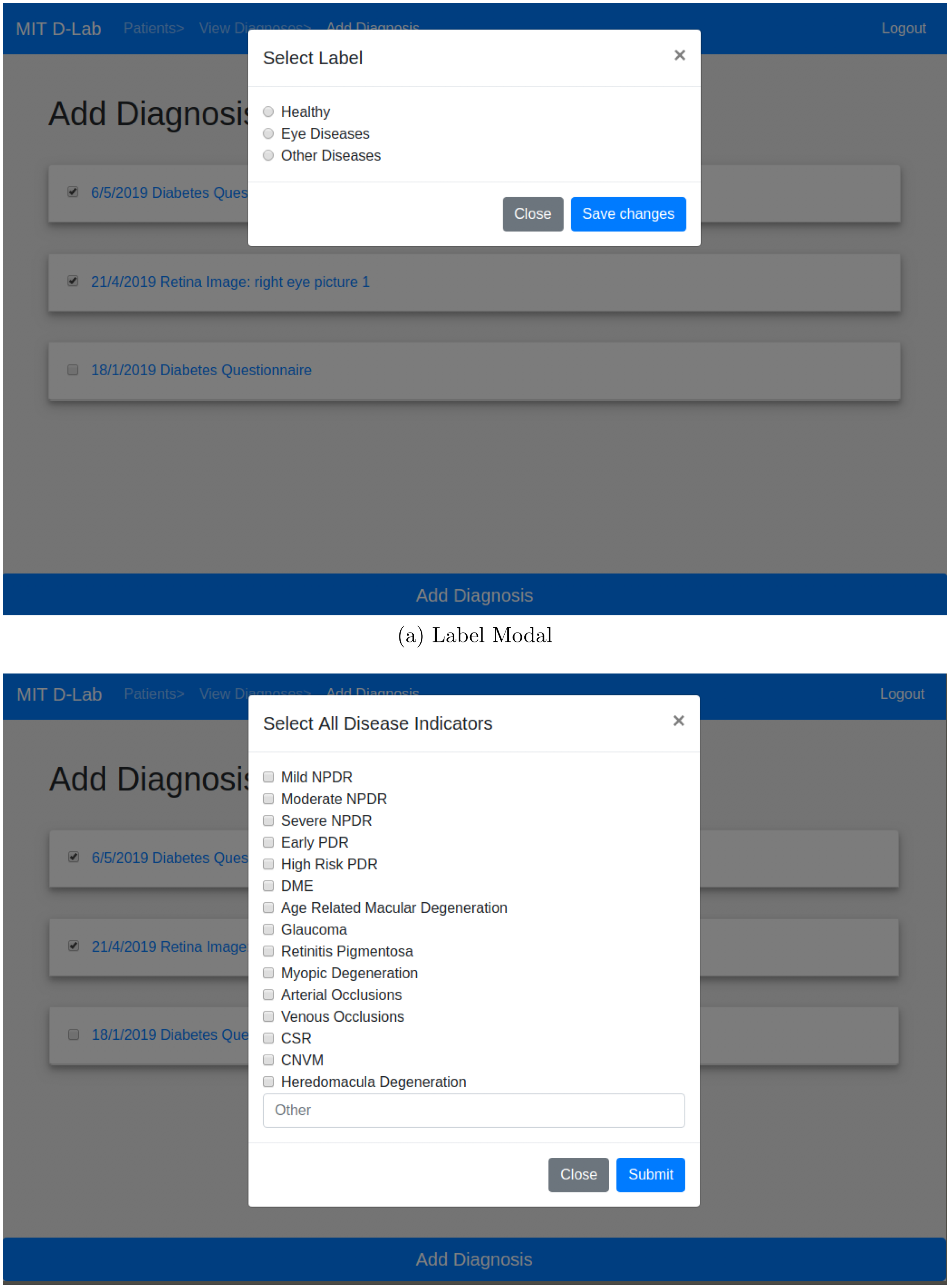
Figure 6-4: Add Diagnosis Page

selected, a second modal shows up with those diagnosis options. Bootstrap checkboxes are used so multiple diagnoses can be selected. The modals are shown in Figure 6-5.

Finally the options that were selected in the modals are sent to the Add Diagnosis API. Successfully adding a diagnosis redirects the user back to the View Diagnoses page.

6.5 Bulk Download

When the Bulk Download API is called, it returns a zip file containing all the measurement images and questionnaire data. The images are stored under folders with the patient usernames, and the questionnaire data is stored as CSV file in the zip file. The API is called first from a URL, then an APIView returns an HTML template response. The template response is a webpage where the clinician selects the UserGroup, patients, measurement types and date range of the measurements that



Select Label ×

○ Healthy
○ Eye Diseases
○ Other Diseases

Close Save changes

☑ 6/5/2019 Diabetes Ques

☑ 21/4/2019 Retina Image: right eye picture 1

□ 18/1/2019 Diabetes Questionnaire

(a) Label Modal

Select All Disease Indicators ×

□ Mild NPDR
□ Moderate NPDR
□ Severe NPDR
□ Early PDR
□ High Risk PDR
□ DME
□ Age Related Macular Degeneration
□ Glaucoma
□ Retinitis Pigmentosa
□ Myopic Degeneration
□ Arterial Occlusions
□ Venous Occlusions
□ CSR
□ CNVM
□ Heredomacula Degeneration

Other

Close Submit

☑ 6/5/2019 Diabetes Ques

☑ 21/4/2019 Retina Image

□ 18/1/2019 Diabetes Que

(b) Disease Indicators Modal

Figure 6-5: Add Diagnosis Modals

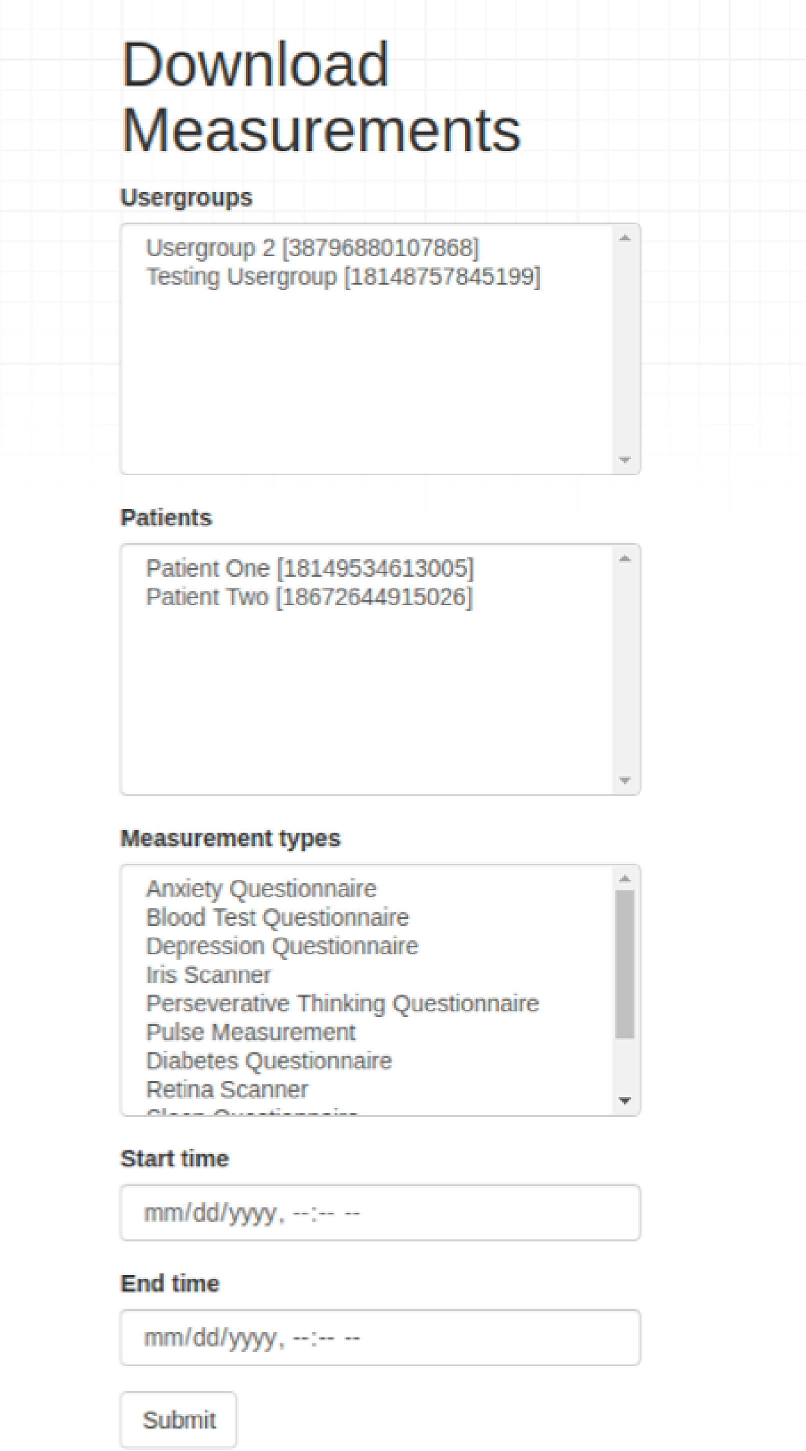
he or she would like to download. The submit button sends an HTTP request to another APIView, which creates the zip folder, and returns it as an HTTP response to the browser. This API has several different server APIs, but we chose to explain it fully in this section because the API is only accessible through the front-end and has several important front-end components.

6.5.1 Api Call and Implementation

When the Bulk Download API, download measurements, is called, the URL is resolved to an APIView `ViewDownloadMeasurementUI`. The GET method in this view renders an HTML template. The HTML template uses a Django REST framework HTMLFormRenderer, which renders the data returned by the request serializer into an HTML form. The request serializer returns the UserGroups, patients, measurement types, start time and end time, which are then rendered in a form as shown in Figure 6-6.

This form takes input from the clinician, who then makes the appropriate selections from these categories to filter the downloaded measurements. Once the clinician hits submit, the form passes on the selections to the POST method within another view in download_measurements, `DownloadMeasurementsApi`.

Within this POST method, a check is performed to ensure that the clinician has access to the patients and UserGroups selected, then the download zip file is created as is the CSV file to store questionnaire data. For each type of measurement, the measurement fields are written to the CSV file, unless the measurement is a file, which cannot be easily written to a CSV file. Instead if the field is a file, the measurement is added as a file in a folder in the download zip file. The zip file organization is shown in Figure 6-7. Then an HTTP response is returned with the zip file download.



