A PROJECT REPORT ON

INVESTMENT PLANNING AND TAX AUTOMATION USING REINFORCEMENT LEARNING

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

Of

SAVITRIBAI PHULE PUNE UNIVERSITY

By

JAYESH BAPU AHIRE	**71706642J**
ATHARVA ABHAY KARKHANIS	**71707122H**
SATYAM KUMAR	**71707600J**
ISHANA VIKRAM SHINDE	**71707640H**

Under the guidance of
Prof. S.P. Bholane

SINHGAD TECHNICAL EDUCATION SOCIETY-PUNE
तमसो मा ज्योतिर्गमय
Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING
SINHGAD COLLEGE OF ENGINEERING, PUNE-41
Accredited by NAAC

2019-20

Sinhgad Technical Education Society,
Sinhgad College of Engineering , Pune-41
Department of Computer Engineering

Sinhgad Institutes

Date: 25 April 2020

CERTIFICATE

This is to certify that the project report entitled

“Investment Planning and Tax Automation using Reinforcement Learning”

Submitted by

Jayesh Bapu Ahire	**Exam Seat No: 71706642J**
Atharva Abhay Karkhanis	**Exam Seat No: 71707122H**
Satyam Kumar	**Exam Seat No: 71707600J**
Ishana Vikram Shinde	**Exam Seat No: 71707640H**

is a bonafide work carried out by him/her under the supervision of Prof. S.P. Bholane and it is approved for the partial fulfillment of the requirements of Savitribai Phule Pune University, Pune for the award of the degree of Bachelor of Engineering (Computer Engineering) during the year 2019-20.

Prof. S.P. Bholane
Guide
Department of Computer Engineering

Prof. M.P. Wankhade
Head
Department of Computer Engineering

Dr. S.D. Lokhande
Principal
Sinhgad College of Engineering

Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project report. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank Prof. S.P. Bholane, for providing us with this wonderful opportunity and providing us with all the support and guidance, which made us complete the project report duly. We are extremely thankful to him for providing such a nice support and guidance.

We owe our deep gratitude to Dr. S.D. Lokhande, Principal, Sinhgad College of Engineering and Prof. M.P. Wankhade, HOD Department of Computer Engineering, who took keen interest on our project work and for providing all facilities and every help for smooth progress of project.

We are thankful and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of Department of Computer Engineering which helped us in successfully completing our project. At last we would like to thank all the unseen authors of various articles on the Internet, helping us become aware of the research currently ongoing in this field and all my colleagues for providing help and support in our work.

Jayesh Bapu Ahire 71706642J
Atharva Abhay Karkhanis 71707122H
Satyam Kumar 71707600J
Ishana Vikram Shinde 71707640H

Abstract

In an increasingly complex business world, automating the taxation process is necessary for increasing accuracy and driving efficiency. Tax automation improves transparency and compliance, mitigates undue burden on personnel, allows for greater collaboration and ultimately allows for a more sustainable tax platform. Portfolio management is a financial problem where an agent constantly redistributes some resource in a set of agents to maximise the returns. This project provides a single platform, which works as a customer assistant in providing Tax automation, Portfolio management and Investment planning completely backed by AI algorithms. Supervised Learning is used to achieve tax calculations for Tax automation. Portfolio management is achieved using Reinforcement Learning.

List of Figures

List of Tables

Acronyms

SRS	System Requirements Specification
UML	Unified Modelling Language
RL	Reinforcement Learning
AI	Artificial Intelligence
ML	Machine Learning
RRL	Recurrent Reinforcement Learning
EMH	Efficient Market Hypothesis
ANN	Artificial Neural Networks
EL	Ensemble Learning
OCR	Optical Character Recognition

Contents

Chapter 1

INTRODUCTION

1.1 Background and Basics

In today's age, AI research is constant and continues to grow. Over the last five years AI research has grown by 12.9 percent annually worldwide, according to technology writer Alice Bonasio.There is so much that artificial intelligence is being used for and so much more potential that it is hard to picture our future without help it especially when it comes to business.

Engineering methods and systems are routinely used in financial market applications, including signal processing, control theory and advanced statistical methods. The computerization of the markets encourages automation and algorithmic solutions, which are now well understood and addressed by the engineering communities. Moreover, the recent success of Machine Learning has attracted interest of the financial community, which permanently seeks for the successful techniques from other areas, such as computer vision and natural language processing to enhance modelling of financial markets.

Along with Tax Automation, Investment Planning and Portfolio Management are also some of the areas where AI can be useful. Better investment decisions come, in part, from more precise asset pricing. More-in-depth analysis provides more accurate inputs for valuation models. AI's freedom from emotions and behavioral biases leads to better investment decisions. Moreover AI also helps us in managing Portfolios in a better way to reduce risks and to improve profits.

1.2 Literature Survey

1.2.1 Deep Q-trading

Authors: Yang Wang, Dong Wang, Shiyue Zhang, Yang Feng, Shiyao Li and Qiang Zhou

Algorithmic trading is a hot topic in machine learning. Compared to other methods, reinforcement learning (RL), particularly Q-learning, can learn decision rules directly with reasonable reward, and therefore is suitable for learning trading strategies. Recently, Q-learning based on deep neural models, also known as deep Q-learning, has been successfully applied to some challenging tasks like game playing and robot motion. This paper proposes to employ deep Q-learning to build an end-to-end deep Q-trading system which can automatically determine what position to hold at each trading time. The experimental results show that the deep Q-trading system can outperform the buy-and-hold strategy as well as the strategy learned by recurrent reinforcement learning (RRL) that was known to be more effective than Q-learning.

1.2.2 Deep Direct Reinforcement Learning for Financial Signal Representation and Trading

Authors: Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai

One of the challenges in the recent times is to train the computer to beat experienced traders for financial assert trading. In this paper, they tried to address this challenge by introducing a recurrent deep neural network (NN) for real-time financial signal representation and trading. The model is inspired by two biological-related learning concepts of deep learning (DL) and reinforcement learning (RL). In the framework, the DL part automatically senses the dynamic market condition for informative feature learning. Then, the RL module interacts with deep representations and makes trading decisions to accumulate the ultimate rewards in an unknown environment. The learning system is implemented in a complex NN that exhibits both the deep and recurrent structures. Hence, we propose a task-aware backpropagation through time method to cope with the gradient vanishing issue in deep training. The robustness of the neural system is verified on both the stock and the commodity future markets under broad testing conditions.

1.2.3 Portfolio Management using Reinforcement Learning

Authors: Olivier Jin, Hamza El-Saawy

Accurate stock market predictions can lead to lucrative results, which is no wonder why investors are turning toward machine learning applications to analyze financial markets. However, one of the inherent difficulties with this approach is producing an accurate model of the current market and predicting future stock behaviors. In fact, one school of thought argues that, given the efficient market hypothesis (EMH), it is impossible for any agent to truly 'beat the market' by exceeding benchmark predictions. In this paper they attempted to evaluate this challenge by utilizing artificial neural networks (ANN), due to their ability to model nonlinear relationships between variables, as well as their lower need for formal statistical training. In addition, they use Q-learning since it is a model-free algorithm relying only on Q-factors without attempting to model the environment (which, in the case of the stock market, would be entirely unfeasible). Q-learning provides the added benefit of balancing between 'exploration' and 'exploitation' in order to provide the most optimal outcome.

1.2.4 Stock Trading with Recurrent Reinforcement Learning (RRL)

Authors: Gabriel Molina

One relatively new approach to financial trading is to use machine learning algorithms to predict the rise and fall of asset prices before they occur. An optimal trader would buy an asset before the price rises, and sell the asset before its value declines. In this paper, an asset trader has been implemented using recurrent reinforcement learning (RRL). It is a gradient ascent algorithm which attempts to maximize a utility function known as Sharpe's ratio. By choosing an optimal parameter w for the trader, they attempted to take advantage of asset price changes. Test examples of the asset trader's operation, both 'real-world' and contrived, are illustrated.

1.3 Project Undertaken

1.3.1 Problem Definition

To implement a Tax automation and Portfolio Management system to improve transparency and compliance, mitigate undue burdens on personnel, allow for greater collaboration and ultimately allow for a more sustainable tax platform; at the same time solving the problem of financial illiteracy with single platform which works as customer assistant in making complex financial services accessible to everyone.

1.3.2 Scope Statement

This project involves building a web-based single platform which will work as a customer assistant and will facilitate the ability to calculate taxable income, suggesting investment plan with asset allocation strategies. Data management and privacy protection are very critical because of the use of sensitive financial information of an individual; hence we might think of implementing a blockchain based solution for that in the near future.

1.4 Organization of Project Report

The overall report revolves around the objective of achieving Tax Automation, Investment Planning and Portfolio Management using Reinforcement Learning.

First chapter deals with introduction of Tax Automation, Investment Planning and Portfolio Management using Machine Learning and its need. In that we have included background and basics, literature survey and Project Undertaken.

Second chapter deals with project planning and managements. In that we include details of System Requirements Specification (SRS), Project Process Modeling, Cost and Effort Estimates and finally Project scheduling.

Third chapter deals with Analysis and Design. In this we include Idea Matrix and UML Diagrams.

Fourth chapter deals with the testing modules implemented on the project. In this we include Unit Testing, Integration Testing, Acceptance Testing and GUI Testing.

In this way we complete the report in the given sequence of chapters.

Chapter 2

PROJECT PLANNING AND MANAGEMENT

This chapter covers the project planning and management details. It also covers System Requirement specifications. SRS is considered as the base for the effort estimations and project scheduling.

2.1 Detail System Requirement Specification (SRS)

2.1.1 System Overview

INVARJAN integrates Tax Automation platform with Portfolio Management and Investment Planning platform creating a consistent singular platform, which caters to every customer need regarding Financial Services. The following are the main features that are included in INVARJAN:

1. Cross platform support: Offers operating support for most of the known and commercial operating systems.

2. User account: The system allows the user to create their accounts in the system and provide features of updating and viewing profiles.

3. Search: search is simply local search engine based on key words.

4. FAQs Bot: Frequently asked section contains answers of problems which users frequently face.

5. Tax Automation: Based on the Form 16 provided by the user appropriate calculation method is decided to calculate the taxable amount.

6. Asset Allocation: Based on the users needs and requirements, suitable assets are allocated.

7. Portfolio Management: Once the user has been allocated with different assets for investment, this module then helps in managing these assets to achieve maximum profits with minimum risks.

8. Payment Gateway: A Payment Gateway is provided for the user to initiate payments.

2.1.2 Functional Requirements

- Once the PDF file of the Form 16 is received, an optimised OCR (Optical Character Recognition) will be performed on the file to extract the data.
- Using NLP all the required data will be segregated.
- Based on the user's data an appropriate calculation method will be chosen.
- User will be asked to fill a questionnaire with all the information related to his goals, his taxable amount of income and amount allocated for investing.
- The data imported from Tax Automation module will then be verified with this data and stored in another database which will be used for analysis.
- Record of the users invested assets.
- General awareness of the real time fluctuations in assets which come under stochastic environments.

2.1.3 Non-Functional Requirements

2.1.3.1 Performance Requirements

When we scale our primary requirement will be near real-time performance of application. We will be fetching real-time data and inferring from it which is critical and data changes every minute and it should impact our suggestions and recommendations upto certain extent.

2.1.3.2 Security Requirements

Data management and privacy protection are very critical because of the use of sensitive financial information of an individual; we might think of implementing a blockchain based solution for that. For now, we will focus on data encryption and storage as an alternative.

2.1.3.3 Software Quality Attributes

- **User active involvement:** the active participation of users is one of the most important features. We're taking user inputs for whether the suggestions were helpful for him/her or not and that will help us to incorporate real-time RL in our platform.
- **Sharing information:** Data from these two separate services will be stored in one single database and shared among them to reduce user efforts of submit data individually but for this user consent will be taken via licence agreement.
- **Endless beta condition:** considering the above two features it is easy to understand that this product will be undergoing training in real time all the time means it's learning even when it's working so there is no 'final version' of it.

2.1.4 Deployment Environment

2.1.4.1 Hardware Requirements

- Working PC with minimum 2GB RAM and processor version higher than Pentium.
- Working internet connection.

2.1.4.2 Software Requirements

- Custom Finance API's created and open-sourced.
- Tax calculation API's written in python to ease tax calculation process.
- Visualization libraries including plotly and some of the in-house options which are hybrid of open-source libraries available.
- Stanford CoreNLP to facilitate all NLP tasks including inferring from data discovered from Form-16.
- stripe as a payment gateway.
- Tensorflow for building TFX pipelines to be used in multiple scenarios.
- Google cloud document discovery API's along with custom AutoML based OCR system.

- Hosted and served through Google Cloud Platform using cloud ML engine and hosting services.

2.1.5 External Interface Requirements

2.1.5.1 User Interfaces

- Login and permission manager to allow only authorized users to access platform.
- Tax Automation dashboard to accept Form-16 as input.
- Portfolio management dashboard with choice of assets and time period for which user wants to try investment. Dashboard will contain virtual investment option where user can try investing in particular plan virtually and see if that works for him/her.
- payment gateway to facilitate particular payments from both portfolio management and tax automation dashboard.

2.1.5.2 Communication Interfaces

• Web Browser with good Internet connection eg:- Microsoft Edge, Firefox Mozilla, Google Chrome, Internet Explorer 9+
• Stable network with a minimum internet speed of 400-500Kbps for uploading the Form16 document to the cloud bucket and receiving the extracted csv file.

2.1.6 Other Requirements (Software Quality Attributes)

2.1.6.1 Completeness

All external libraries including their respective license will be documented.

2.1.6.2 Usability

• Web application users have grown to expect easy Web transactions.
• Given HELP portal for new users for better understanding of features and its capabilities.

2.1.6.3 Maintainability

Application code will be cohesive and have easily recognizable functionality. Classes will be abstract enough to facilitate changes in data structures. Class and function modularity will be implemented to avoid the need for major refactoring.

2.2 Cost & Efforts Estimates

Cost of the project will be the cost of hardware (mainly CPU) plus the cost of work that is put in. Basic COCOMO:

- **Project class:** We have determined our project ts the characteristics of Semi-detached Mode as project is college level and requirements are rigid and less than rigid.
- **Number of code Lines:** We estimate our project will have 10000 Delivered Source instructions.

So, the Basic COCOMO model equations are as follows:
Effort Applied (E) = $a_b\,(KLOC)\,b_b$[man months]
Development Time (D) =$c_b\,(E)\,d_b$[months]
People Required (P) = E/D [count]
Where:
KLOC (Kilo lines of code) is the estimated number of delivered lines (in thousands) of code for project.
E is the effort applied per person per month.
D is the development time in consecutive months.
The coefficients a_b, b_b, c_b, d_b are predetermined according to project class given in the following table:
Calculations: So, this project comes under the semi-detached mode.

2.3 Project Scheduling

2.3.1 Time Line Chart

Table 2.1: Cost Estimate Table

Software Project Class	$\mathbf{a}_b$	$\mathbf{b}_b$	$\mathbf{c}_b$	$\mathbf{d}_b$
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embeded	3.6	1.20	2.5	0.32

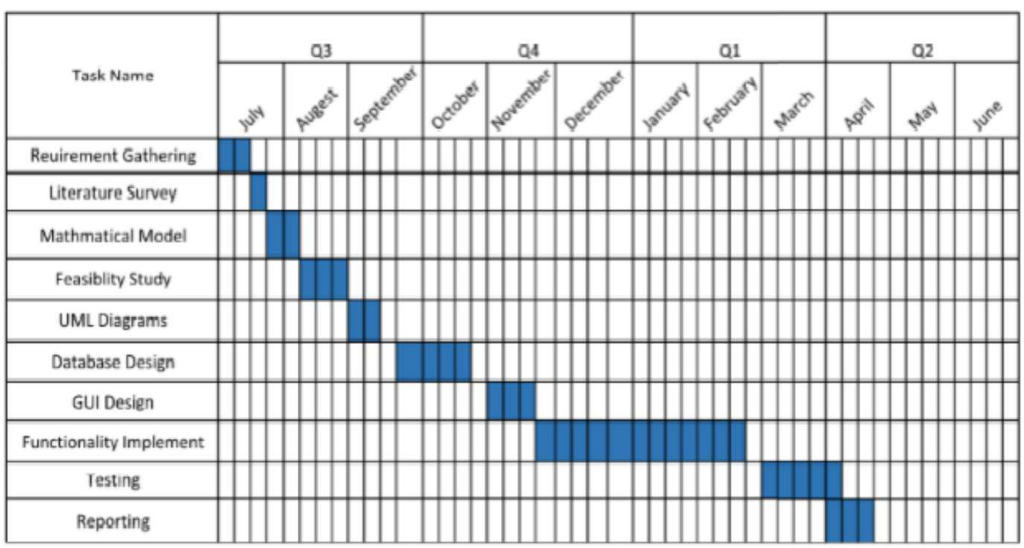

Figure 2.1: Time Line Chart

Chapter 3

ANALYSIS AND DESIGN

3.1 IDEA Matrix

An IDEA matrix is a concept that evaluate various effects that the idea has. This tells us almost everything about the project.

Table 3.1: IDEA Matrix

IDEA	Deliverable	Parameters Affected
Innovate	1. Single platform for Personalised Portfolio Management and Tax Automation 2.Process of extracting details of Form 16 of user	1. Single multi-functional platform 2. Data extraction method
Increase	1. Investment options based on short and long term goals	1. Investment Options
Improve	1. User experience for Tax filing process and investment assistance	1. User Interface
Deliver	1. Personalised investment portfolio based on goals. 2. Suggestions for Direct Tax reductions	1. Investment Service
Decrease	1. Dependencies on Chartered Accountants (CA) 2. Time in research for investment options	1. Cost 2. Time
Eliminate	1. Mutual-fund focused investment platform	1. Risk based investments
Educate	1. Educate project members and users	1. Project Members 2. Users
Advance	1. Advancement in investment planning based on risk 2. Optimised investment solutions	1. Optimisation
Advantage	1. No dependency on CA 2. Reduced costs 3. Cloud based deployment	1. Performance 2. Cost
Avoid	1. Avoiding use of local servers for deployment	1. Deployment

The letters in the word idea are the initials of the components analyzed regarding the

project.
It is represented in a tabular form. Along with the component name, its description is written beside it in another column.

3.2 Use-Case Diagrams

3.2.1 Use-Case Diagram 1

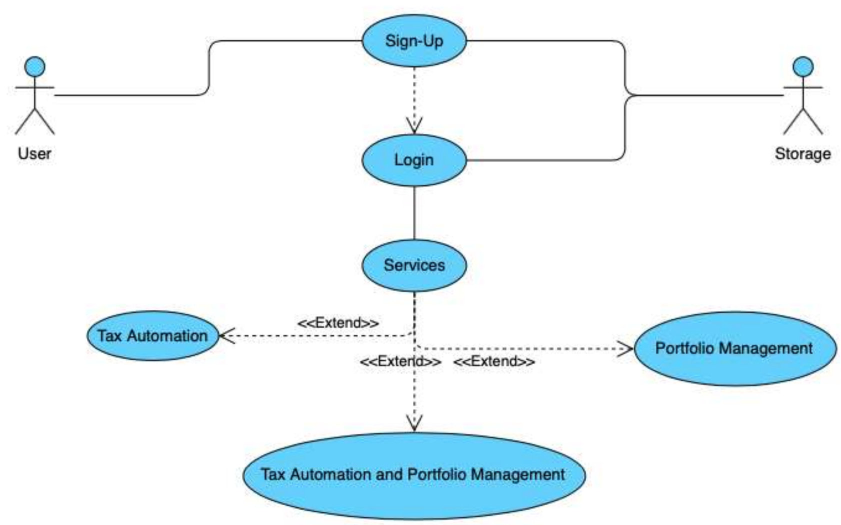

Figure 3.1: Use-Case Diagram 1

This use case diagram is used to determine the flow of the system. The user and the storage are the actors in this scenario. The user is first directed to sign-in page where if he is a new user, he is redirected to the sign-up process. Once the user is successfully is logged-in, the services option will be provided to the user. The services activities extended three other activities from which the user can select any one service. The three services provided are Tax Automation, Portfolio Management and lastly a combination of both Tax Automation and Portfolio Management.

3.2.2 Use-Case Diagram 2

This diagram displays the use case module for the Tax Automation service. The actor in this case will be the Form16 which will be provided by the user. Once the form is

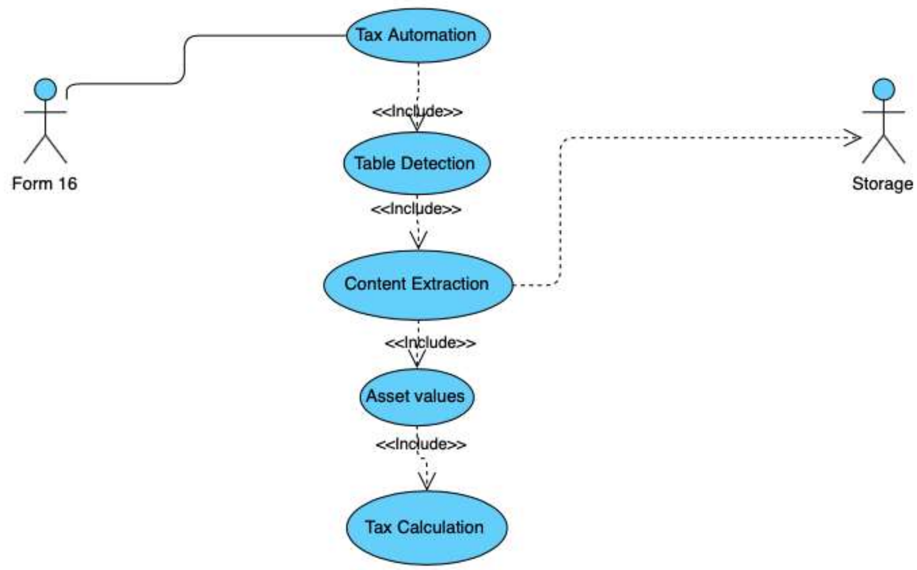

Figure 3.2: Use-Case Diagram 2

uploaded, the actions like table detection, content extraction are performed. The asset values will be determined based on the form provided. Based on this the tax will be calculated for each user and then he or she will be directed for payment.

3.2.3 Use-Case Diagram 3

This diagram displays the use case module for the Portfolio Management service. The actor in this case will be the User. The user will be required to fill up Questionnaire which will be stored for further requirement. Based on the users information, a recommendation will be provided to the user. When the user invests in the recommendation, he or she will be provided with two options - Asset plans and Investment plans. Lastly the user will be directed to payment option.

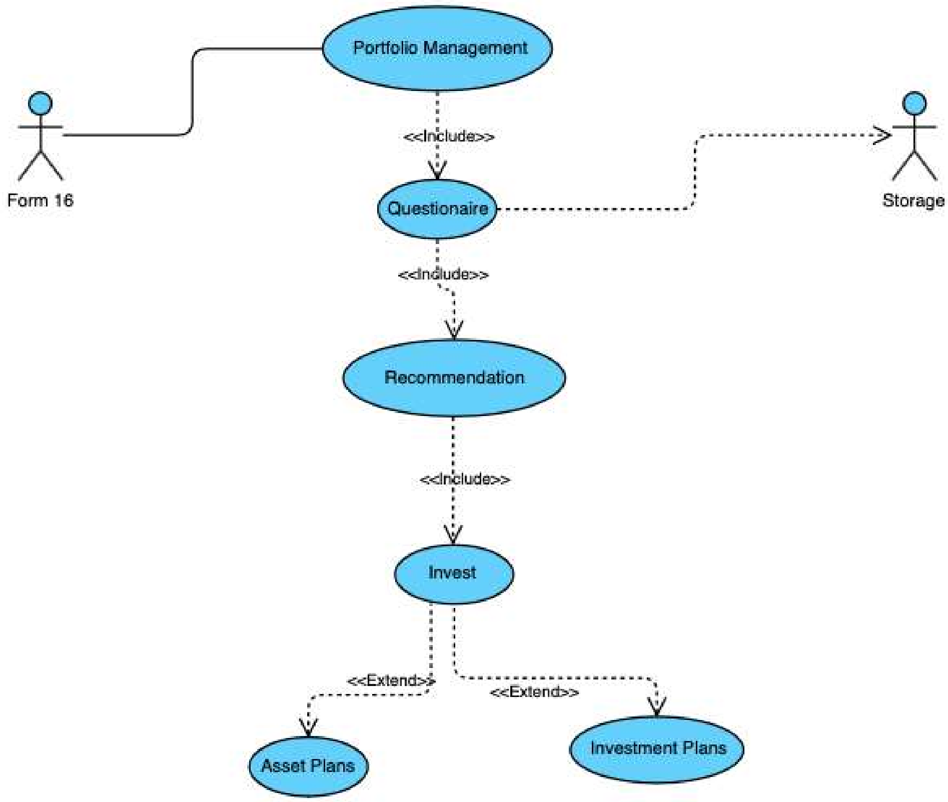

Figure 3.3: Use-Case Diagram 3

3.2.4 Use-Case Diagram 4

This diagram displays the use case module for Tax Automation and Portfolio Management service. This module include all the functionalities of the tax automation module combined with the functionalities of the portfolio management module.