Download Measurements

Usergroups

Usergroup 2 [38796880107868]
Testing Usergroup [18148757845199]

Patients

Patient One [18149534613005]
Patient Two [18672644915026]

Measurement types

Anxiety Questionnaire
Blood Test Questionnaire
Depression Questionnaire
Iris Scanner
Perseverative Thinking Questionnaire
Pulse Measurement
Diabetes Questionnaire
Retina Scanner
Sleep Questionnaire

Start time

mm/dd/yyyy, --:-- --

End time

mm/dd/yyyy, --:-- --

Figure 6-6: Download Measurements Form

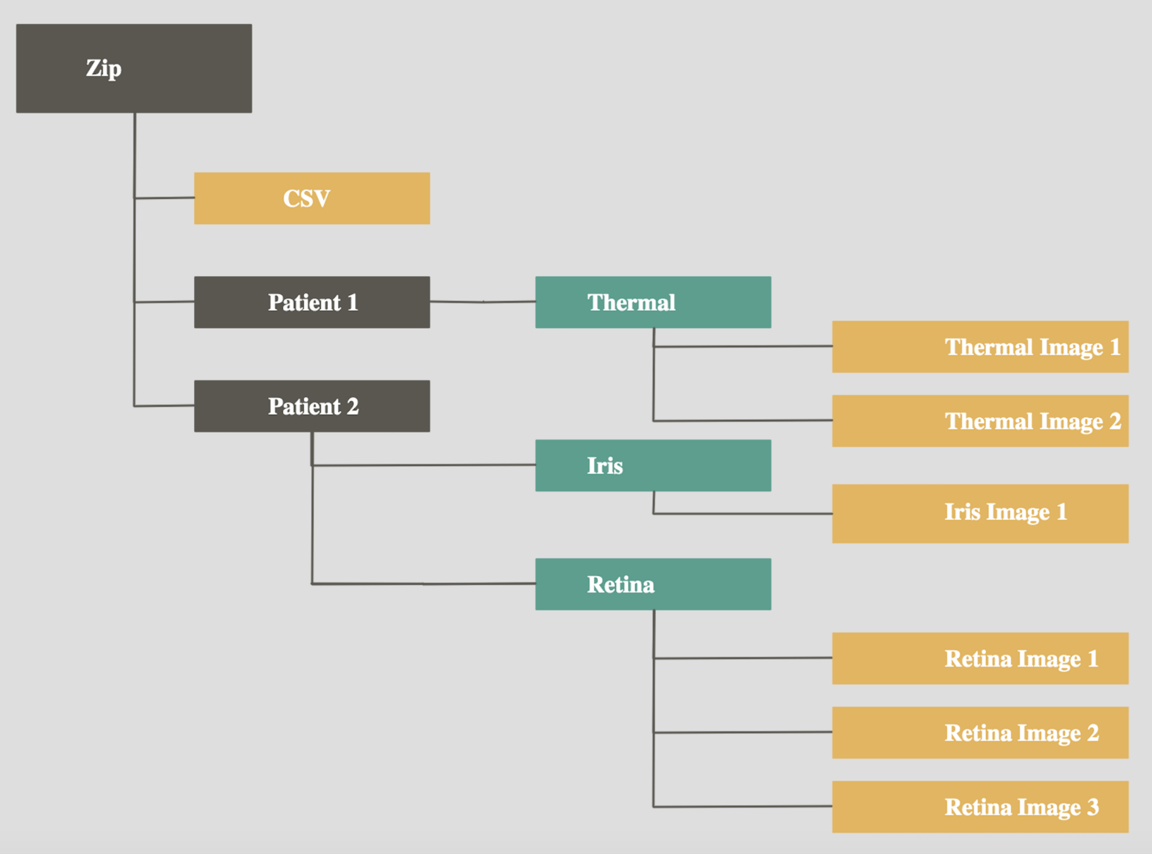


Figure 6-7: Download Measurement Zip File Organization

Chapter 7

Mobile Application

The Android applications are built using PyMedServer-Android, an Android library which was also created as part of John Mofor's MIT master's thesis [51]. The mobile applications were built using this template. Most of the implementation falls outside the scope of this project, and is only briefly summarized here, but there were some customizations made on top of the basic PyMedServer-Android apps for the needs of the diabetes project. In this chapter we discuss one such modification made to the application in order to calculate measurement analyses for questionnaires.

7.1 Android Libraries

PyMedServer-Android is divided into two libraries, one of which, APILib, enables communication with the server. APILib issues HTTPs requests to a PyMedServer, in this case the diabetes server. It serializes a Java Request Object into a format understandable by the server, a JSON object, and parses the response returned from the server back into a Java object [51]. PyMedServer-Android also has other built-in functionalities, such as authentication and offline support, that make communication with the server seamless.

7.2 Mobile Applications for Measurements

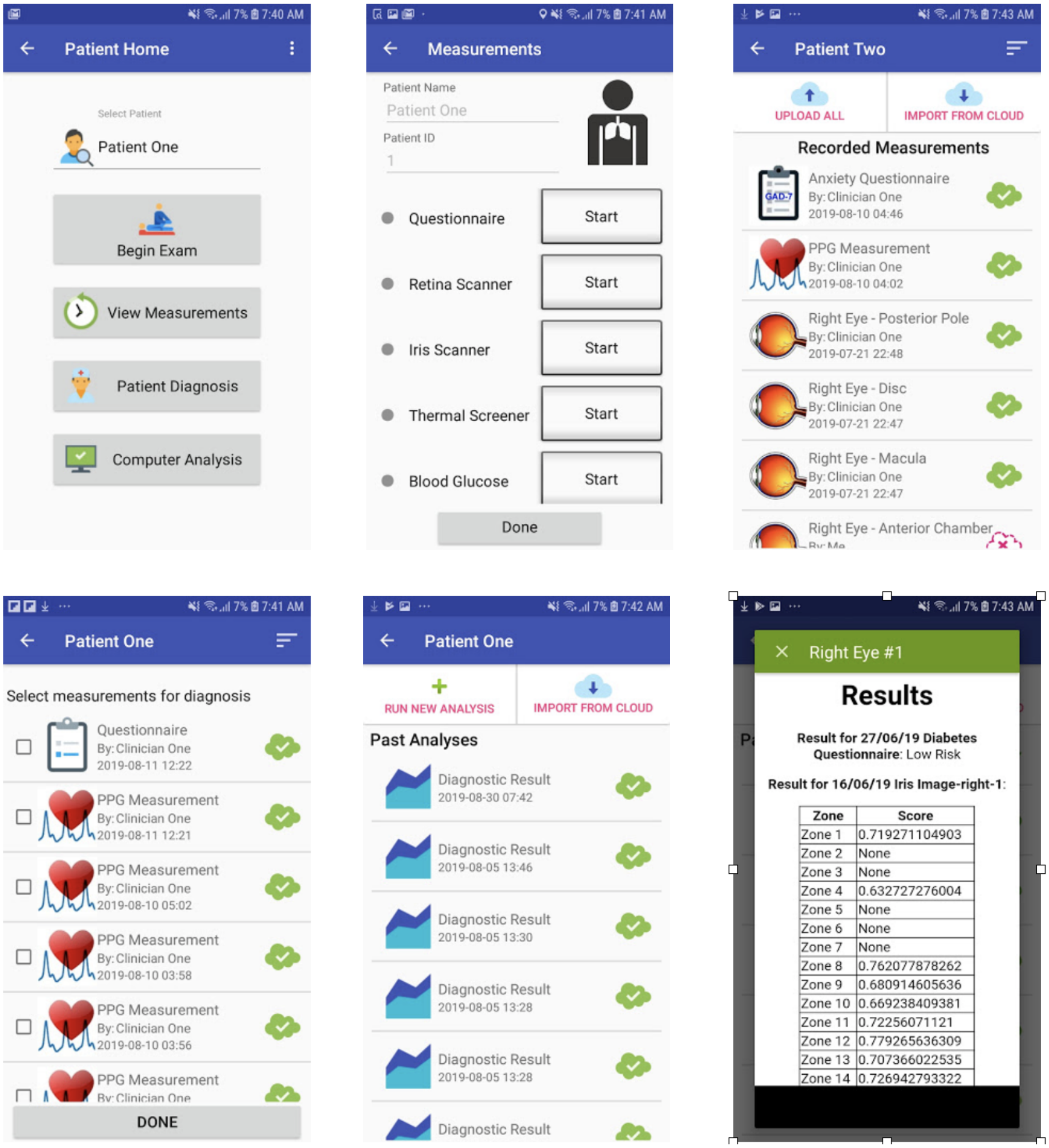
Separate mobile applications were created to record the non-invasive measurements described in Chapter 4. We have separate apps to record the diabetes questionnaire, psychology questionnaires (one for each questionnaire), thermal face images, iris images, retina images, visual acuity test, PPG and blood test measurements. Each of these measurement apps can be used on their own to record and label measurements, but they can also be launched from a container app. The container app is able to aggregate data from the different measurements types to allow clinician diagnoses and server analyses. All of these apps, including the container app, have been published on the Google Play Store. Some sample screen captures of the mobile applications are displayed in Figure 7-1.

7.3 Measurement Analysis on Mobile Application

After a user finishes taking a measurement, a results screen is displayed indicating that the measurement was successfully completed. For some measurements, such as the PSQI, GAD-7, PHQ-9 and diabetes questionnaire, this screen additionally shows a score based on the answers entered. The screens are displayed in Figure 7-2.

As part of this project, we modified the basic PyMedServer-Android application to calculate the measurement scores from the standard scoring for these questionnaires. We chose to calculate the scores in the application instead of the server because of the low computation power required to score these particular measurements and display the results shortly afterwards on the results page.

The PSQI, GAD-7, PHQ-9 questionnaires are all scored according to standard guidelines for these questionnaires, and the diabetes questionnaire is scored according to the IDRS guidelines. Both a risk score, the raw value attained from the calculation, and a more human-readable label associated with the score are determined before the results page is shown to the user and saved. Then these values are uploaded to the server along with the rest of the measurement values. The questionnaire score and the label are both preserved so that the score could be used with a different labeling system in the future. When the view diagnosis API is called, it returns this risk score label as the initial analysis of these measurements.



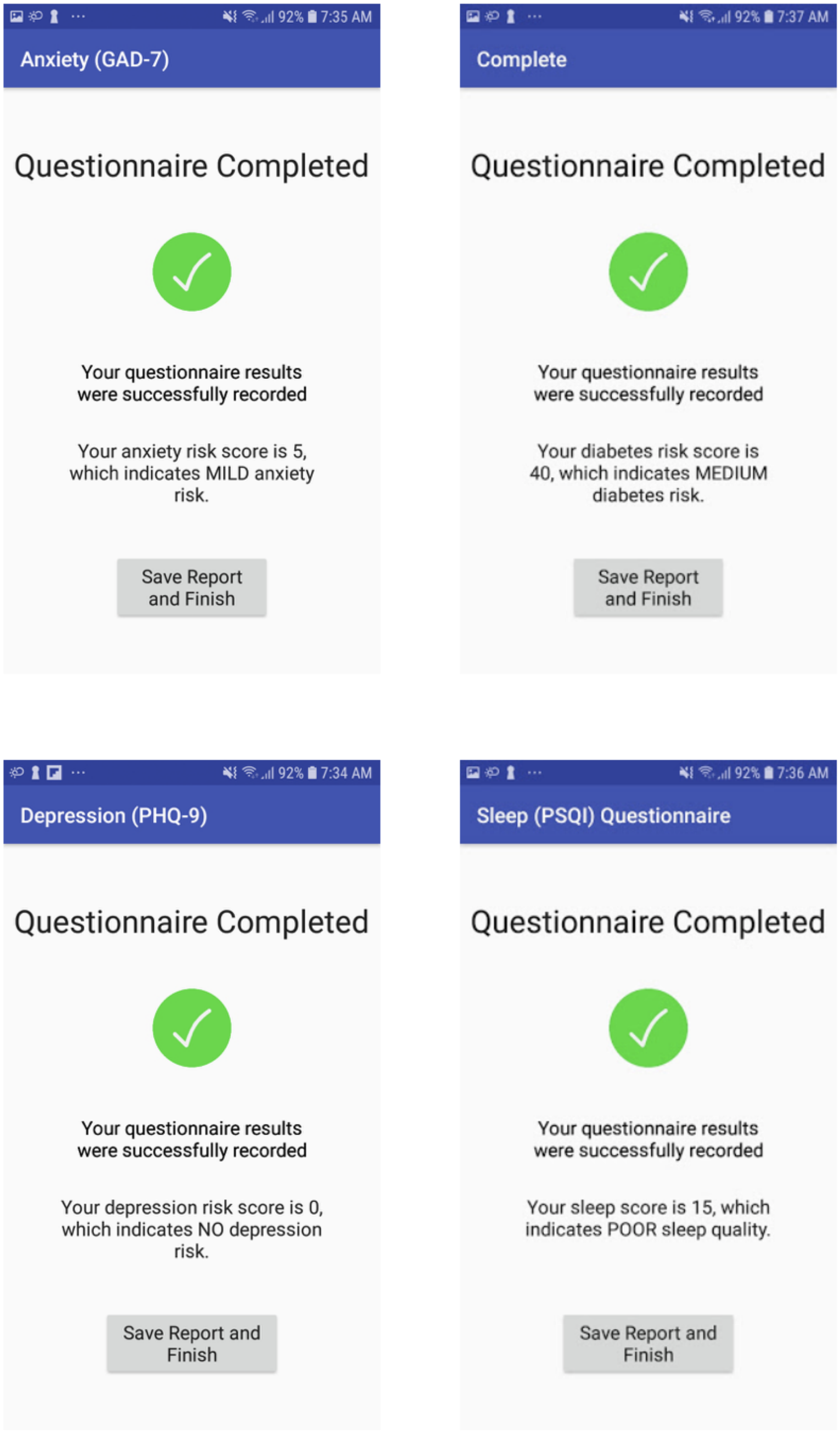
Results

Result for 27/06/19 Diabetes Questionnaire: Low Risk

Result for 16/06/19 Iris Image-right-1:

Zone	Score
Zone 1	0.719271104903
Zone 2	None
Zone 3	None
Zone 4	0.632727276004
Zone 5	None
Zone 6	None
Zone 7	None
Zone 8	0.762077878262
Zone 9	0.680914605636
Zone 10	0.669238409381
Zone 11	0.72256071121
Zone 12	0.779265636309
Zone 13	0.707366022535
Zone 14	0.726942793322

Figure 7-1: Screen Captures of Measurement and Container Mobile Applications



Questionnaire Completed

Your questionnaire results were successfully recorded

Your anxiety risk score is 5, which indicates MILD anxiety risk.