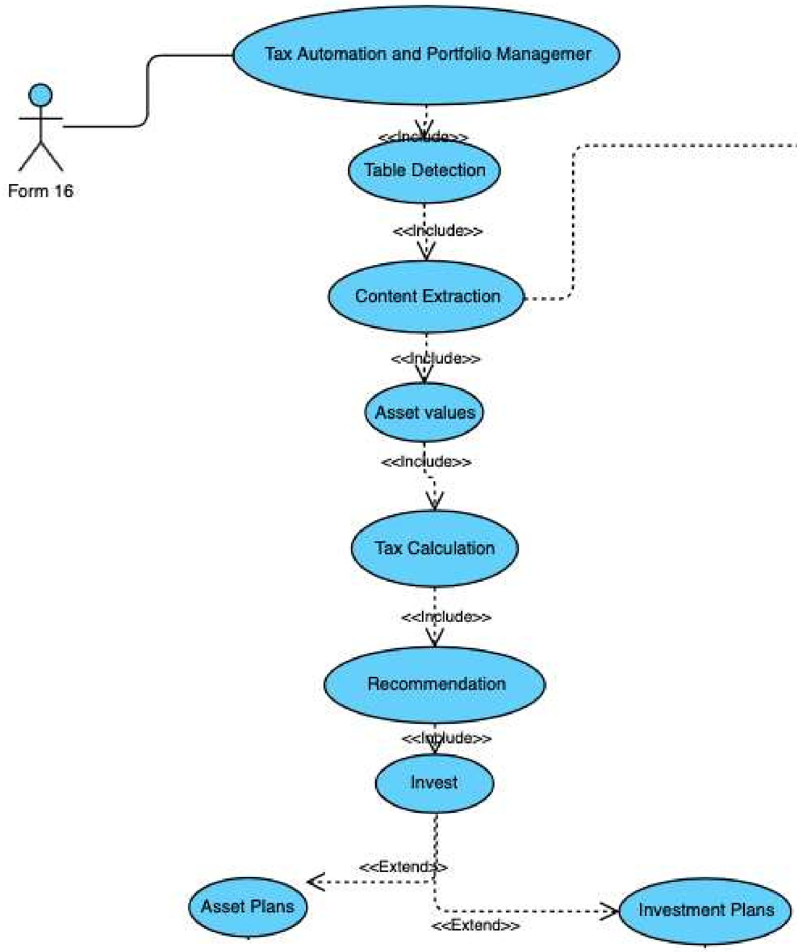

Figure 3.4: Use-Case Diagram 4

3.3 Activity Diagram

This diagram represents various activities in the system along with the different actions that take place within each activity. It also depicts the flow of the actions within the activities and relation between different activities.

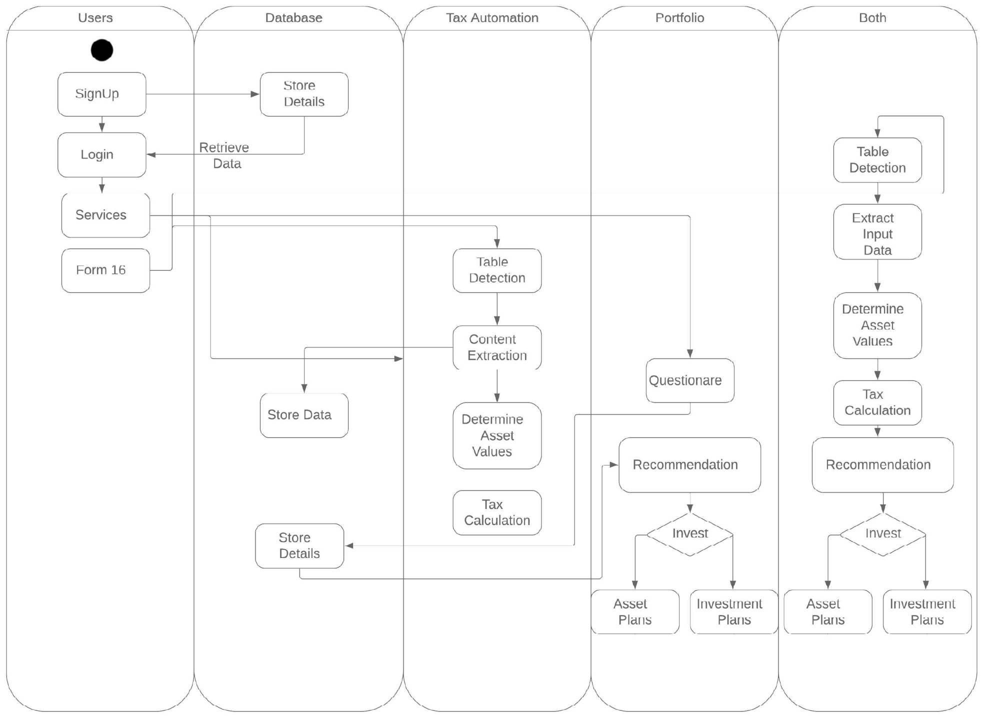

Figure 3.5: Activity Diagram

3.4 Class Diagrams

The User class has one to one association with the Accounts class meaning that the user can create exactly one account and consists of different attributes like name, age, income, etc. The login class is an aggregation of the accounts class wherein the user will be granted access if his account exists. Once successfully logged in the user will be associated with the service class where the user can choose one service that he/she desires. The attributes of the service class are the different services offered by the system. This consists of Tax Automation Service, which scans the form 16 from the user and gives the tax calculated. The second service consists of Portfolio management where in the user is directed to fill out a questionnaire which gives user recommendation on the basis of their answers. They are further provided with investment option which the can choose accordingly. The system also provides a service which includes both tax automation and portfolio management. Lastly the user is associated with the payment class where he or she can pay for the services opted using suitable payment methods.

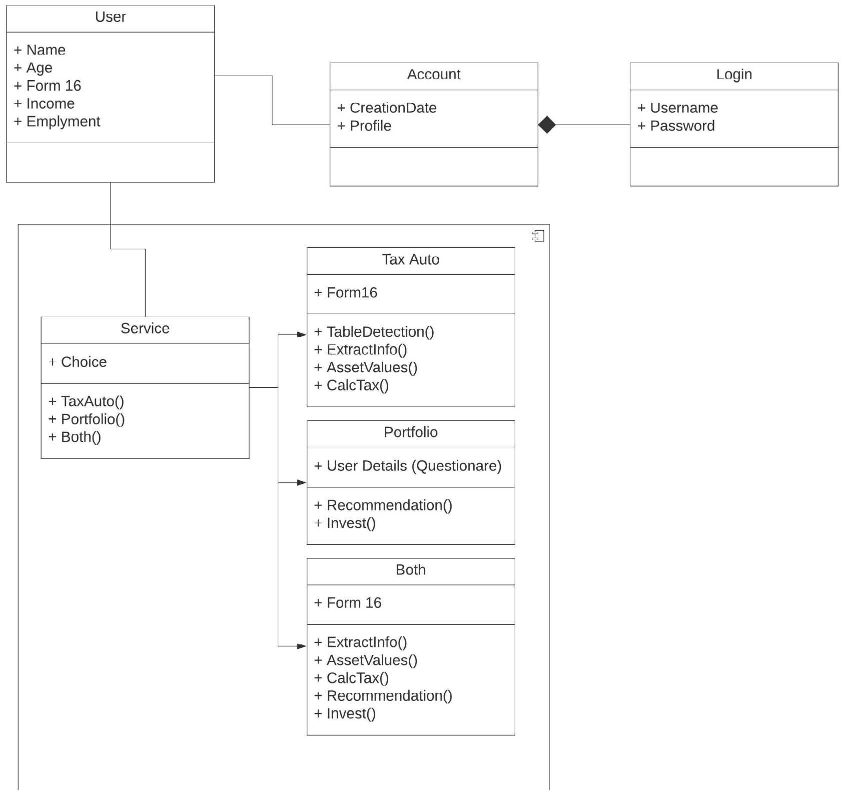

Figure 3.6: Class Diagram

3.5 State Transition Diagrams

This diagram depicts the different states present in the system and the transition flow from one state to the other.

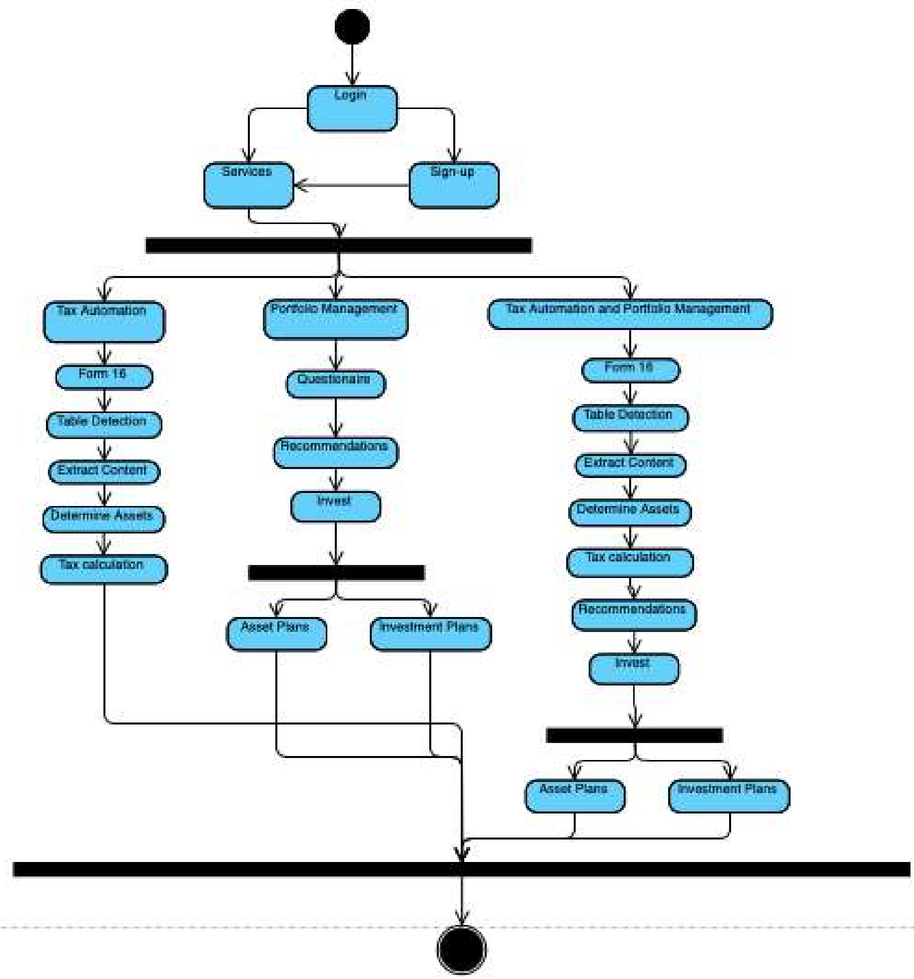

Figure 3.7: State Transition Diagram

3.6 Interface Diagram

The interface diagram includes interfaces like the user interface which consists of service interface. The services interface include various interfaces like tax automation, portfolio management and combination of tax automation and portfolio management. The service interface accesses the data of the user from the file database.

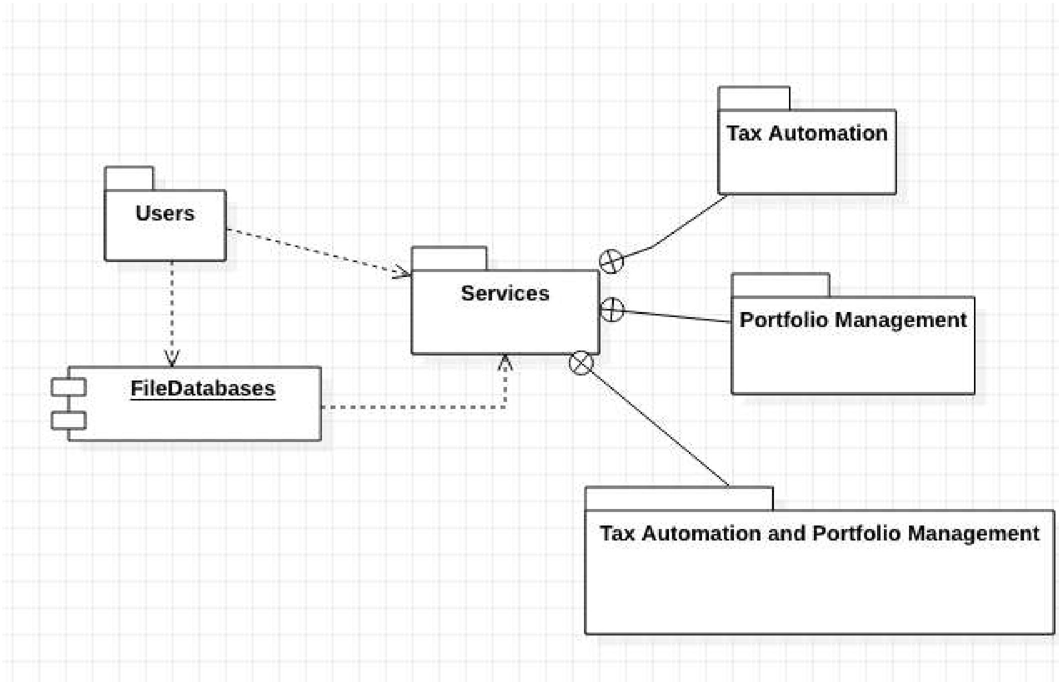

Figure 3.8: Interface Diagram

3.7 Package Diagram

The package diagram includes packages like UI framework which includes a service package. This service package accesses the tax automation package, portfolio management package and the tax automation and portfolio management package. The UI framework imports these three packages when the user asks for various services.

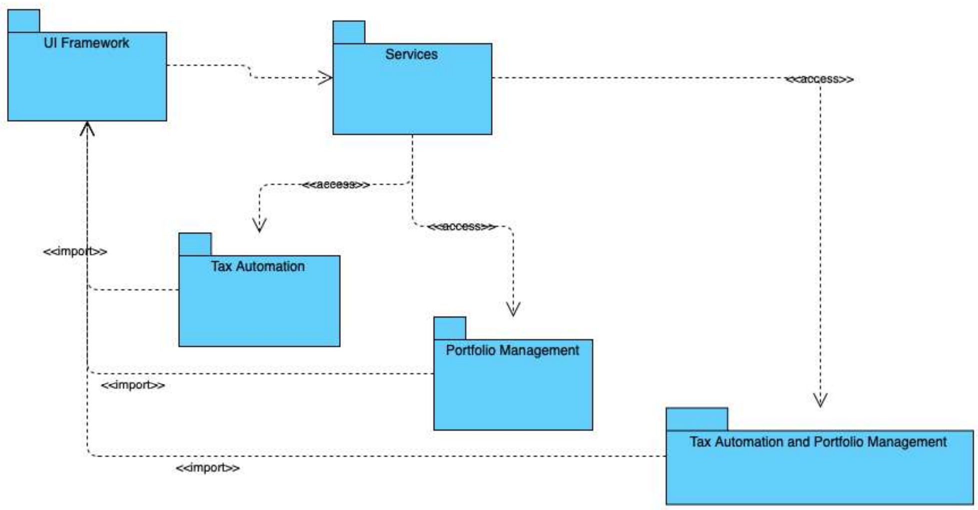

Figure 3.9: Package Diagram

Chapter 4

IMPLEMENTATION CODING

4.1 Introduction

This chapter covers the role of various subsystems/modules/classes along with implementation details listing of the code for the major functionalities.

4.2

4.2.1 Operational Details

The project consists of two major modules. These two modules then in turn contain further functionalities and classes which will be discussed further.

The following are the two main modules:

1. Tax Automation module
2. Portfolio Management module

Following are the major functionalities contained in the main modules:

1. Tax Automation

   (i) Upload Form16 to the Cloud Bucket.
   (ii) Extract data from the Form16 using Cloud and Table Discovery.
   (iii) Get the Financial breakdown from the extracted data and display the output.

2. Portfolio Management

   (i) Pairing assets to create a portfolio.

(ii) Analyzing the performance of the two assets over time.

(iii) Maintaining the returns from the assets.

4.2.2 Major Classes

* Following are the major classes of the Tax Automation module's functionalities:

1. Uploading Form16:

(i) Upload class: This class helps us to upload the Form16 file to the bucket on the cloud service being used for further processing.

2. Data Extraction:

(i) Textractor class: This class helps us to extract data from the uploaded Form16 file. It has functions like 'validateinput' and 'processdocuments' that recognize the data from the tables and then use it for further processing on the cloud.

3. Output generation from Form16:

(i) OutputGenerator class: This class helps us to generate the output to be displayed from the uploaded Form16. I has functions like 'outputwords', 'outputtext' and 'outputtables', which help in extracting words, sentences and tables from the uploaded document.

* Following are the major classes of the Portfolio Management module's functionalities:

1. Generating asset pairs:

(i) 'getpairdata': This helps us to generate pairs of the assets chosen by the user to build up a portfolio of the assets. This portfolio is then maintained over the period of investment.

2. Analyzing performance:

(i) 'buildportfolioperfchart' and 'buildsummarytable': These help us to accurately analyze and keep a track of the values of the asset pairs over a period of time. Analyzing is important as only after the analyzing phase can we maintain the assets to make the portfolio more profitable.

3. Maintaining the portfolio:

(i) 'assetreturnlist' and 'getportfolioreturns': These help us to maintain the portfolio by shifting the asset values over the investment period according to the chances anticipated in the market, so as to improve the return profit margin.

4.2.3 Code Listing

Code 1

```
import boto3
def upload(source_file, bucket_name, object_key):
    s3 = boto3.resource('s3')
    try:
        s3.Bucket(bucket_name).upload_file(source_file, object_key)
    except Exception as e:
        print(e)
```

The code snippet above is to upload the Form16 file from the user's local device to a bucket on the cloud for further processing.

Code 2

```
def validateInput(self, args):
    event = self.getInputParameters(args)
    ips =
    if(not 'documents' in event):
        raise Exception("Document or path to a folder or S3
       bucket containing documents is required.")
    inputDocument = event['documents']
    idl = inputDocument.lower()
    bucketName = None
    documents = []
    awsRegion = 'us-east-1'
    if(idl.startswith("s3://")):
        o = urlparse(inputDocument)
        bucketName = o.netloc
        path = o.path[1:]
        ar = S3Helper.getS3BucketRegion(bucketName)
        if(ar):
            awsRegion = ar
        if(idl.endswith("/")):
            allowedFileTypes = ["jpg", "jpeg", "png", "pdf"]
            documents = S3Helper.getFileNames(awsRegion,
            bucketName, path, 1, allowedFileTypes)
        else:
            documents.append(path)
    else:
        if(idl.endswith("/")):
```

```
            allowedFileTypes = ["jpg", "jpeg", "png"]
            documents = FileHelper.getFileNames(inputDocument,
            allowedFileTypes)
        else:
            documents.append(inputDocument)
        if('region' in event):
            awsRegion = event['region']
    ips["bucketName"] = bucketName
    ips["documents"] = documents
    ips["awsRegion"] = awsRegion
    ips["text"] = ('text' in event)
    ips["forms"] = ('forms' in event)
    ips["tables"] = ('tables' in event)
    ips["insights"] = ('insights' in event)
    ips["medical-insights"] = ('medical-insights' in event)
    if("translate" in event):
        ips["translate"] = event["translate"]
    else:
        ips["translate"] = ""
    return ips
```

The code above is for validating the input given to the text extractor after the Form16 document is uploaded tot the cloud.

Code 3

```
def processDocument(self, ips, i, document):
    print(”Document : ”.format(i,
      document))
    print(’=’ * (len(document)+30))
     Get document textracted
    dp = DocumentProcessor(ips[”bucketName”], document,
      ips[”awsRegion”], ips[”text”], ips[”forms”], ips[”tables”])
    response = dp.run()
    print(”Received Textract response...”)
    FileHelper.writeToFile(”temp-response.json”,
      json.dumps(response))
    Generate output files
    print(”Generating output...”)
    name, ext = FileHelper.getFileNameAndExtension(document)
    opg = OutputGenerator(response,
       ”-”.format(name, ext),
       ips[”forms”], ips[”tables”])
    opg.run()
   if(ips[”insights”] or ips[”medical-insights”] or ips[”translate”]):
   opg.generateInsights(ips[”insights”], ips[”medical-insights”],
      ips[”translate”], ips[”awsRegion”])
    print(” textracted successfully.”.format(document))
```

The code above is a part of the Texttractor class of the Tax Automation module. It helps us to actually recognize and recover the words, lines and tables from the uploaded document. We use Amazon Web Services cloud for the extraction process.

Code 4

```
def outputForm(self, page, p):
    csvData = []
    for field in page.form.fields:
        csvItem = []
        if(field.key):
            csvItem.append(field.key.text)
            csvItem.append(field.key.confidence)
        else:
            csvItem.append("")
            csvItem.append("")
        if(field.value):
            csvItem.append(field.value.text)
            csvItem.append(field.value.confidence)
        else:
            csvItem.append("")
            csvItem.append("")
        csvData.append(csvItem)
    csvFieldNames = ['Key', 'KeyConfidence', 'Value',
   'ValueConfidence']
    FileHelper.writeCSV("-page--forms.csv".format(self.fileName,
    p), csvFieldNames, csvData)
```

The code above helps us to convert the extracted data into csv format so that it can be used for further processing to be displayed on the site.

Code 5

```
def outputTable(self, page, p):
    csvData = []
    for table in page.tables:
        csvRow = []
        csvRow.append(”Table”)
        csvData.append(csvRow)
        for row in table.rows:
            csvRow = []
            for cell in row.cells:
                csvRow.append(cell.text)
            csvData.append(csvRow)
        csvData.append([])
        csvData.append([])
    FileHelper.writeCSVRaw(”-page--tables.csv”
   .format(self.fileName, p), csvData)
```

The code above is for the extracted data, which is stored as key-value pair in the dictionary, to be appended to the csv file, so that it can be used for the further calculations for the Financial breakdown, which is then displayed on the site.

Code 6

```
getpairdata ¡- function(asset1 = ”eth”,
asset2 = ”AMZN”,
portstartdate = Sys.Date()-183,
portenddate = Sys.Date()-3,
initialinvestment=1000)
if(asset1 %in% crypto_list == T)
cryptourl1 ¡- paste0(”https://coinmetrics.io/data/”,asset1,”.csv”)
asset1data ¡- fread(cryptourl1)
asset1data ¡- asset1data[,c(1,5)]
names(asset1data) ¡- c(”date”,asset1)
else
asset1data ¡- as.data.frame(getSymbols(asset1, src = ”yahoo”,
auto.assign=FALSE))
setDT(asset1data, keep.rownames = TRUE)[]
asset1data ¡- asset1data[,c(1,7)]
names(asset1data) ¡- c(”date”,asset1)
asset1data ¡- asset1data % ¿ % fill(paste(asset1))
```

The code snippet above, references to the part where we pair off the assets which have been selected by the user as per their interest.