Questionnaire Completed

Your questionnaire results were successfully recorded

Your diabetes risk score is 40, which indicates MEDIUM diabetes risk.

Questionnaire Completed

Your questionnaire results were successfully recorded

Your depression risk score is 0, which indicates NO depression risk.

Questionnaire Completed

Your questionnaire results were successfully recorded

Your sleep score is 15, which indicates POOR sleep quality.

Figure 7-2: Results Page of Questionnaire Mobile Apps

Chapter 8

Computational Models for Assessing Diabetes Stage and Predicting Risk

Using the tools and measurements described in the previous chapters, I have developed different algorithms to predict a subject's stage of diabetes. In this chapter, three different algorithms are proposed to classify data: a Bayesian algorithm with questionnaire data, a linear classifier with iris image data, and an integrated Bayesian model with both questionnaire and iris data.

8.1 Bayesian Network Algorithm with Questionnaire

8.1.1 Background and Prior Work

In this project we propose a Bayesian algorithm to screen for diabetes risk. Bayesian networks of varying levels of complexity have previously been used to diagnose different diseases. Naive Bayesian networks (NBN) are the simplest type of Bayesian network and have been proven useful in predicting different types of diseases, such as brain disease and breast cancer. A literature review that examined studies using

NBNs from 2005 to 2015 found that NBNs, though simple, can help improve physicians' decisions in disease diagnosis [57]. More specialized Bayesian networks have been created for disease diagnosis as well, such as a 94-variable network that was used for diagnosing liver diseases. The model was created in collaboration with medical experts and can be directly used by clinicians to help diagnose diseases [58].

In our own Mobile Technology Group, a Bayesian network was created by a previous student, Aneesh Anand, to diagnose pulmonary diseases. The Bayesian network outperformed logistic regression models previously used in the lab due to its ability to better handle missing data, flexibility, and ability to model the problem causally [3]. The model had high median AUC values above 0.9 for most disease groups and median sensitivity values also above 0.9. In this work we directly build off this work to build a screening algorithm better able to predict risk for different stages of diabetes.

8.1.2 Overview

Bayesian networks are acyclic directed graphs that aim to model causation. Each node in the network represents a unique random variable, and an edge in the graph from a parent to a child represents a conditional dependence between the variables. The joint probability distribution over the variables in the graph is represented in factorized form as follows, where $P(X_i|\,\Pi_{X_i})$ represents the probability table for each variable X_i [59]:

$$P(X_1, X_2, ..., X_N) = \prod_{i=1}^{N} P(X_i|\,\Pi_{X_i}) \tag{8.1}$$

The Bayesian network is specified by providing each variable's conditional probability tables given that variable's parents in the graph structure, or if it has no parents, the variable's prior probability distribution.

8.1.3 General Network Structure

We use a three-level Bayesian network that was developed by Aneesh Anand in our lab to diagnose pulmonary disease. The three layers are risk factors, disease, and symptoms. The general three-layer Bayesian network is displayed in Figure 8-1. In addition, Figure 8-2 shows an three-layer Bayesian model adapted for diabetes, with sample risk factors, symptoms and stages of disease (for clarity some edges were omitted). All the nodes in the graph represent binary random variables, taking on a value of 1 if the risk factor, disease or system is present, and 0 otherwise.

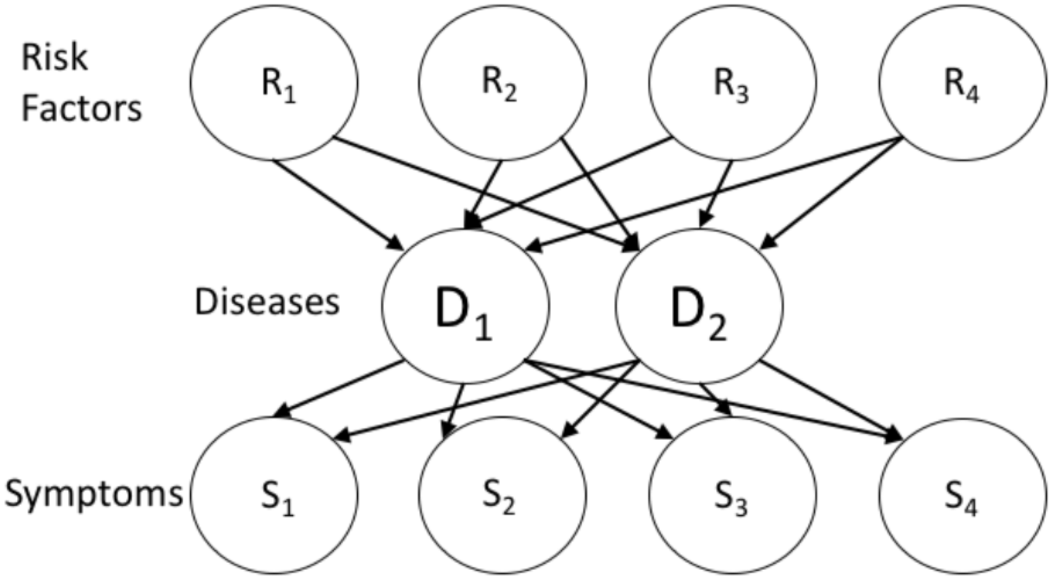


Figure 8-1: Three-Layer Generalized Bayesian Model [3]

Just as in Aneesh's thesis, the Bayesian network is a leaky noisy OR model with the following characteristics.

- Instead of creating a Conditional Probability Table (CPT) to account for all the different configurations of binary parent variables, we use a simple OR gate with the assumption that if any of the parents are present, then the child node will also be present, thus reducing the complexity of parameter estimation.

- In our simplified model, we have assumed that if a parent is present, the child will also be present, but there is a possibility that a positive parent node does not result in a positive child node. We account for this in the link probability through the following. Suppose X_i is a parent risk factor of Y, and c_i is the

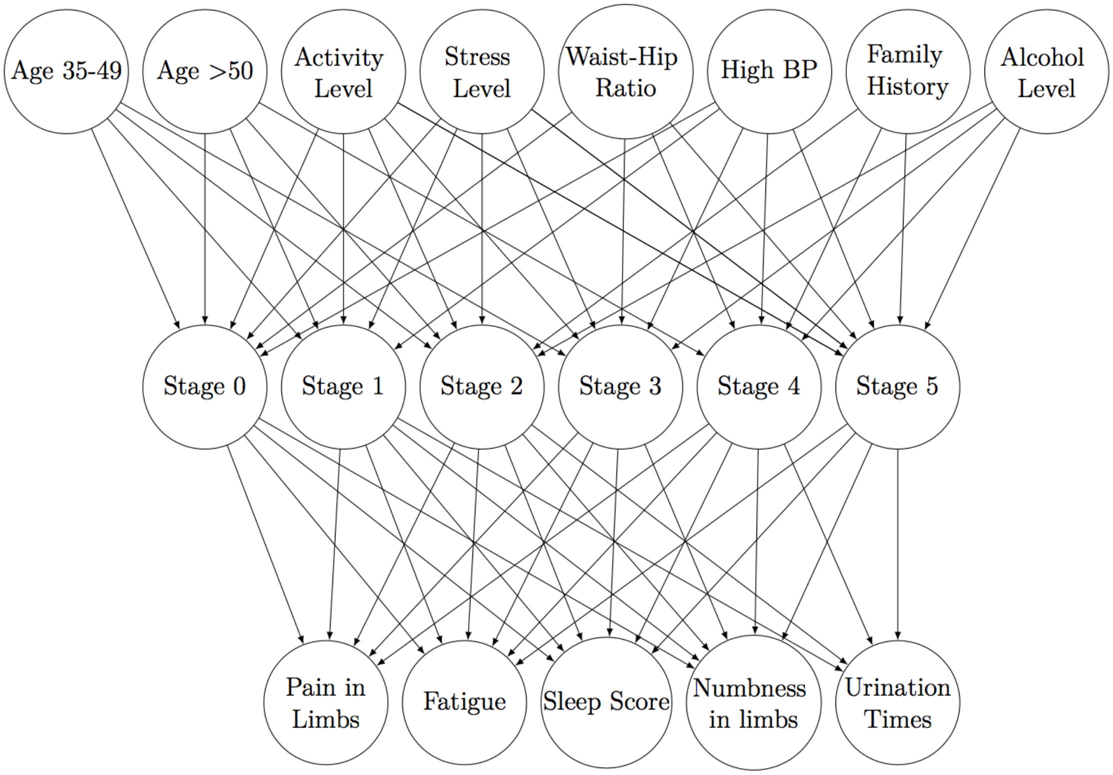


Figure 8-2: Bayesian Model with Non-Diabetic Stage (Stage 0) and Fives Stages of Diabetes

probability that X_i causes Y. Then $q_i = 1 - c_i$ is the probability that Y is not observed when X is present. Then we have the following equations where $I^+(x)$ is the set of X_i variables that are present [3].

$$P(y = 0|x) = \Pi^n_{i \in I^+(x)} q_i = \Pi^n_{i \in I^+(x)} (1 - c_i) \tag{8.2}$$

$$P(y = 1|x) = 1 - \Pi^n_{i \in I^+(x)} q_i = 1 - \Pi^n_{i \in I^+(x)} (1 - c_i) \tag{8.3}$$

- We also introduce the presence of a leak probability, which is equivalent to another parent node that explains why a child node is presenting even if all of the parent nodes are 0. This could happen if a disease or symptom results from an outside risk factor or disease not within the model. We add the link probability c_L in the following manner [3]:

$$P(y = 1|x) = 1 - (1 - c_L)\Pi^n_{i \in I^+(x)}(1 - c_i) \tag{8.4}$$

Once we have the model structure and parameters, we can calculate the probability of each stage of disease given the present risk factors and symptoms by using the Markov Chain Monte Carlo inference algorithm.

8.1.4 Model Simplification

As a first step towards creating a generalized Bayesian model for diabetes screening, we have decided to simplify the model by aggregating stages 0-3 into one node. This simplification has several advantages:

- By focusing on fewer stages, fewer patients are needed to create a robust model.
- Stages 0-3, particularly stages 1-3, of prediabetes are more difficult to distinguish because they are less well-defined and diagnosis is based on factors other than blood glucose.

For these reasons we have combined the initial stages of diabetes as a reasonable first step in modeling.

8.1.5 Design and Implementation

Using this new simplified design, we must not only assign risk factors and symptoms, but also parameterize prior probabilities of the risk factor and disease layers and the link probabilities connecting the risk factors to the diseases, and the diseases to the symptoms. The prior probabilities come from population-level studies, while the conditional probabilities are derived from the data. In this next section we lay out the binary nodes in all three layers of the network, as well as the parameters. The full network is illustrated in Figure 8-3, though some edges are omitted for a clearer image.