The assets selected are paired and then the pair defines the portfolio for the user.

This portfolio is then maintained over the period of time.

Code 7

```
build_summary_table <- function(portfolio_data)
asset_names <- c(names(portfolio_data[2]), names(portfolio_data[3]))
asset_portfolio_max_worth <- c(max(portfolio_data[4]),
max(portfolio_data[5]))
asset_portfolio_latest_worth <-
c(as.numeric(tail(portfolio_data[4], 1)),
as.numeric(tail(portfolio_data[5], 1)))
asset_portfolio_absolute_profit <-
c(as.numeric(tail(portfolio_data[4], 1))
-as.numeric(head(portfolio_data[4], 1)),
as.numeric(tail(portfolio_data[5], 1)) - as.
numeric(head(portfolio_data[5], 1)))
asset_portfolio_rate_of_return <-
c(((as.numeric(tail(portfolio_data[4], 1))
-as.numeric(head(portfolio_data[4], 1)
))/as.numeric(head(portfolio_data[4], 1))) * 100,
((as.numeric(tail(portfolio_data[5], 1)) - as
.numeric(head(portfolio_data[5], 1)))/as.nu
meric(head(portfolio_data[5], 1))) * 100)
asset_summary_table <-
data.frame(asset_names, asset_portfolio_max_worth, asset_portfolio_latest_worth,
asset_portfolio_absolute_profit, asset_portfolio_rate_of_return)
colnames(asset_summary_table) <- c("AssetNames",
"Asset Portfolio Max Worth",
"Asset Portfolio Latest Worth",
"Asset Portfolio Absolute Profit",
"Asset Portfolio Rate of Return")
return(asset_summary_table)
```

This code analyzes the portfolio over time and predicts the future rise and drop of the assets in the market.

Code 8

get$_{p}ortfolio_{r}eturns < -function(portfolio_{d}ata, period = ”weekly”)$

asset$_{1n}ame_{s}tr < -names(portfolio_{d}ata[2])$

asset$_{2n}ame_{s}tr < -names(portfolio_{d}ata[3])$

asset$_{1x}ts < -xts(x = portfolio_{d}ata[,2],$

order.by=as.Date(portfolio$_{d}ata$date))

asset$_{2x}ts < -xts(x = portfolio_{d}ata[,3],$

order.by=as.Date(portfolio$_{d}ata$date))

asset$_{1r}eturns < -periodReturn(asset_{1x}ts, period = period)$

asset$_{2r}eturns < -periodReturn(asset_{2x}ts, period = period)$

names(asset$_{1r}eturns) < -paste0(asset_{1n}ame_{s}tr, ”_{r}eturns”)$

names(asset$_{2r}eturns) < -paste0(asset_{2n}ame_{s}tr, ”_{r}eturns”)$

asset$_{r}eturns_{l}ist < -list(asset_{1r}eturns, asset_{2r}eturns)$

return(asset$_{r}eturns_{l}ist)$

The code snippet above, helps us to maintain the assets by predicting the future values and then adjusting the portfolio accordingly.