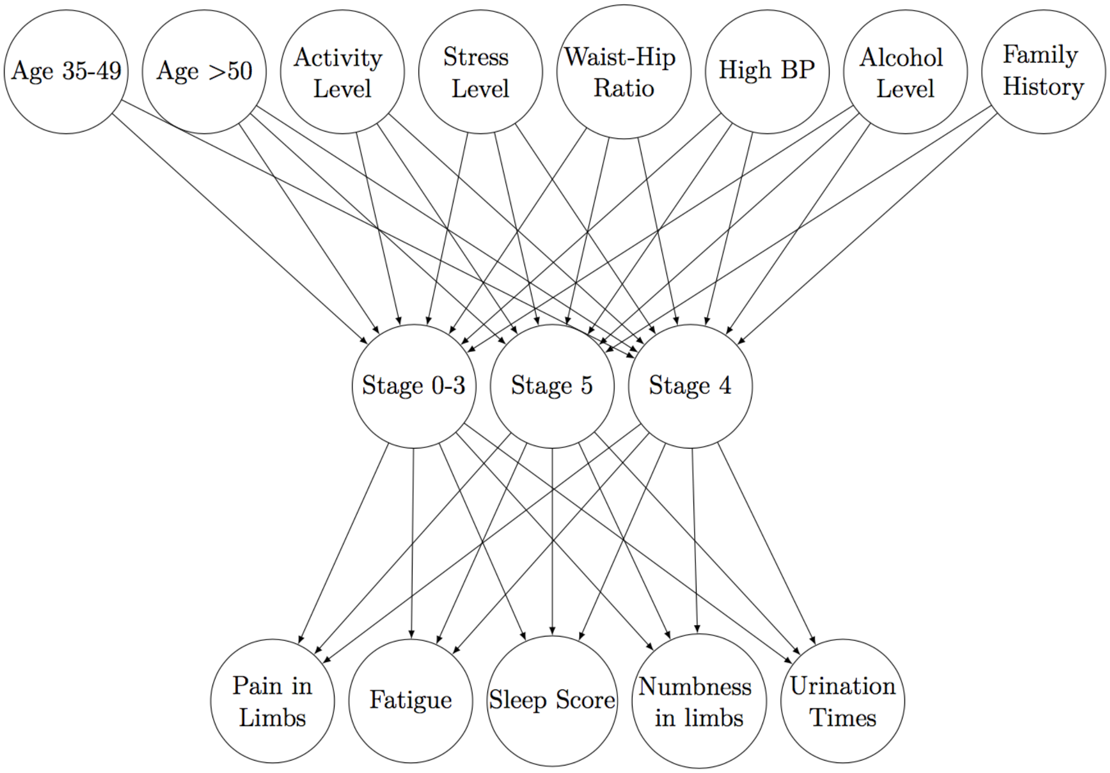


Figure 8-3: Bayesian Network for Stage 0-3, 4, and 5 of Diabetes

Risk Factors

The first layer in the network is the risk factors that contribute to a patient's diabetes risk score. We chose features included in the IDRS: age, activity level, waist-to-hip ratio, and family history of diabetes. In addition, we added features from the diabetes questionnaire created with our collaborators in India. Table 8.1 shows all of the risk factors used in the Bayesian network.

To compute the priors, we compared the values found in our study with several population-level studies measuring the same characteristics. In particular, we looked at large studies that also occurred in Mumbai or slum areas [60] [61] [62] [63]. This assessment of the prior is appropriate because our dataset is small and not an accurate representation of the target population. The subjects in our study were mainly recruited from people participating in diabetes health camps; therefore, our dataset includes a disproportionately high number of diabetics. The prior probabilities se-

Risk Factor	**Risk Factor Description**
Age	Age of patient divided into two categories (35-49, 50 and over)
Family History of Diabetes	Whether the patient has a family history of diabetes, either one parent or both parents
Stress Level	The patients stress level
Alcohol	The patients alcohol level
Activity level	The patients activity level, no activity, or mild activity
High blood pressure	Whether the patient has high blood pressure, which is defined as having a systolic blood pressure over 130 or diastolic blood pressure over 80
Waist-to-hip ratio	The patients waist-to-hip ratio divided into three categories: (under .89, .9-1, over 1)

Table 8.1: Risk Factors for Bayesian Network

lected are shown in 8.2.

To calculate the link probabilities between the risk factors R and the diseases D, we calculate the following from our data set:

$$c_r = \frac{N(R, D)}{N(R)} \tag{8.5}$$

$N(R, D)$ is the number of subjects in the training set with both disease D and risk factor R, and $N(R)$ is the number of total subjects with risk factor R.

Risk Factor	**Risk Factor Description**
age 35-49	.35
age 50 and over	.65
Family history of diabetes either one parent or both parents	.11
Stress Level	.3
Alcohol	.13
no activity	.2
moderate activity	.6
High blood pressure	.23
waist-to-hip ratio $\leq$ 89	.1
waist-to-hip ratio .9-1	.5
waist to hip ratio $>$ 1	.35

Table 8.2: Prior Probabilities for Risk Factors

Diseases

The diseases included in the model are stage 0-3 of diabetes, stage 4 of diabetes and stage 5 diabetes. It is important to note that the node labeled stage 0-3 includes individuals that do not currently have the disease (stage 0) as well as individuals in the prediabetic stages (stages 1-3).

Stage of Diabetes	**Prior Probability**
Stage 0-3	.76
Stage 4	.15
Stage 5	.02

Table 8.3: Prior Probabilities for Diabetes Stages

Symptoms

The symptoms and descriptions used in the final layer of the Bayesian network are listed below in Table 8.4.

Symptom	**Symptom Description**
Fatigue	Whether the patient experiences fatigue
Numbness	Whether the patient experiences numbness in their limbs
Pain in Limbs	Whether the patient experiences pain in their limbs
Sleep Quality	Whether the patient experiences poor sleep quality
Urination times	Whether the patient urinates frequently: over five times a day

Table 8.4: Symptoms for Bayesian Network

Given a disease D and a symptom S, the link probabilities are calculated similarly to the first layer with the equation below, where $N(S, D)$ is the number of subjects with both the disease and symptom, while $N(D)$ is the number of subjects with the disease.

$$c_D = \frac{N(S, D)}{N(D)} \tag{8.6}$$

8.2 Iris Analysis Algorithm

Another algorithm used in our screening platform was devoted to iris analysis. While the effectiveness of iris analysis is still debated within the scientific community, we have decided to continue exploring this measurement based on an encouraging previous study done in our group [2]. We discuss this algorithm in the following section.

8.2.1 Background and Prior Work

As mentioned in Section 4.1, the iris analysis algorithm was developed by a prior student in the Mobile Technology Group, Tania Yu. In her algorithm, she first preprocessed and segmented the target range of the iris (from 24 possible ranges). The 24 different segments of eye correlate to different regions in the body. Of particular interest are regions 18 and 21, which correspond to the pancreas. After segmenting the target range, Tania extracted several different texture and color features from the iris, and used linear and non-linear classifiers to classify the images. We use her top-performing method of extracting features, local Gabor binary pattern histogram sequence (LGBPHS), on our image dataset. In addition, while Tania used a binary classifier to classify her image as diabetes or not diabetes, we use a multiclass classifier to categorize which stage of diabetes the patient is in based on his or her iris image.

8.2.2 Multiclass Classification

Multistage diabetes prediction can be posed as a multiclass classification problem because we are trying to determine the stage of diabetes given a specific iris image. To do this we take a one-vs-all strategy, in which we fit one classifier per class. This is a commonly used strategy because of its efficiency (only n-classifiers are needed) and interpretability.

The classifiers that we use are linear classifiers in order to avoid overfitting. The linear classifier used is a support vector machine (SVM). The result of this analysis using a test data set is presented in Chapter 9.

8.3 Bayesian Network with Diabetes Questionnaire and Iris Analysis

In this model, we use the classification provided by the iris analysis algorithm and add it to the Bayesian network in the symptom layer to help predict diabetes stage; the risk factor and disease layers stay the same. The nodes added are binary nodes in the symptom layer, one each for stage 0-3, stage 4 and stage 5. The full Bayesian network is shown in Figure 8-4.

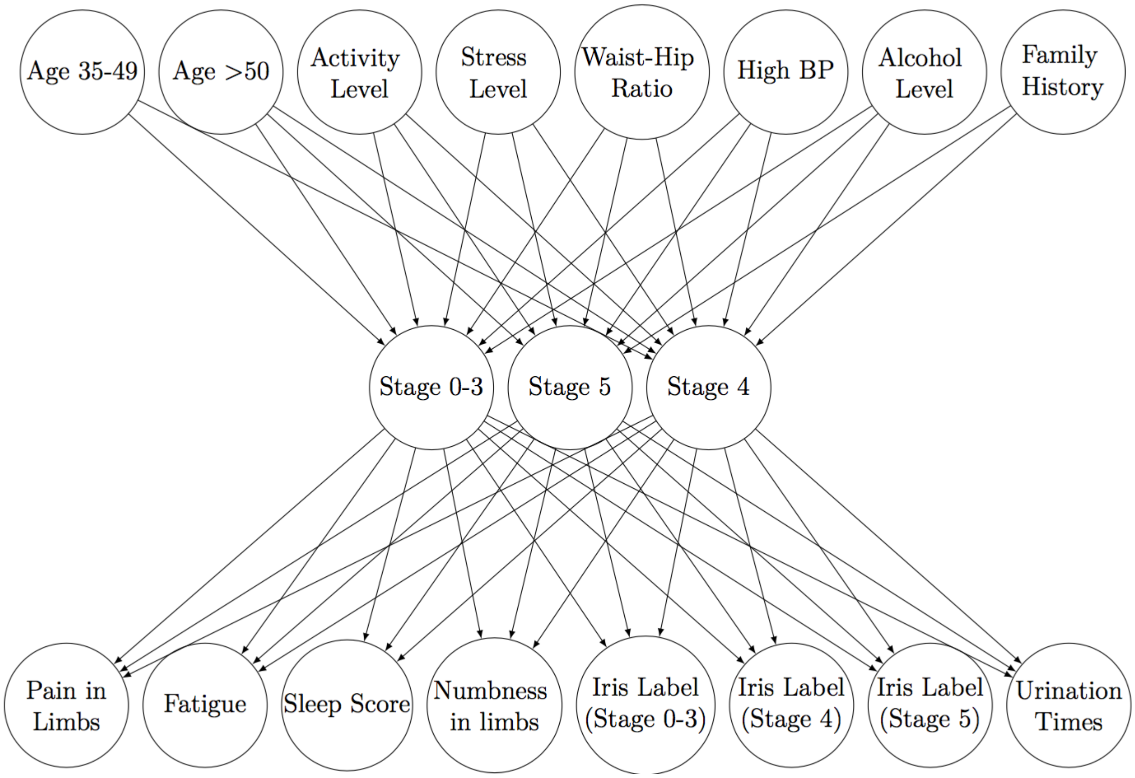


Figure 8-4: Bayesian Network with Iris Classification for Stage 0-3, 4, and 5 of Diabetes

We train a linear SVM classifier with LGBPHS features from region 21 over all available, labeled images collected from subjects. Then for each test subject, we use the trained classifier to classify the subject's iris images. The most frequent classification is used to classify the subject overall. Then the value of the Bayesian symptom node is observed if the iris classifier classifies the subject as in that stage. It is important to note that not all of the subjects have iris data available, so for these patients, none of the iris classification nodes are observed.

8.3.1 Modified Symptoms Layer

The only modified layer is the symptoms layer. Table 8.5 shows the added symptoms:

Symptom	**Symptom Description**
Fatigue	Whether the patient experiences fatigue
Numbness	Whether the patient experiences numbness in their limbs
Pain in Limbs	Whether the patient experiences pain in their limbs
Sleep Quality	Whether the patient experiences poor sleep quality
Urination times	Whether the patient urinates frequently: over five times a day
stage 0-3 Iris Classification	Whether the iris classifier classifies the subject as being in stage 0-3
Stage 4 Iris Classification	Whether the iris classifier classifies the subject as being in stage 4
Stage 5 Iris Classification	Whether the iris classifier classifies the subject as being in stage 5

Table 8.5: Additional Symptoms for Bayesian Network with Iris Classifier

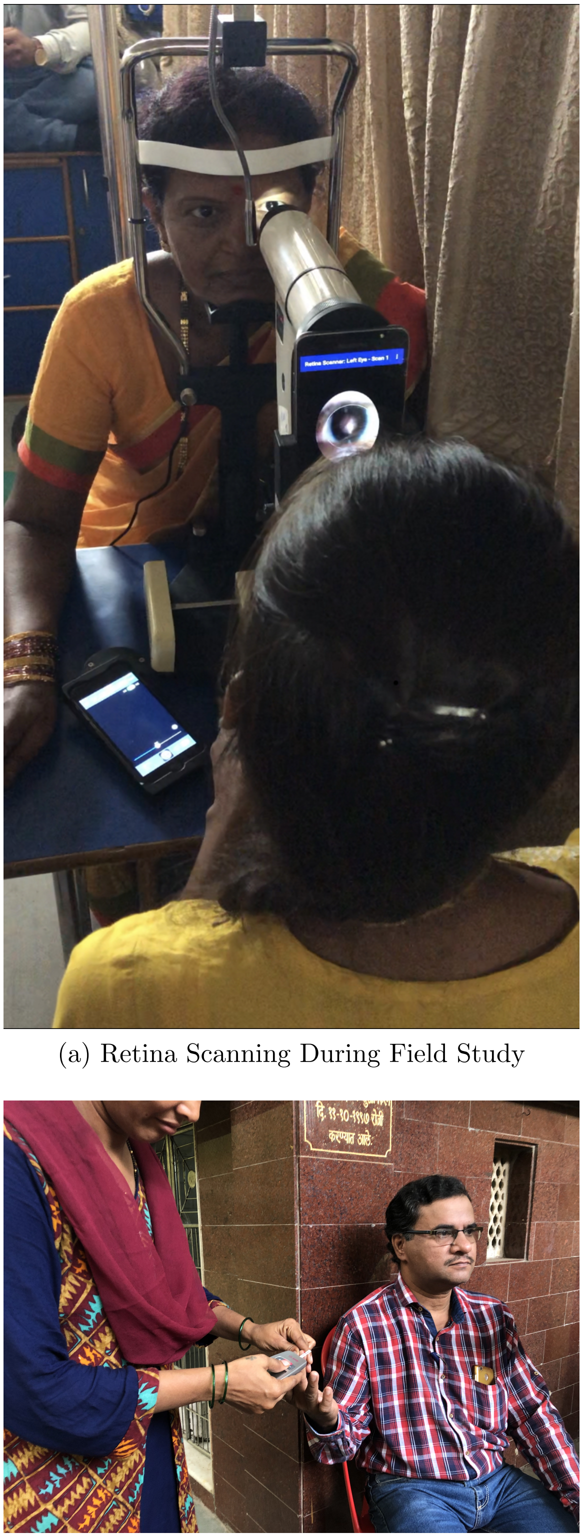
Chapter 9

Field Study Description and Preliminary Results

9.1 Study Design

Data collection is ongoing at two different sites in India: the Aditya Jyot Foundation for Twinkling Little Eyes (AJFTLE) in Mumbai, and the Swami Vivekananda Yoga Anusandhana Samsthana (S-VYASA) site in Bangalore. At AJFTLE all the measurements except for the psychology-related questionnaires (PSQI, GAD-7, PHQ-9, PTQ-15) are taken since this site sees more patients, and does not have the bandwidth to administer the psychology questionnaires to each patient. At S-VYASA, all non-invasive measurements except for the vision test are taken. In order to validate our solution against standard screening tools and to help develop our machine learning algorithms, we also administer blood glucose tests at both sites using a standard Alere glucometer and perform retina scanning using the Remidio Fundus camera.

Subjects over the age of 35 in different stages of diabetes were chosen for this study. At AJFTLE, subjects were recruited from among regular visitors to an urban health clinic where free eye screening services are provided to low-resource individuals from surrounding slum areas. In addition, a mobile van was deployed to hard-to-access urban areas to provide screening services to the surrounding population, so subjects were also recruited from these van visits. At S-VYASA, subjects were recruited from visitors to the university yoga hospital on campus. The protocol received approval from the Institutional Review Boards at MIT, S-VYASA, and AJFTLE. At the time of writing, data from 42 subjects have been collected. The study is ongoing, with a goal of collecting data from 500 subjects. Figure 9-1 shows data collection occurring at one of the sites in Mumbai.



(a) Retina Scanning During Field Study

(b) Finger Prick Blood Glucose Test

Figure 9-1: Data Collection in Mumbai

9.2 Standards for Model Training and Validation

For the purpose of developing, training and validating our model, we included in our study some standard measurements which are described below.

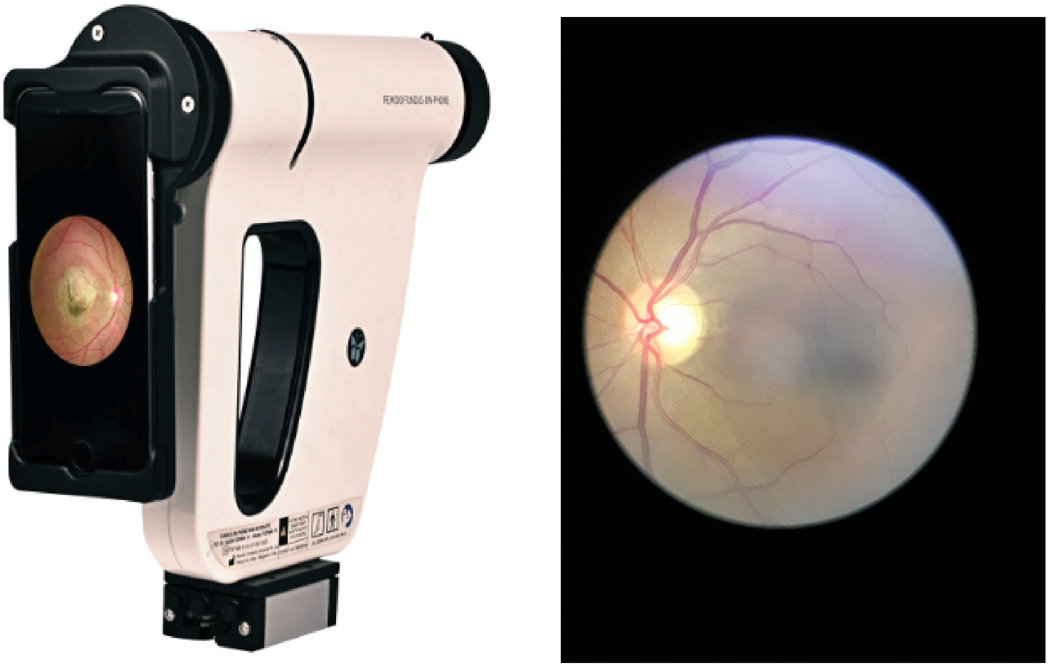
9.2.1 Blood Glucose Measurements

At both the AJFTLE and S-VYASA sites, we are measuring RBS using a standard Alere glucometer. At S-VYASA, additional blood tests are analyzed at the yoga university's laboratory. For some patients at both sites, HbA1c is also measured, though we are still working on administering this test to all patients to be able to use this data during analysis.

9.2.2 Retina Imaging

To validate our model in the later stages of diabetes, we have chosen a fundus camera to image the retina for signs of diabetic complications such as diabetic retinopathy, which occurs in stage 5 of the disease. In collaboration with AJFTLE, we selected the Remidio Fundus Camera which is manufactured in Bangalore, India and is shown in Figure 9-2. This instrument makes use of an Apple iPhone 6 to capture images and is capable of operating in both non-mydriatic (non-dilated) and mydriatic (dilated) modes.

For our data collection, we employed the mydriatic setting, which requires the patient's pupils to be dilated before the image is taken. For each eye, four images are captured: anterior chamber, optic disc, macula and posterior pole, allowing us to image a wide range of the retina in order to check for signs of diabetes-related disease. A sample posterior pole image is shown in Figure 9-2.



(a) Remidio Fundus Camera

(b) Posterior Pole Retina Image Collected with Remidio

Figure 9-2: Retina Measurement

9.3 Study Protocol

After the subject agrees to the study and signs the consent form, the following measurements are taken:

- **Diabetes questionnaire**: This questionnaire asks about general diabetes risk factors and symptoms. The technician administering the questionnaire also takes several standard measurements, such as height, weight and blood pressure as part of this measurement.
- **Pyschology-related questionnaires**: If the subject is at the S-VYASA site, they are administered several psychology-related questionnaires.
- **Iris Images**: The iris imaging device is used to take two pictures each of the patient's left and right eyes, for a total of four iris images per patient.
- **Thermal Images**: The Seek Thermal camera is then used to capture two thermal images of the the face.

- **Vision Test**: At the AJFTLE site, the subject will take a standard vision test, which consists of reading a chart at a distance of six meters. The results of the test are recorded in the Android mobile app.

- **Blood Test**: At both sites, a finger prick blood test is read by an Alere glucometer and used to record the RBS value. Additional blood glucose tests such as HbA1c may also be recorded.

- **PPG**: The photoplethysmographic waveform is recorded using the Android mobile phone.

- **Retina Images**: Following these steps, the patient's eyes are dilated for retinal imaging. Though the previous measurements can be collected in any order, retinal scanning happens last after all of the other measurements have been recorded because we cannot collect iris images once the pupil has been dilated. A total of four retina images are collected per eye, eight images in total.

9.4 Participant Demographics

Of the 42 subjects, complete questionnaire and blood test measurements were available for 29 of the subjects. Of these 29 subjects, 13 were female and 16 were male. Other patient demographic information is show in Table 9.1.

The subjects were classified into a stage of diabetes using the RBS value and the answer to a question about current medications from the diabetes questionnaire. If the subject had an RBS below 200 mg/dl without the aid of any tablets or insulin, they were classified as being in stage 0-3. If the patient had an RBS above 200 mg/dl, which is the RBS cut-off for diabetes, they were either classified as being in Stage 4 or Stage 5 . If the subject was taking insulin for treatment, or their RBS was above 260 mg/dl, they were classified as being in Stage 5 of diabetes. This cutoff was chosen based off the normal RBS value of 140 mg/dl. However, in future tests more comprehensive blood tests, such as HbA1c, should be used to make this classification. The subjects not classified as being in stage 0-3 or stage 5 were classified as stage 4.

Patient Demographics					
Gender	Male	Female			
	16	13			
Education Level	None	Primary	Secondary	High School	University
	9	1	1	15	2
Living Area	Urban Area	Town	Rural Area		
	29	0	0		
Marital Status	Single	Married			
	1	28			

Table 9.1: Characteristics of subjects

These were subjects who had an RBS value between 200-260, or were taking treatments such as tablets to manage their blood sugar levels. The number of patients from each stage of diabetes is shown in Table 9.2.

	Stage 0-3	**Stage 4**	**Stage 5**
Number of Subjects	6	17	6

Table 9.2: Number of Subjects in Stages of Diabetes

9.5 Bayesian Algorithm with Diabetes Questionnaire

We used the PyMC3 Python library to implement our Bayesian network model. We reserved 20% of the data for training, while 80% was used to train the network. Similar to the iris data, the split was stratified so that equal proportions of stage 0-3, stage 4, and stage 5 subjects were in the training and test data set.

We evaluated the model using 10 iterations of randomly selected test sets. We input the observed risk factors and symptoms into the Bayesian network, then we ran 1500 iterations of MCMC sampling iterations to infer the marginal probabilities of the three disease stages. For each subject in the test set, the model was designed to output a prediction for each of the three stages. In the current configuration of our model, the subject is classified as being in the stage with the highest probability.