4.3 Screen shots

Figure 4.1: Form16 Upload Page

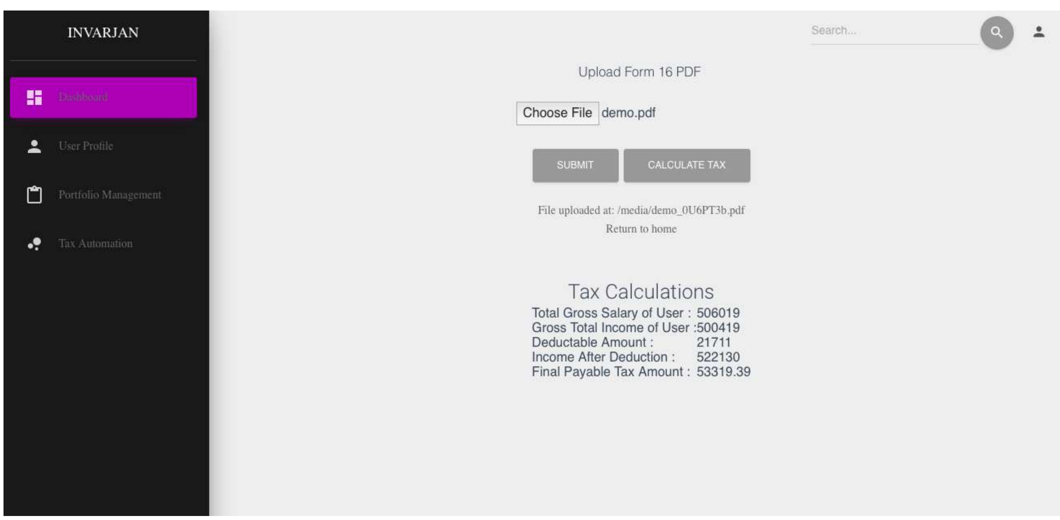

Figure 4.2: Tax Calculation Result Page

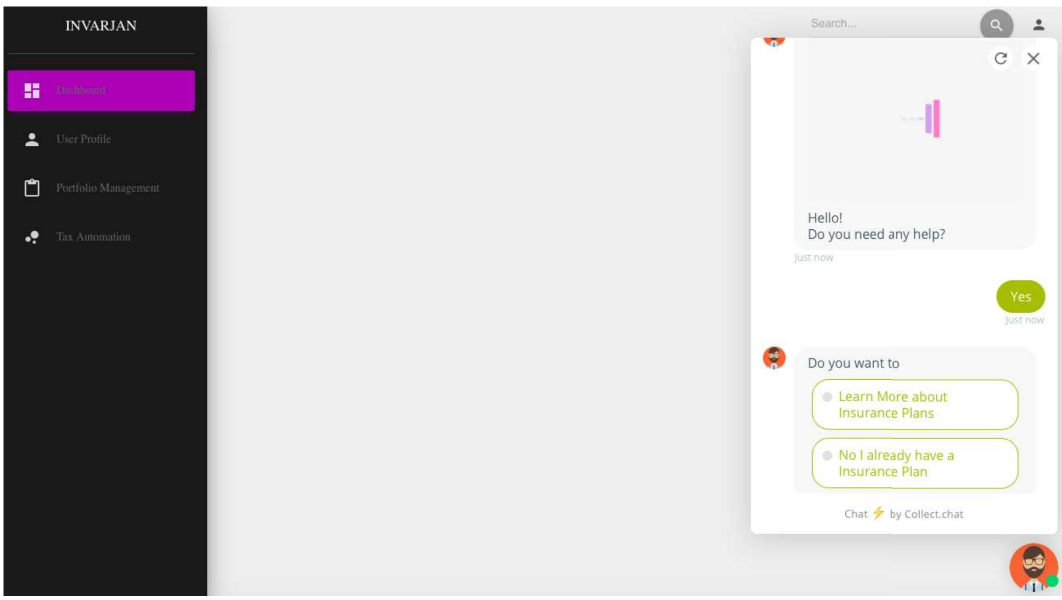

Figure 4.3: Chat-Bot Interface

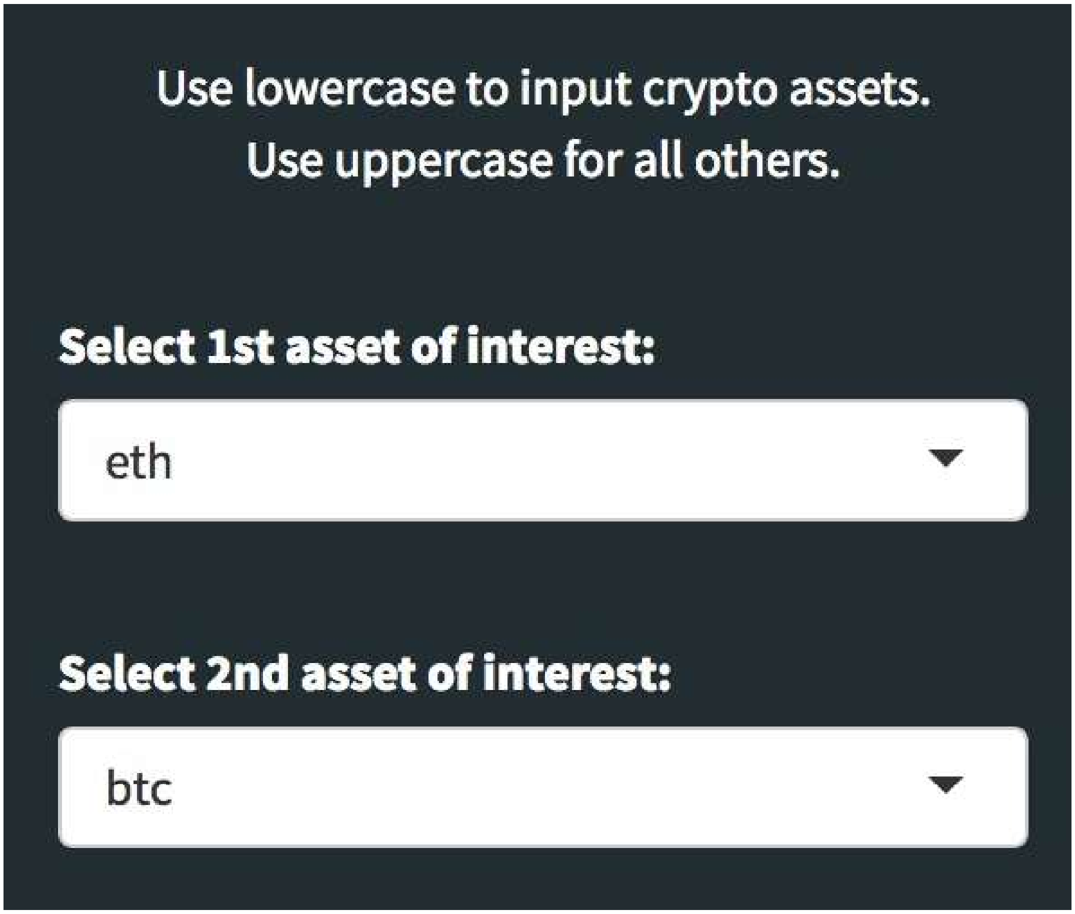

Figure 4.4: Asset Input Bar

Figure 4.5: Date Range Bar

Figure 4.6: Title links Bar

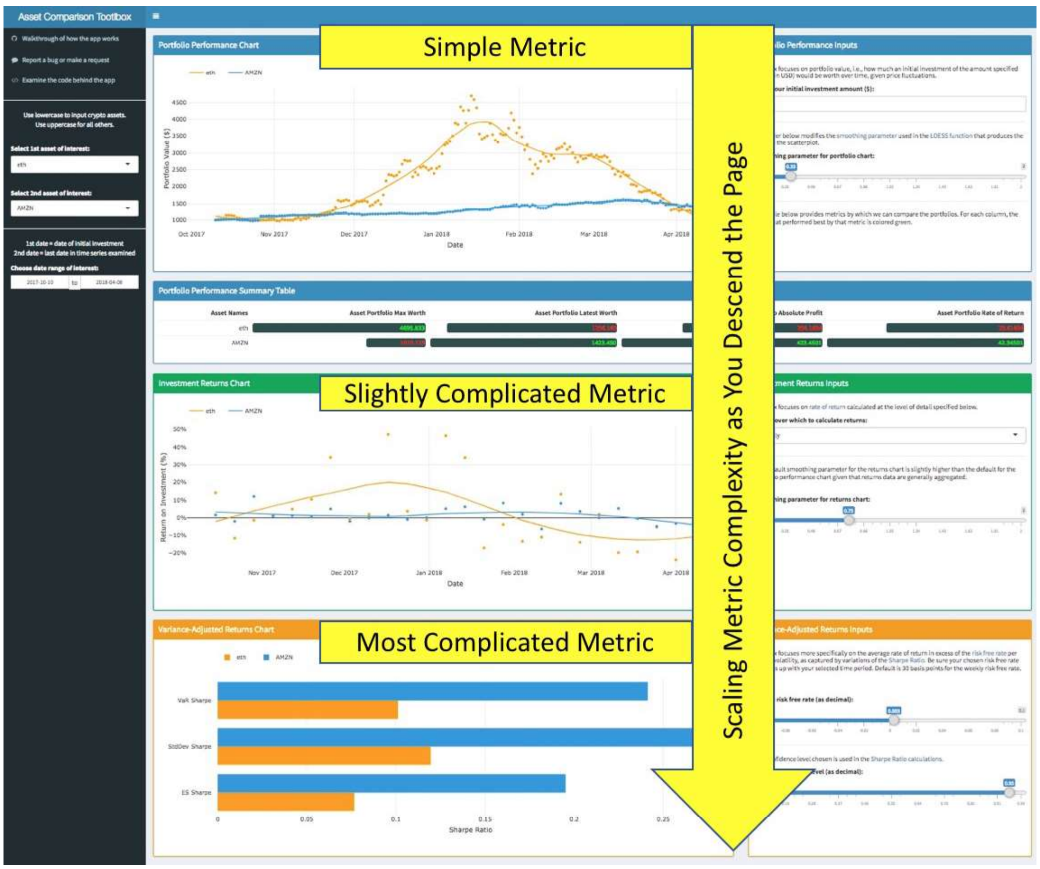

Figure 4.7: User Interface Scaling Complexity

Chapter 5

TESTING

5.1 Unit Testing

Table 5.1: Unit testing

TEST ID	TEST CASE	TEST SCENARIO	TEST INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
1.	Successful login	Check if the user is valid by interacting with the database.	User ID, Password.	Successfully logged in, direct to Home page.	Successfully logged in, direct to Home page.	Passed
2.	Unsuccessful login	Check if the user is valid by interacting with the database.	User ID, Password.	Unsuccessful login, user not found, redirect to registration page.	Unsuccessful login, user not found, redirect to registration page.	Passed

TEST ID	TEST CASE	TEST SCENARIO	TEST INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
3.	Successful registration	Signup a new user by accepting his details provided all entries are valid and in required format.	User ID, Password, Name.	Successfully registered, redirect to login page.	Successfully registered, redirect to login page.	Passed
4.	Unsuccessful registration	All necessary fields are not entered/ entries not in valid format.	User ID, Password, Name.	Registration unsuccessful, register again.	Registration unsuccessful, register again.	Passed
5.	Home page recommendation	Once the user is logged in, he should receive recommendations on his home page for choosing between the three services.	User ID, Module selection.	User successfully chooses on of , the 3 services.	User successfully chooses on of , the 3 services.	Passed

5.2 Integration Testing

Table 5.2: Integration testing

TEST ID	TEST CASE	TEST SCENARIO	TEST INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
1.	Main-page	Check if the user is valid by interacting with the database.	User ID, Password.	Successfully logged in/signup and direct to Home page. If not then remain to same page and tell user to try again.	Successfully logged in/signup and direct to Home page. If not then remain to same page and tell user to try again.	Passed
2.	Homepage	On clicking login button user should login successfully and receive service options.	User ID, Module selection.	Display correct service options by interacting with home module.	Display correct service options by interacting with home module.	Passed
3.	Accepting Form 16	Once Tax Automation is selected by user accept Form 16.	User ID, Form 16(PDF).	Successful scanning of Form 16 and data extraction.	Successful scanning of Form 16 and data extraction.	Passed
4.	Investment Questionnaire	User fills a Questionnaire regarding his goals and Investment choices.	User ID, Questionnaire.	Accept ratings and update to database.	Accept ratings and update to database.	Passed
5.	Logout	Once the user clicked on logout button it should direct to main page.	User ID.	Redirect to main-page by logging out.	Redirect to main-page by logging out.	Passed

5.3 Acceptance Testing

Table 5.3: Acceptance testing

TEST ID	TEST CASE	TEST SCENARIO	TEST INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
1.	Sign Up	Sign Up a new user by accepting user details.	User details.	New User has successfully created a new account	New User has successfully created a new account	Passed
2.	Login	Old user will login successfully by entering correct credentials	User ID, Password.	Old User has successfully logged in using his/her username and password	Old User has successfully logged in using his/her username and password	Passed
3.	Service selection	User selects one of the three services	User ID, Module selection.	User is redirected to appropriate module.	User is redirected to appropriate module.	Passed
4.	Form 16 Acceptance	User uploads his Form 16.	User ID,Form 16(PDF).	Once the Form16 is successfully uploaded data is extracted from it.	Once the Form16 is successfully uploaded data is extracted from it.	Passed
5.	Investment Questionnaire	User fills up a Questionnaire.	User ID,Questionnaire details.	User is then redirected to the Investment Suggestion module.	User is then redirected to the Investment Suggestion module.	Passed
6.	Asset Allocation	User is suggested assets to invest in.	User ID, Questionnaire details tax calculations.	Based on the user's taxable income and investment Questionnaire the user is allocated appropriate investment assets	Based on the user's taxable income and investment Questionnaire the user is allocated appropriate investment assets	Passed
7.	Portfolio Mnagement	User's Investment Portfolio is managed.	User ID,Questionnaire, tax calculations, assests allocated.	User's Portfolio is properly managed to increase profits and decrease risks.	User's Portfolio is properly managed to increase profits and decrease risks.	Passed
8.	Payment Gateway	User is redirected to Payment Gateway.	User ID, tax calculations.	Once user's Tax calculation process is complete, the user is redirected to Payment Gateway.	Once user's Tax calculation process is complete, the user is redirected to Payment Gateway.	Passed
9.	Logout	User clicks on Logout button and logs out of the system.	User ID.	User clicked on Logout button and logged out of the system.	User clicked on Logout button and logged out of the system.	Passed

5.4 GUI Testing

Table 5.4: GUI testing

TEST ID	TEST CASE	TEST SCENARIO	TEST INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
1.	Login Page	Login Page should be loaded	User ID, Password.	Login Page loaded successfully	Login Page loaded successfully	Passed
2.	Sign Up Page	Sign Up Page should be loaded along with instructions	User details.	Sign Up Page loaded successfully along with instructions	Sign Up Page loaded successfully along with instructions	Passed
3.	Home Page	Home Page should be loaded and it should display all three service options. It should contain logout button as well.	User ID,Module selection.	Home Page loaded successfully and it displayed service options. It contains a logout button and three services to choose from.	Home Page loaded successfully and it displayed service options. It contains a logout button and three services to choose from.	Passed
4.	Tax Automation module	On selection of this service, user is redirected to Tax Automation module.	User ID,Form 16(PDF).	Tax Automation module loads successfully and the Form 16 is successfully uploaded along with the details extracted from it.	Tax Automation module loads successfully and the Form 16 is successfully uploaded along with the details extracted from it.	Passed
5.	Investment Planning and Portfolio Management	After clicking on this service, the user fills a Questionnaire and is then further allocated assets.	User ID,tax calculations, Questionnaire.	After clicking on this service, the Questionnaire is displayed, on successful completion of which, assets are allocated.	After clicking on this service, the Questionnaire is displayed, on successful completion of which, assets are allocated.	Passed