9.5.1 Preliminary Results

For each of the 10 evaluations of randomly selected test sets, we calculate the sensitivity and specificity. We also compute the receiver-operating-characteristic (ROC) curve and calculate the area under this curve (AUC) and report these values. Figure 9-3 shows an ROC curve for the different stages of diabetes from one of the test set iterations. We report the sensitivity, specificity, and AUC score in Table 9.3. The results are taken as a mean over the 10 test sets, while the ranges indicate the maximum and minimum values observed.

By examining the results for the Bayesian network with questionnaire data, we see that the model best classifies stage 0-3 and stage 5 of diabetes with AUC scores of 1.0 and .7 respectively. However the model shows very low sensitivity for both these models, and low specificity for classifying stage 4.

	Bayesian Network with Questionnaire		
	AUC	Sensitivity	Specificity
Stage 0-3	.7 (.7-.7)	0.0 (0.0-0.0)	.78 (.6-.8)
Stage 4	.675 (.66-.72)	.63 (.33-.66)	0.0 (0.0-0.0)
Stage 5	1.0 (1.0-1.0)	0.0 (0.0-0.0)	1.0 (1.0-1.0)

Table 9.3: Mean AUC, Sensitivity, and Specificity Values for Bayesian Network

9.5.2 Sensitivity Analysis

The purpose of a sensitivity analysis is to show which risk factors and symptoms have the most impact on the model's predictions. This is important in informing clinicians and patients of which risk factors and symptoms most impact the result

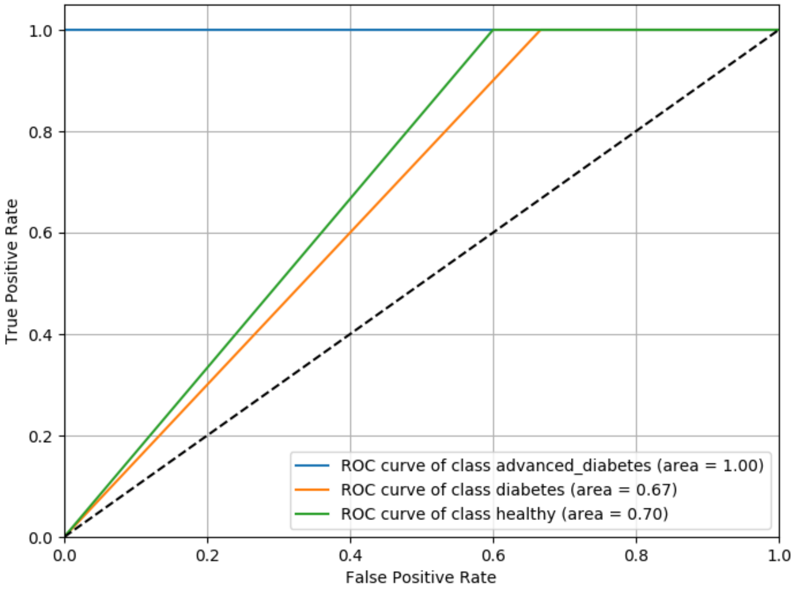


Figure 9-3: ROC curve for Bayesian Network

of the algorithm, so as to justify the screening classification and allow proper action to be taken. The sensitivity analysis is conducted by using cross-entropy as a cost function. To calculate the effect of evidence e on variable Y, we use the following equation [3]:

$$H(P(Y|e);P(Y)) = \sum\nolimits_i P(y_i|e)\frac{P(y_i|e)}{P(y_i)} \tag{9.1}$$

To compute this value we first find the distribution over the variable Y, in this case the different stages of diabetes, by running 5000 MCMC iterations with no observed evidence to get the distribution over each stage. We treat a positive observed risk factor or symptom as one piece of evidence, and a negative value as another piece, so we run 5000 MCMC distributions with each piece of evidence individually observed.

For each stage of disease, we report the risk factors or symptoms with the highest cross-entropy. The cross-entropy values are separated into positive and negative diagnoses, where negative means that the presence of the observed evidence led to less risk for that stage and positive means it led to higher risk for that stage.

Stage 0-3

The cross-entropy values of stage 0-3 are illustrated in Table 9.4. Of all the stages, the cross-entropy scores for stages 0-3 are the lowest. This stage may be hardest for the model to classify because it encompasses two categories of individuals: those who do not have diabetes, and those who are prediabetic. According to the features with the highest cross-entropy, the model is regarding the patients as more similar to prediabetics and diabetics, rather than non-diabetics. This is evident because the risk factors of diabetes, such as positive pain in limbs, are positively influencing the model to classify a subject in stages 0-3. We also see that some risk factors, like urination times over 5, negatively affect the model if they are not present. A low number of urination times would indicate a non-diabetic subject, or stage 0, but since this is not the case here, we have further evidence that the model is more closely predicting stages 1-3 rather than stage 0.

Diagnosis	Feature	State	Entropy
Negative	Urination Times over 5	0: Absent	.01
Positive	Numbness In Limbs	1: Present	.009
Positive	Waist-to-hip ratio 0.9-1	1: Present	.008
Negative	Fatigue	0: Absent	.007
Positive	Alcohol	1: Present	.006

Table 9.4: Cross-Entropy Values of Stage 0-3

Stage 4

The cross-entropy values of stage 4 are illustrated in Table 9.5. The features with the highest cross-entropy values for stage 4 align with common risk factors and symptoms of diabetes: urination times, numbness in limbs, waist-to-hip ratio and fatigue, although numbness in limbs is typically a symptom of stage 5 diabetes. In addition, the model found that drinking alcohol was also a significant predictor for classifying a subject in stage 4 of diabetes.

Diagnosis	Feature	State	Entropy
Negative	Urination Times over 5	0: Absent	.068
Negative	Fatigue	0: Absent	.031
Positive	Waist-to-hip ratio 0.9-1	1: Present	.026
Negative	Waist-to-hip ratio 0.9-1	0: Absent	.022
Positive	Age 35-49	1: Present	.022
Positive	Family History	1: Present	.021

Table 9.5: Cross-Entropy Values of Stage 4

Stage 5

The cross-entropy values for stage 5 of diabetes are the highest among the different stages. These values are displayed in Table 9.6. In particular, the model found that the presence of poor sleep quality, a waist-to-hip ratio over 1 and stress were all indicators that a subject should be classified in stage 5 of diabetes. The absence of fatigue and high urination times made a subject less likely to be classified as stage 5. It is notable that we did not see typical symptoms of advanced diabetes among these features, such as pain in limbs or numbness in limbs. However, waist-to-hip ratio over 1 is the risk factor with the highest cross-entropy and this risk factor is also a significant predictor of diabetes risk in the IDRS.

Diagnosis	Feature	State	Entropy
Positive	Poor Sleep Quality	1: Present	.247
Negative	Fatigue	0: Absent	.162
Negative	Urination Times over 5	0: Absent	.065
Positive	Waist-to-hip ratio over 1	1: Present	.025
Positive	Stress	1: Present	.023

Table 9.6: Cross-Entropy Values of Stage 5

9.6 Iris Algorithm Analysis

Of the 42 subjects, 41 had iris data collected, but not all these subjects had complete questionnaire and RBS data. Of these subjects, we were able to determine the stage of diabetes for 25 subjects based on their RBS and questionnaire answers. The number of subjects in each stage were as follows: 4 of the subjects were in stage 0-3, 17 had stage 4 diabetes, and 4 had stage 5 diabetes.

9.6.1 Initial Iris Analysis

One of the initial analyses performed was plotting the extracted LGBPHS features against the RBS value to see if there was any correlation between the two. In most cases, such as for region 18, there was no clear correlation as demonstrated in Figure 9-4. Changes in the iris are a result of long term changes, whereas blood sugar levels drastically vary over time—such as before and after meals. Therefore, it is not surprising to see a lack of correlation between iris features and RBS.

However, there was a slight negative correlation between region 4 and RBS values as shown in Figure 9-5. Region 4 corresponds to thyroid levels, and indeed, low thyroid levels are indicative of a high BMI, which is a risk factor of diabetes. To further see if region 4 can be used to predict which stage of diabetes an individual is in, we include it in our regions of interest for the next section in which we compute preliminary results.

9.6.2 Preliminary Results

We evaluate the performance of the LGBPHS filter using a One vs. All multiclass linear SVC classifier on three different regions of the iris: region 4, region 18 and region 21. Evaluation was conducted for 10 iterations with randomly selected test sets stratifed so that an equal proportion of stage 0-3, stage 4 and stage 5 patients were in the test and training set. The metrics reported are: the AUC score, the sensitivity and the specificity for each of the three stages. Because one or more of the target regions are occluded in some of the images, we only show results from images

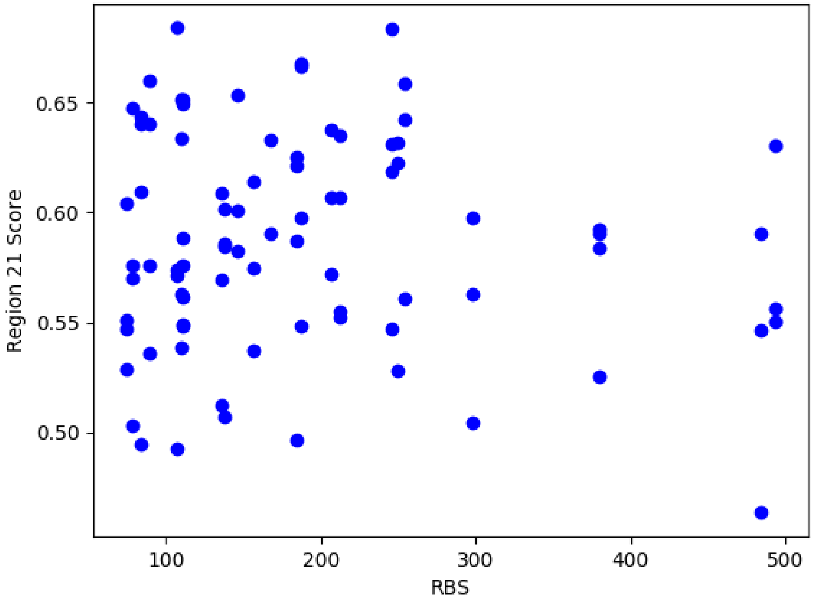


Figure 9-4: Region 18 Iris Features vs. RBS Values

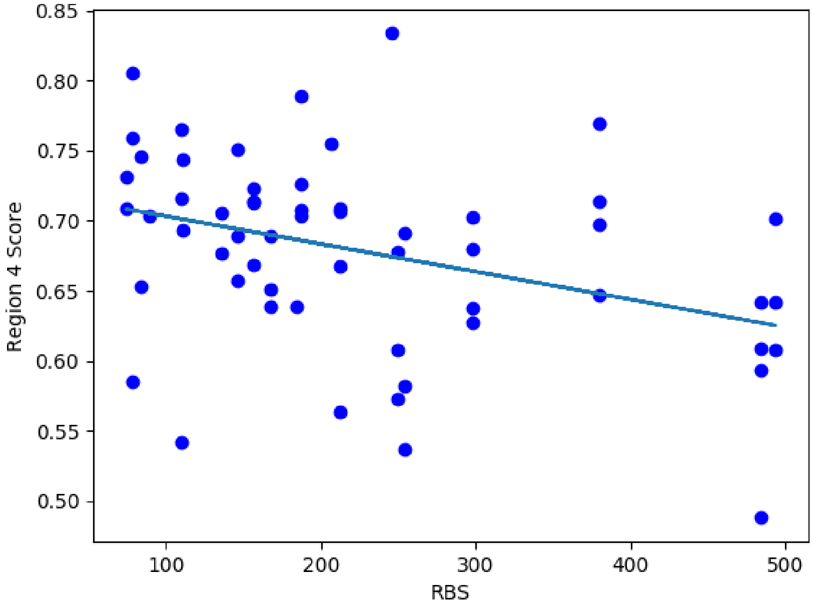


Figure 9-5: Region 4 Iris Features vs. RBS Values

in which the specific regions, i.e. 4, 18, or 21, of the iris are visible. These results are shown in Tables 9.7, 9.8 and 9.9 respectively and a sample ROC curve from a random test set iteration is shown in Figure 9-6.