Chapter 6

RESULTS DISCUSSIONS

6.1 Main GUI snapshots

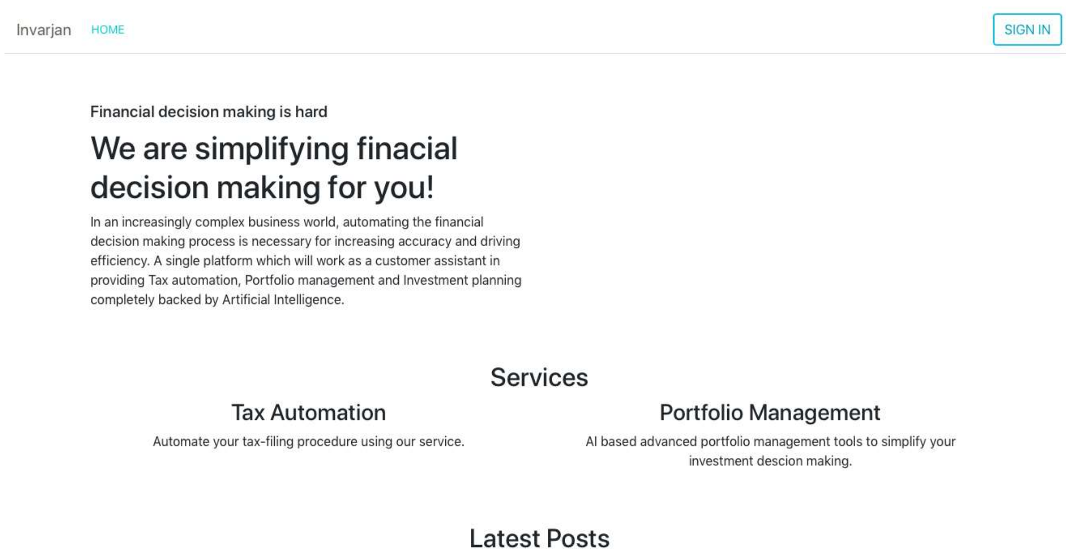

Figure 6.1: Home Page

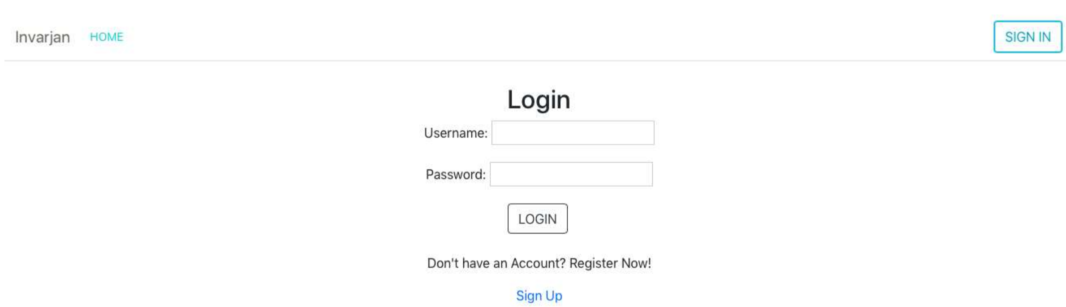

Figure 6.2: Login Page

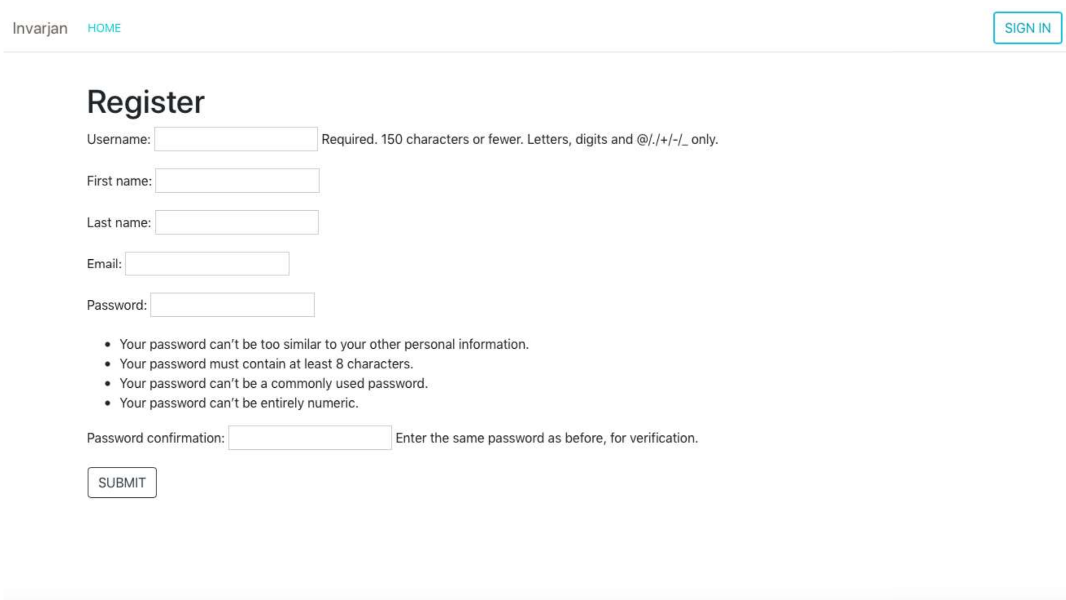

Figure 6.3: Signup Page

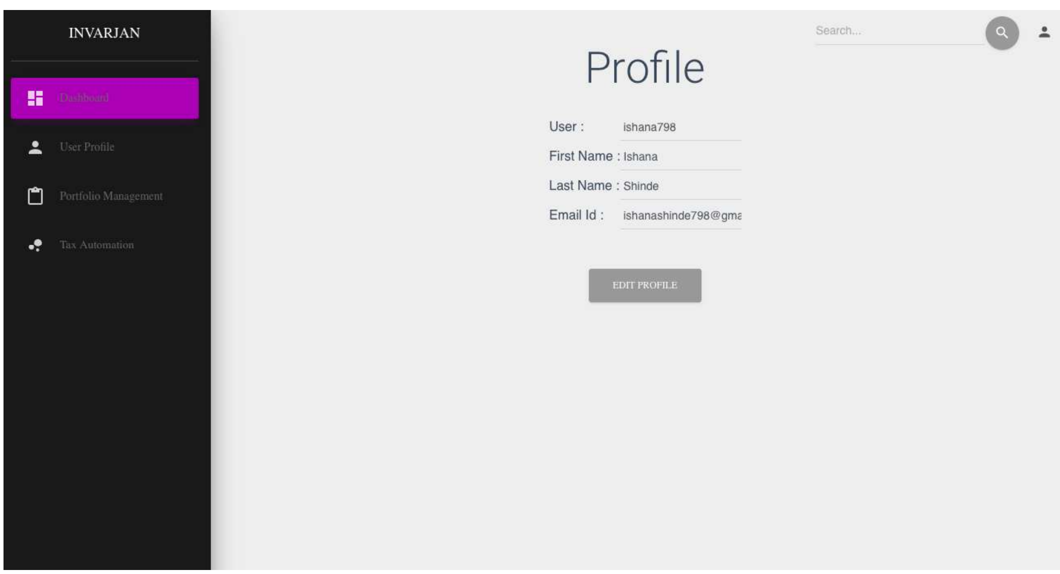

Figure 6.4: Profile

Figure 6.5: Form16 Upload Page

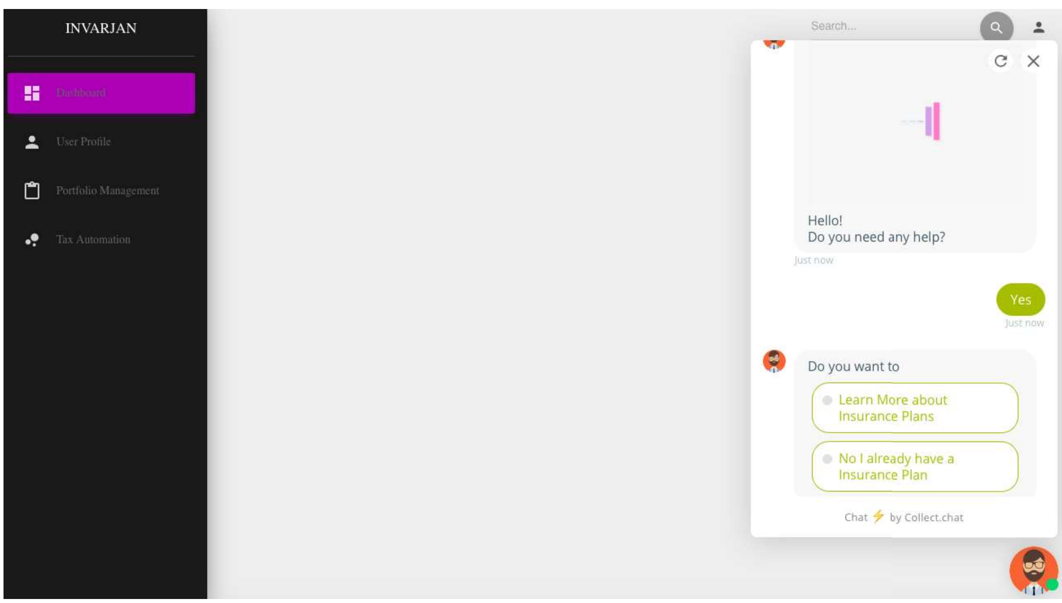

Figure 6.6: Chat-Bot Interface

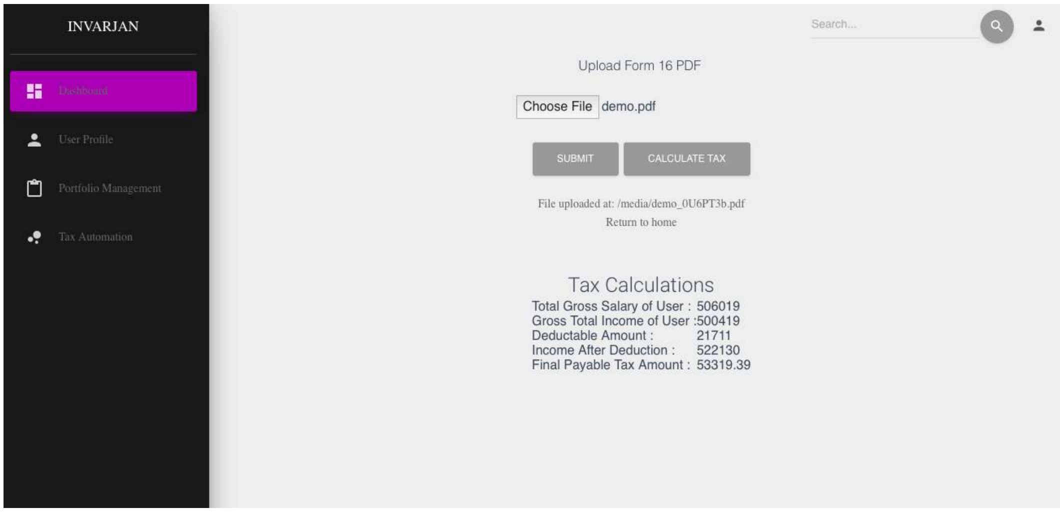

Figure 6.7: Tax Calculation Result Page

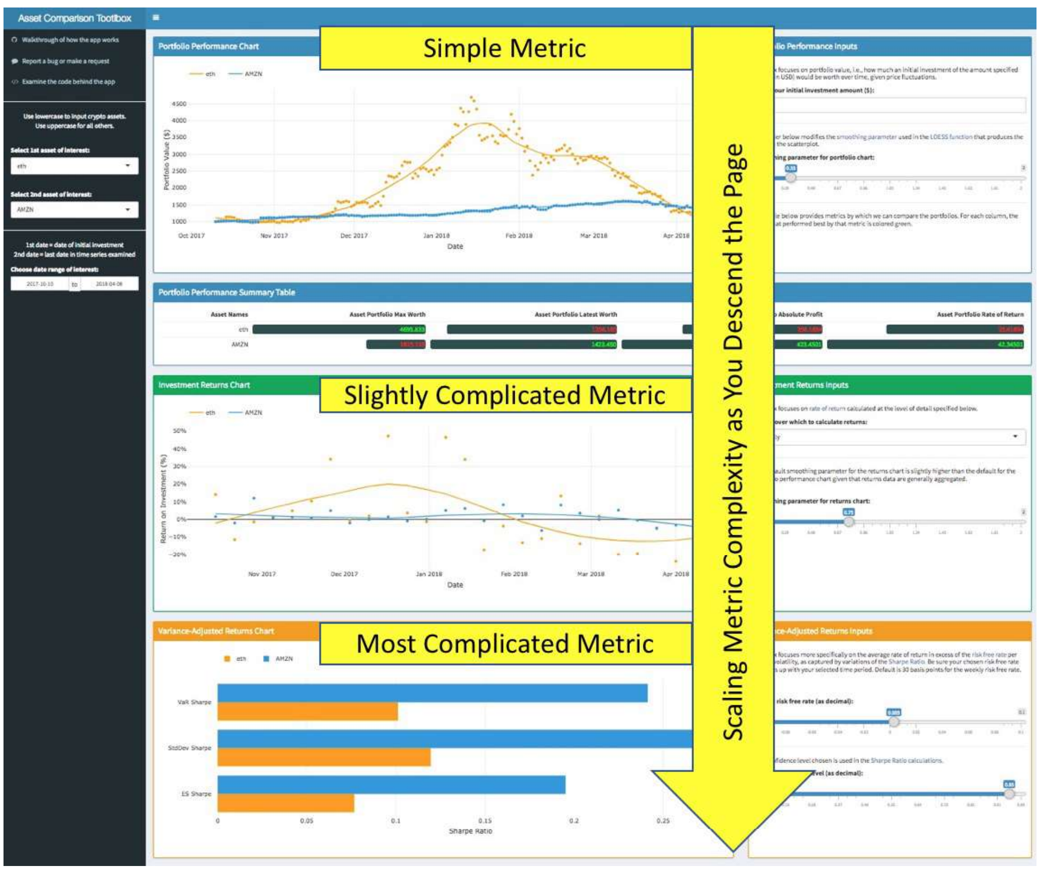

Figure 6.8: User Interface Scaling Complexity

6.2 Table of the results/findings or graphs

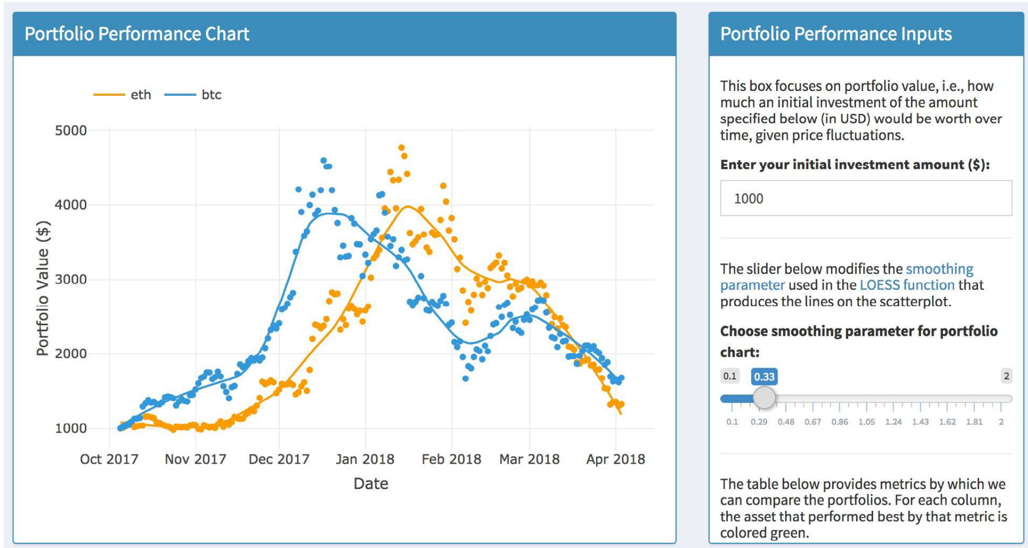

Figure 6.9: Portfolio Box User Interface

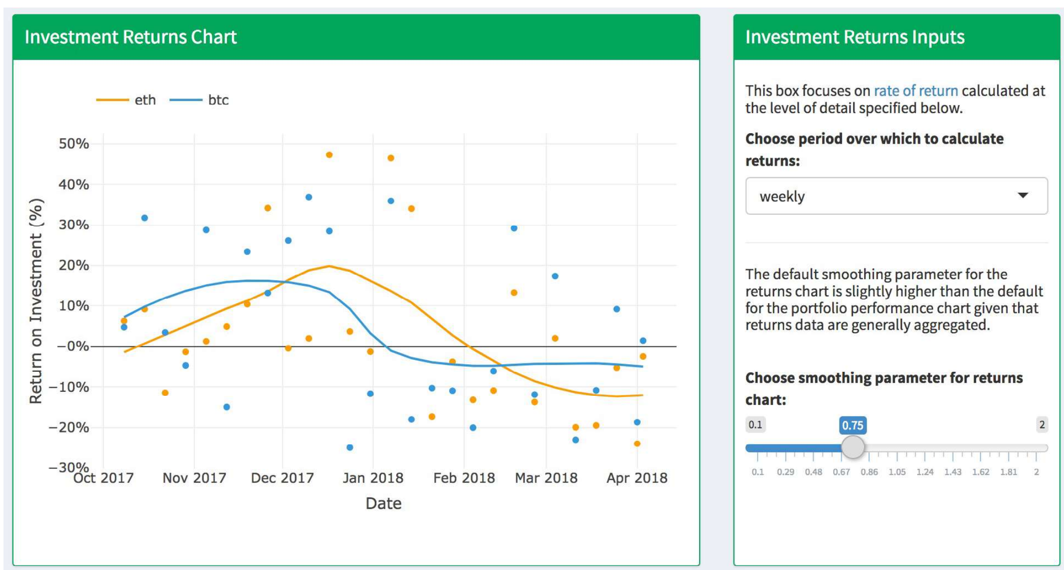

Figure 6.10: Returns Box User Interface

6.3 Discussions

The Homepage describes the basic motive of the site, then directs the user to login into his/her account. If a new user is visiting the site he/she can register and create a new profile. Once logged in, the users can then view their respective profiles. They can even edit their profiles.