	LGBPHS		
	AUC	Sensitivity	Specificity
Stage 0-3	.625 (.31-.69)	0.0 (0.0-0.0)	1.0 (1.0-1.0)
Stage 4	.56 (.55-.58)	.99 (.91-1.0)	0 (0.0-0.0)
Stage 5	.49 (.48-.5)	0.0 (.0-.4)	.99 (.92-1.0)

Table 9.7: Mean AUC, Sensitivity, and Specificity scores for Region 4

	LGBPHS		
	AUC	Sensitivity	Specificity
Stage 0-3	.50 (.31-.66)	0.0 (0.0-0.0)	1.0 (1.0-1.0)
Stage 4	.675 (.56-.58)	.91 (.89-1.0)	.13 (.0-.29)
Stage 5	.60 (.6-.65)	.18 (.0-.4)	1.0 (1.0-1.0)

Table 9.8: Mean AUC, Sensitivity, and Specificity scores for Region 18

	LGBPHS		
	AUC	Sensitivity	Specificity
Stage 0-3	.48 (.32-.69)	0.0 (0.0-0.0)	1.0 (1.0-1.0)
Stage 4	.69 (.47-.95)	1.0 (1.0-1.0)	0.0 (0.0-0.0)
Stage 5	.99 (.9-1)	0.0 (0.0-0.0)	1.0 (1.0-1.0)

Table 9.9: Mean AUC, Sensitivity, and Specificity scores for Region 21

9.6.3 Discussion

After calculating the mean AUC score, sensitivity and specificity, Region 4 performed the best at classifying stage 0-3 of diabetes while Region 21 had the best AUC scores in classifying stage 4 and stage 5 of diabetes. Region 18, however, had slightly better sensitivity and specificity values than the other two regions. For our Bayesian network with iris algorithm, we use the results from our highest scoring individual region, 21.

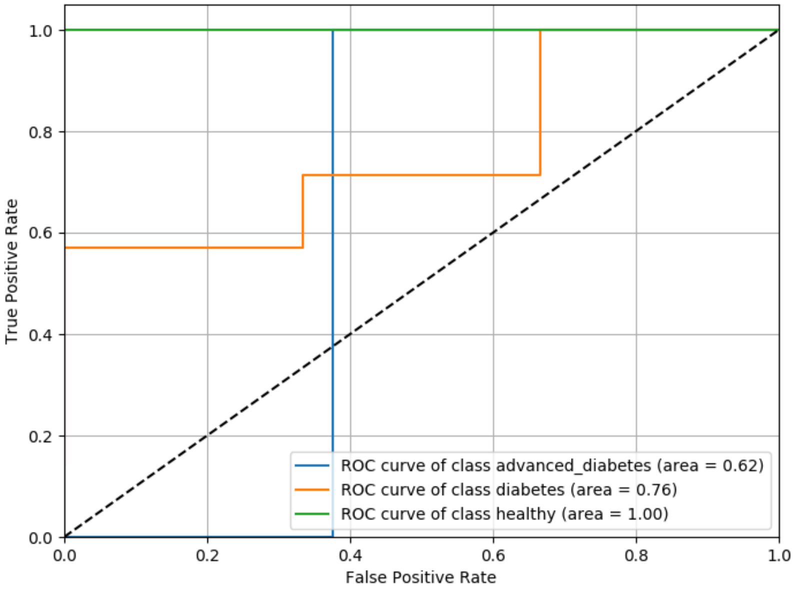


Figure 9-6: Mean ROC curve for Region 21 from Iteration of Randomly Selected Test Set

In the future, we would like to use a combination of regions, but this can cause some images to be excluded from classification if some of the target regions are occluded. In the future, by setting better guidelines for iris image data collection, there is a possibility of using multiple regions together for classification.

9.7 Bayesian Algorithm with Iris Images and Questionnaire

9.7.1 Preliminary Results

Similar to Section 9.5, we use PyMC3 to run 1500 iterations of MCMC sampling iterations and infer the marginal probabilities of the different disease stages. We again evaluated the model using 10 iterations of randomly selected test sets and classified the evaluated test subjects as belonging to the stage with the highest output probability. The mean sensitivity, specificity and AUC score are reported in Table

9.10, and a sample ROC curve from a test iteration is shown in Figure 9-7.

The addition of iris images improves the Bayesian model from the previous design, which only uses questionnaire data; this is evident from comparing the AUC scores, which increase for each of the stages, as does the sensitivity. The specificities for stage 4 and stage 5 are also increased in this new model with iris classification, though the specificity for stage 0-3 is slightly decreased.

	Bayesian Network with Questionnaire		
	AUC	Sensitivity	Specificity
Stage 0-3	1.0 (1.0-1.0)	1.0 (1.0-1.0)	.68 (.6-.8)
Stage 4	1.0 (1.0-1.0)	.667 (.667-.667)	.866 (.667-.1)
Stage 5	1.0 (1.0-1.0)	0.5 (.5-.5)	1.0 (1.0-1.0)

Table 9.10: Mean AUC, Sensitivity, and Specificity Values for Bayesian Network with Iris Classification

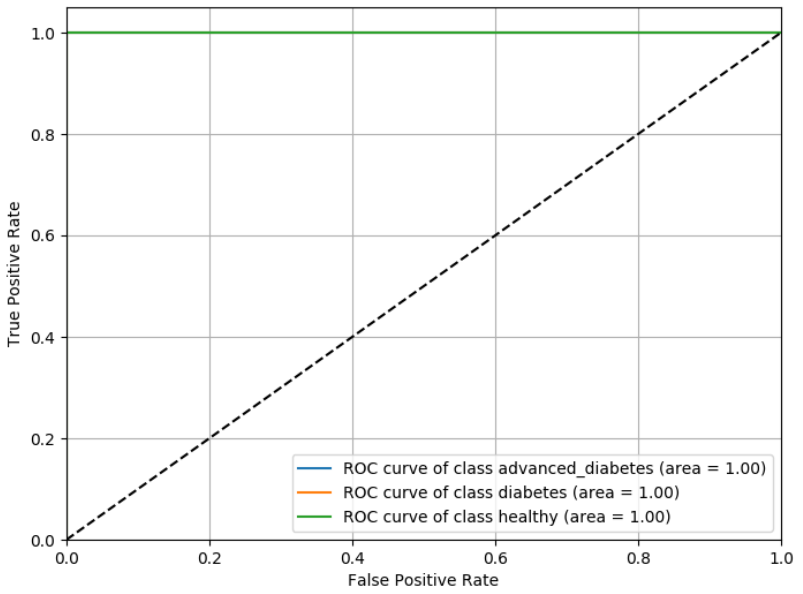


Figure 9-7: ROC curve for Bayesian Network with Iris Classification

9.7.2 Sensitivity Analysis

We compute the cross-entropy values as detailed in equation 9.1 to conduct the sensitivity analysis for the Bayesian nework with questionnaire data and iris images.

Stage 0-3

With the addition of the iris data, the cross-entropy values are higher than with just the questionnaire data. The values are displayed in Table 9.11. We see some of the same features as before, such as urination times, fatigue and waist-to-hip ratio, though this time, a negative observation of waist-to-hip ratio has higher cross-entropy than a positive observation. We can also see the significance of the iris data in the top features: a positive stage 0-3 iris classification observation has the highest cross-entropy value. Less clear is the effect of not observing an iris classification of stage 4, which also reduces the classification probability of stage 0-3.

Diagnosis	Feature	State	Entropy
Positive	Iris Classification (Stage 0-3)	1: Present	.023
Negative	Fatigue	0: Absent	.012
Negative	Urination Times over 5	0: Absent	.011
Negative	Iris Classification (Stage 4)	0: Absent	.01
Negative	Iris Classification (Stage 0-3)	0: Absent	.009
Negative	Waist-to-hip ratio 0.9-1	0: Absent	.007

Table 9.11: Cross-Entropy Values of Stage 0-3

Stage 4

The cross-entropy values of stage 4 are illustrated in Table 9.12. Again we see that the cross-entropy scores are higher with the addition of iris data to the model, and we also see some of the same features from the previous stage 4 sensitivity analysis: fatigue, urination times over 5, family history and waist-to-hip ratio. The feature with the highest cross-entropy is an iris classification for stage 4; if it is absent, it reduces the probability of being classified as this stage. In addition, we see many of the same top features as in stage 0-3 such as waist-to-hip ratio, fatigue, and urination times.

Diagnosis	Feature	State	Entropy
Negative	Iris Classification (Stage 4)	0: Absent	.310
Negative	Urination Times over 5	0: Absent	.076
Negative	Fatigue	0: Absent	.042
Positive	Family History	1: Present	.026
Positive	Waist-to-hip ratio 0.9-1	1: Present	.026

Table 9.12: Cross-Entropy Values of Stage 4

Stage 5

The cross-entropy values of stage 5 are illustrated in Table 9.13. Stage 5, as before, has the highest cross-entropy values. It also has the most difference between the cross entropy values of its top features. An iris classification feature, whether the subject is classified as being in stage 0, is the top feature in this class, again showing the impact of the iris data to the Bayesian network. And just like in the previous sensitivity analysis, poor sleep quality remains an important feature.

Diagnosis	Feature	State	Entropy
Positive	Iris Classification (Stage 5)	1: Present	.5025
Positive	Poor Sleep Quality	1: Present	0.248
Negative	Fatigue	0: Absent	.152
Negative	Urination Times over 5	0: Absent	.069

Table 9.13: Cross-Entropy Values of Stage 5

9.7.3 Discussion

In this model we can see more clearly than before that the same pieces of information are relevant when classifying the different stages of diabetes. We know this because certain features, such as urination times over 5, have top cross-entropy scores across the different stages. This is not surprising because the same diabetic symptoms will be applicable in all stages of the disease, but level of severity varies as an individual progresses through the different stages. One limitation of our model is that our features are discrete, not continuous, making fine differentiation difficult when trying to classify between similar stages.

Chapter 10

Conclusion

10.1 Contributions of Work

10.1.1 Data Collection and Assessment Platform

In this thesis, I discussed the creation of the diabetes server, which allows for non-invasive measurements in this study to be uploaded to a cloud server, analyzed and returned back for display on a mobile application or web front-end. The platform allows for non-invasive measurements to be collected by health workers and trained technicians in the field, uploaded to the server, and then labeled by a clinician for patient diagnosis and machine learning purposes. The measurements can also be analyzed in the server to return real-time results at the point-of-care for clinicians and patients.

The web application allows clinicians to monitor patients in their study, view and label images and add diagnoses. It also allows for additional functionalities such as bulk download of measurements. The addition of this front-end will greatly aid adoption of our platform among clinicians.

10.1.2 Algorithms to Classify Diabetes Stage Classification

In order to explain diabetes stage classification, I first described the six different stages of diabetes pathogenesis. Then I discussed three different algorithms to classify a subject into one of these stages: using the questionnaire data in a Bayesian network, using iris image features with a linear classifier and combining the iris classifier with the Bayesian network.

In this thesis, I laid out the design and implementation of these algorithms, and then evaluated the models on preliminary data collected from an ongoing field study in India. With the questionnaire alone, I was able to achieve an AUC value of 1.0 for stage 5, and .675 and .7 for stage 0-3 and stage 4 respectively. Then by adding the questionnaire data, I was able to achieve AUC values of 1.0 for all three different stages as well as improve the mean sensitivity and specificity values.

Though these findings are still preliminary, and more data is needed to verify the results, we are encouraged by results from our preliminary model. The current work provides a good foundation from which to add more stages of diabetes and non-invasive measurements in the future.

10.2 Future Work

The first main area for future work is to update the models and results as data collection progresses. Additional data will make it possible to expand the model so that stages 0-3 each have their own disease nodes in the Bayesian network. In future iterations of the model when stage 0-3 is divided into individual nodes, the additional stages will be added to the disease layer of the network alongside the existing stages, as shown in Figure 8-2. The current risk factors and symptoms will not need to be modified as the Bayesian network can discern which observed variables most affect which stages, but the addition of more stages may require more risk factors or symptoms in the model for accurate classification.

Experimenting with which factors and symptoms should be included in the model offers another interesting avenue for future work. Since we saw good results from adding the multiclass linear SVM's iris classification, future work could try directly using different iris feature representations as symptoms in the Bayesian network. The improvement in results from using the iris analysis also encourages the addition of other non-invasive measurements to the Bayesian model. Some measurements, like retina images, will be used to predict whether an individual is in a more advanced stage of diabetes (through detection of diabetic retinopathy), while other non-invasive measurements such as pulse wave analysis will be more useful in classifying earlier stages of diabetes.

10.3 Larger Impact

The work described in this thesis has wide-ranging impact because it is the first of its kind to collect both retina and iris data. This is significant because the use of retina images to diagnose advanced stages of diabetes is well-accepted in the medical community whereas the use of iris images is still controversial. By taking both measurements, we hope to correlate changes in the iris and retina in order to determine whether the iris is a credible tool for diabetes screening.

In general, although the work in this thesis is specifically focused on diabetes, the non-invasive measurements collected allow for a more general assessment of a patient's cardiometabolic health. We would like to better understand the early links between related diseases, such as diabetes and cardiovascular disease, in order to design a model that can output risk scores for dysfunctions that fall under the general topic of cardiometabolic syndrome. Overall, we hope this work provides a catalyst to encourage additional collaboration between homeopathic and allopathic schools of medicine.

Bibliography

[1] Kwabena A. Ofori-Atta. Atime evolution of cardiometabolic syndrome. Technical report, Massachusetts Institute of Technology, 2019.

[2] Tania W. Yu. Iris imaging for health diagnostics. Master's thesis, MIT, 2018.

[3] Aneesh Anand. Bayesian models for screening and diagnosis of pulmonary disease. Master's thesis, MIT, 2018.

[4] Cleveland Clinic. Diabetes mellitus: An overview, 2015. Accessed: 2018-12-10.

[5] Mayo Clinic. Type 2 diabetes, 2019. Accessed: 2019-08-10.

[6] World Health Organization. Diabetes, 2018. Accessed: 2018-12-10.

[7] Christian Bommer, Vera Sagalova, Esther Heesemann, Jennifer Manne-Goehler, Rifat Atun, Till Bärnighausen, Justine Davies, and Sebastian Vollmer. Global economic burden of diabetes in adults: Projections from 2015 to 2030. *Diabetes Care*, 41(5):963–970, 2018.

[8] Sarah Wild, Gojka Roglic, Anders Green, Richard Sicree, and Hilary King. Global prevalence of diabetes. *Diabetes Care*, 27(5):1047–1053, 2004.

[9] International Diabetes Federation. India, 2019. Accessed: 2019-06-06.

[10] V Mohan. Why are indians more prone to diabetes? *J Assoc Physicians India*, 52:468–474, 2004.

[11] Sanghera. Diabetes is india's fastest growing disease: 72 million cases recorded in 2017, figure expected to nearly double by 2025, 2018. Accessed: 2018-11-10.

[12] Nikhil Tandon et al. The increasing burden of diabetes and variations among the states of india: the global burden of disease study 19902016. *The Lancet Global Health*, 6, December 2018.

[13] Rakesh Malik. India is the diabetes capital of the world!, 2016. Accessed: 2018-11-12.

[14] Phillip Tuso. Prediabetes and lifestyle modification: time to prevent a preventable disease. *The Permanente journal*, 18(3):88–93, 2014. 25102521[pmid].

[15] Richard E. Pratley. The early treatment of type 2 diabetes. *The American Journal of Medicine*, 126(9):S2–S9, Sep 2013.

[16] Gordon C. Weir and Susan Bonner-Weir. Five stages of evolving beta-cell dysfunction during progression to diabetes. *Diabetes*, 53(suppl 3):S16–S21, 2004.

[17] Diabetes.co.uk. Beta cells, 2019. Accessed: 2019-08-10.

[18] Diabetes.co.uk. hyperinsulinemia, 2019. Accessed: 2019-08-10.

[19] Mayo Clinic. Hyperglycemia in diabetes, 2019. Accessed: 2019-08-10.

[20] Medical News Today. What are the early signs of type 2 diabetes? Accessed: 2019-08-10.

[21] Joshua A. Beckman, Mark A. Creager, and Peter Libby. Diabetes and AtherosclerosisEpidemiology, Pathophysiology, and Management. *JAMA*, 287(19):2570–2581, 05 2002.

[22] Soumya D. and Srilatha B. Late stage complications of diabetes and insulin resistance. *Journal of Diabetes and Metabolism*, 2, 2011.

[23] vsp. Retinal imaging offers a better view and early detection, 2019. Accessed: 2018-12-11.

[24] Field Epidemiology Manual. Diagnostic tests versus screening tests. Accessed: 2019-08-26.

[25] Mayo Clinic. Diagnosis, 2019. Accessed: 2018-11-28.

[26] American Diabetes Association. Diagnosis, 2019. Accessed: 2018-08-10.

[27] Diabetes Action Research and Education Foundation. Questions and answers - symptoms of diabetes, 2019. Accessed: 2019-08-29.

[28] Diabetes.co.uk. Blood sugar level ranges, 2019. Accessed: 2019-08-10.

[29] Narendra Kotwal and Aditi Pandit. Variability of capillary blood glucose monitoring measured on home glucose monitoring devices. *Indian journal of endocrinology and metabolism*, 16(Suppl 2):S248–S251, Dec 2012. 23565391[pmid].

[30] Suresh Somannavar, Anbazhagan Ganesan, Mohan Deepa, Manjula Datta, and Viswanathan Mohan. Random capillary blood glucose cut points for diabetes and pre-diabetes derived from community-based opportunistic screening in india. *Diabetes Care*, 32(4):641–643, 2009.

[31] National Public Radio. Weavers turn silk into diabetes test strips, 2015. Accessed: 2018-12-10.

[32] Vijay Viswanathan and V. N. Rao. Problems associated with diabetes care in india. *Diabetes Manage*, 3(1):31–40, 2013.

[33] Utkarsh Ojha and Raihan Mohammed. Disruption in the diabetic device care market. *Medical devices (Auckland, N.Z.)*, 11:51–56, Feb 2018. 29440935[pmid].

[34] Heejung Bang, Alison M. Edwards, Andrew S. Bomback, Christie M. Ballantyne, David Brillon, Mark A. Callahan, Steven M. Teutsch, Alvin I. Mushlin, and Lisa M. Kern. Development and validation of a patient self-assessment score for diabetes risk. *Annals of internal medicine*, 151(11):775–783, Dec 2009. 19949143[pmid].

[35] Viswanathan Mohan, R Deepa, Mohan Deepa, S Somannavar, and Manjula Datta. A simplified indian diabetes risk score for screening for undiagnosed diabetic subjects. *The Journal of the Association of Physicians of India*, 53:759–63, 10 2005.

[36] Peek. What are peek solutions, 2019. Accessed: 2018-12-11.

[37] Eyenuk. Harnessing deep learning to prevent blindness, 2019. Accessed: 2018-12-11.

[38] Francis Ring. Thermal imaging today and its relevance to diabetes. *Journal of diabetes science and technology*, 4(4):857–862, Jul 2010. 20663449[pmid].

[39] Esther G. Gerrits, Helen L. Lutgers, Nanne Kleefstra, Reindert Graaff, Klaas H. Groenier, Andries J. Smit, Rijk O. Gans, and Henk J. Bilo. Skin autofluorescence. *Diabetes Care*, 31(3):517–521, 2008.

[40] Andries J. Smit, Jitske M. Smit, Gijs J. Botterblom, and Douwe J. Mulder. Skin autofluorescence based decision tree in detection of impaired glucose tolerance and diabetes. *PloS one*, 8(6):e65592–e65592, Jun 2013. 23750268[pmid].

[41] Peniel Argaw. Implementing a mobile solution to non-invasive optical screening of diabetes. Technical report, Massachusetts Institute of Technology, 2017.

[42] Yue Qiao, Zhaohua Gao, Yong Liu, Yan Cheng, Mengxiao Yu, Lingling Zhao, Yixiang Duan, and Yu Liu. Breath ketone testing: a new biomarker for diagnosis and therapeutic monitoring of diabetic ketosis. *BioMed research international*, 2014:869186–869186, 2014. 24900994[pmid].

[43] Zhennan Wang and Chuji Wang. Is breath acetone a biomarker of diabetes? a historical review on breath acetone measurements. *Journal of breath research*, 7:037109, 08 2013.

[44] B T E Andrews, W Denzer, G Hancock, A D Lunn, R Peverall, G A D Ritchie, and K Williams. Measurement of breath acetone in patients referred for an oral glucose tolerance test. *Journal of Breath Research*, 12(3):036015, may 2018.

[45] Bernard Jensen. *Seminumerical Algorithms*. Healthy Living Publications, 1980.

[46] E. Ernst. Iridology: Not Useful and Potentially Harmful. *JAMA Ophthalmology*, 118(1):120–121, 01 2000.

[47] R Sneha, Vinay Raj K, Basavaraj Hiremath, Ragavendrasamy Balakrishnan, and Mithun B S. Iris diagnosis-a quantitative non-invasive tool for diabetes detection. 07 2018.

[48] Lasya Thilagar. Application of mobile thermal imaging for health assessment. Technical report, Massachusetts Institute of Technology, 2016.

[49] Seek. Compact, 2019. Accessed: 2019-08-29.

[50] Thomas R. Dawber, JR H. Emerson Thomas, and Patricia M. McNamara. Characteristics of the dicrotic notch of the arterial pulse wave in coronary heart disease. *Angiology*, 24(4):244–255, 1973. PMID: 4699520.

[51] John M. Nkaze. Pymedserver: A server framework for mobile data collection and machine learning. Master's thesis, MIT, 2019.

[52] Django. Models, 2019. Accessed: 2019-08-10.

[53] Django. Class-based views, 2019. Accessed: 2019-08-10.

[54] Django REST framework. Class-based views, 2019. Accessed: 2019-08-10.

[55] Django REST framework. Serializers, 2019. Accessed: 2019-08-10.

[56] Django. Django shortcut functions, 2019. Accessed: 2019-08-10.

[57] Mostafa Langarizadeh and Fateme Moghbeli. Applying naive bayesian networks to disease prediction: a systematic review. *Acta informatica medica : AIM : journal of the Society for Medical Informatics of Bosnia & Herzegovina : casopis Drustva za medicinsku informatiku BiH*, 24(5):364–369, Oct 2016. 28077895[pmid].

[58] Agnieszka Onisko, Marek J. Druzdzel, Ph. D, Hanna Wasyluk, and Ph. D. A bayesian network model for diagnosis of liver disorders. Technical report, 1999.

[59] Bayesian networks, 2019. Accessed: 2019-08-10.

[60] Reshma. Patil and Jayashree. Gothankar. Assessment of risk of type 2 diabetes using the Indian Diabetes Risk Score in an urban slum of Pune, Maharashtra, India: a cross-sectional study. *WHO South-East Asia Journal of Public Health*, 5(1):53–61, 2016.

[61] Aditya Oruganti, Avinash Kavi, and Padmaja R. Walvekar. Risk of developing diabetes mellitus among urban poor south indian population using indian diabetes risk score. *Journal of family medicine and primary care*, 8(2):487–492, Feb 2019. 30984660[pmid].

[62] Hemavathi Dasappa, Farah Naaz Fathima, Rugmani Prabhakar, and Sanjay Sarin. Prevalence of diabetes and pre-diabetes and assessments of their risk factors in urban slums of bangalore. *Journal of family medicine and primary care*, 4(3):399–404, 2015. 26288781[pmid].

[63] Mridula J Solanki Prasad Tukaram Dhikale and Saurabh Ram Bihari Lal Shrivastava. A study of epidemiology of hypertension in an urban slum community of mumbai. *Biology and Medicine*, 3, 2015.