If the user wishes to go to the Tax Automation service then, she/he can opt for it on

Figure 6.11: Sharpe Box User Interface

Portfolio Performance Summary Table

Asset Names	Asset Portfolio Max Worth	Asset Portfolio Latest Worth	Asset Portfolio Absolute Profit	Asset Portfolio Rate of Return
eth	4773.303	1322.916	322.9156	32.29156
btc	4604.339	1679.069	679.0689	67.90689

Figure 6.12: Summary Table User Interface

the dashboard.Once selected, the user can then upload a copy of their Form16. The file format for the Form16 can be '.jpeg','.jpg','.png' or '.pdf'. A proper Financial breakdown of the user is then generated. This then helps in creating a proper Investment plan for the user.

In the Portfolio Management section, the user can select two assets of their choice. Once the assets are chosen, the time period for the investment is to be chosen. The model then generates the portfolio and maintains it over the specified period of time, to ensure guaranteed returns at the end of the period.

Chapter 7

CONCLUSION

A comprehensive and integrated system has been successfully created, which unifies the Tax Automation and Investment services under a single umbrella. The Tax Automation platform helps us understand the financial breakup of the customers in a comprehensive way, which in turn helps us to devise a proper Investment Plan suited to the individual's need.

Another sub-module of the Tax Automation system has been successfully designed, which is a new and innovative way to extract data from the Form 16 documents.The implemented module of Table Discovery is an improvement on the conventional OCR (Optical Character Recognition) systems. Such a module helps us eliminate the rigidity of data format that comes along with the conventional OCR systems. Apart from that, the system is also able to extract data from different file formats like '.pdf', '.jpeg', '.jpg' and '.png'.

The Investment Planning and Portfolio Management system successfully creates a comprehensive portfolio suitable to the user's needs and helps in maximizing the profit returns. Various assets are considered while devising the portfolio which provides the user with benefits of diversified investments. Ensemble Learning along with the robust adaptability of Reinforcement Learning in stochastic environments, as in the case of assets like Stocks, helps in creation of a strong Investment agent, which in turn guarantees great returns.

To sum it all up in a nutshell, we think we have achieved our goal of increasing the Financial Literacy of people without confusing them with the unnecessary details. This project of ours is sure to help people with guaranteed returns and thus encourage those who might not be contributing to the Tax System.

Chapter 8

FUTURE WORK

In the current system we have been successful in integrating the Tax Automation and Investment services into a single bracket. The current system can easily break down one's financial hierarchy and successfully suggest a profit yielding Portfolio with maximized returns.

However we would like to work further to achieve the following goals in the near future:

1. The Table Discovery method used to achieve the extraction of data from the Form 16 though is successful in extracting accurate data from printed documents of many different formats, it still needs to be optimised in order to achieve the same accuracy success for the documents that are Hand-written. In case of Hand-written documents, the model sometimes fails to recognize certain characters. Given the proper resources and time, a great accuracy level can be achieved in the future.

2. Currently, the system achieves a diversified investment portfolio that helps to maximise the returns. There still a few assets that could be added to the portfolio list for future versions of the software.

3. Even though the current system caters to the salaried Personal Tax payers perfectly, we would like to Scale Up the Software to accommodate more sophisticated and more complex Professional or Business Tax payers as well.

4. The current system is yet not able to tackle unannounced risks that might occur in the market eg. Corona-virus impact on the market. We would like to make the model even more robust, to tackle such risks in the future, so that our customers are not vulnerable during such desperate times in the slumping markets.

REFERENCES

[1] Yang Wang, Dong Wang, Shiyue Zhang, Yang Feng, Shiyao Li and Qiang Zhou, ”Deep Q-trading”,Dept. Of Computer Science, Tsinghua University, printed in *CSLT TECHNICAL REPORT*, 20160036, January 8, 2017

[2] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai, ”Deep Direct Reinforcement Learning for Financial Signal Representation and Trading”, presented at *IEEE Transactions on Neural Networks and Learning Systems*, 2017

[3] John Moody, Lizhong Wu, Yuansong Liao, and Matthew Saffell, “Performance functions and reinforcement learning for trading systems and portfolios,” printed in *Journal of Forecasting*, vol. 17, no. 56, pp. 441–470, 1998.

[4] Jae Won Lee, Jonghun Park, O Jangmin, Jongwoo Lee, and Euyseok Hong, “A multiagent approach to q-learning for daily stock trading,” presented at *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 6, pp. 864–877, 2007

[5] Authors: Olivier Jin, Hamza El-Saawy, ”Portfolio Management using Reinforcement Learning”, Stanford University, CS229 Project Report, 2016

[6] Gabriel Molina, ”Stock Trading with Recurrent Reinforcement Learning (RRL)”, Stanford University, CS229 Application Project Report, SUID 5055783, 2016

[7] A. Fernndez and S. Gomez. ”Portfolio selection using neural networks”, Stanford University, Computers and Operations Research, 34(4):1177 – 1191, 2007

[8] X. Gao and L. Chan. ”An Algorithm for Trading and Portfolio Management using Q-Learning and Sharpe Ratio Maximization”, In Proceedings of the *”International Conference on Neural Information Processing”*, 2000

[9] J. Moody and M. Saffell, ”Reinforcement Learning for Trading Systems and Portfolios”, *Advances in Computational Management Science*, 2:129–140, 1998.

[10] M. Tokic. ”Adaptive epsilon-greedy exploration in reinforcement learning based on value differences”. In Proceedings of the *”33rd Annual German Conference on Advances in Artificial Intelligence”*, KI’10, pages 203–210, Berlin, Heidelberg, 2010. Springer-Verlag

[11] David Silver, Google DeepMind. Deep Reinforcement Learning (February 24, 2017). [Online Video]. Available: http://techtalks.tv/talks/deep-reinforcement-learning/62360/

[12] International Center for Finance at Yale SOM, Charles Elkan, New Haven Connecticut. AI in Finance: Present and Future, Hype and Reality (October 24, 2018). [Online Video]. Available: https://www.youtube.com/watch?v=C2OZu83gdIoindex=1list=PLNQtfh5xg82JF ML17PU2rhmSmQVjcZDcf/

Ensembling Reinforcement Learning for Portfolio Management

Atharva Abhay Karkhanis[1], Jayesh Bapu Ahire[2], Ishana Vikram Shinde[3], Satyam Kumar[4]

[1,2,3,4]*Dept. of Computer Engineering, Sinhgad College of Engineering (SPPU), Pune, India*

---***---

Abstract - *The Stereoscopic Portfolio optimization Framework introduces the concept of bottom-up optimization via the utilization of machine learning ensembles applied to some market micro-structure element. But it doesn't work always as expected. The popular deep Q learning algorithm is known to be instability because of the Q-value's shake and overestimation action values under certain conditions. These issues tend to adversely affect their performance. Inspired by the breakthroughs in DQN and DRQN, we suggest a modification to the last layers to handle pseudo-continuous action spaces, as required for the portfolio management task. The current implementation, termed the Deep Soft Recurrent Q-Network (DSRQN) relies on a fixed, implicit policy. In this paper, we have described and developed ensembled deep reinforcement learning architecture which uses temporal ensemble to stabilize the training process by reducing the variance of target approximation error and the ensemble of target values reduces the overestimate and makes better performance by estimating more accurate Q-value. Our aggregate architecture leads to more accurate and optimized statistical results for this classical portfolio management and optimization problem.*

Key Words: Reinforcement Learning, Deep Learning, Artificial Intelligence, finance, Algorithmic Trading

1. INTRODUCTION

Reinforcement learning (RL) algorithms are very suitable for learning to regulate associate agent by material possession it moves with associate atmosphere. In recent years, deep neural networks (DNN) have been introduced into reinforcement learning, and that they have achieved an excellent success on the value function approximation. The first deep Q-network (DQN) algorithm which successfully combines a powerful nonlinear function approximation technique known as DNN together with the Q-learning algorithm was proposed by Mnih et al. In this paper, we are proposing experience replay mechanism. Following the DQN work, a variety of solutions are proposed to stabilize the algorithms. The deep Q-networks classes have achieved unprecedented success in challenging domains like Atari 2600 and few different games.

Although DQN algorithms are made in resolution several issues due to their powerful perform approximation ability and powerful generalization between similar state inputs, they're still poor in resolution some problems. Two reasons for this are as follows: (a) the randomness of the sampling is likely to lead to serious shock and (b) these systematic errors might cause instability, poor performance, and sometimes divergence of learning. In order to address these issues, the averaged target DQN (ADQN) algorithm is implemented to construct target values by combining target Q-networks continuously with a single learning network, and the Bootstrapped DQN algorithm is proposed to get more efficient exploration and better performance with the use of several Q-networks learning in parallel. though these algorithms do scale back the overestimate, they are doing not assess the importance of the past learned networks. Besides, high variance in target values combined with the max operator still exists.

There are some ensemble algorithms solving this issue in reinforcement learning, however these existing algorithms don't seem to be compatible with non linearly parameterized value functions.

In this paper, we propose the ensemble algorithm as a solution to this current downside. so as to boost learning speed and final performance, we combine multiple reinforcement learning algorithms in a single agent with several ensemble algorithms to determine the actions or action probabilities. In supervised learning, ensemble algorithms such as bagging, boosting, and mixtures of experts are often used for learning and combining multiple classifiers. But in Reinforcement Learning, ensemble algorithms are used for representing and learning the value function.

Based on an agent integrated with multiple reinforcement learning algorithms, multiple value functions are learned at the identical time. The ensembles mix the policies derived from the value functions in a final policy for the agent. The majority voting (MV), the rank voting (RV), the Boltzmann multiplication (BM), and the Boltzmann addition (BA) are used to combine RL algorithms. Whereas these ways are costly in deep reinforcement learning (DRL) algorithms, we combine different DRL algorithms that learn separate value functions and policies. Therefore, in our ensemble approaches we combine the different policies derived from the update targets learned by deep Q-networks, deep Sarsa networks, double deep Q-networks, and different DRL algorithms. As a consequence, this results in to reduced over-estimations, a lot stable learning method, and improved performance.

2. REINFORCEMENT LEARNING APPLIED TO FINANCE

There are a multitude of papers which have already used Reinforcement Learning in trading stock, portfolio management and portfolio optimization.

Moody et al. were the pioneers in applying the RL paradigm to the problem of stock trading and portfolio optimization. In our references they proposed the idea of Recurrent Reinforcement Learning (RRL) for Direct Reinforcement. RRL is an adaptive policy search algorithm that can learn an investment strategy on-line. Direct Reinforcement was a term coined to show algorithms that don't need to learn a value function in order to derive a policy. In other words, policy gradient algorithms in a Markov Decision Process framework are generally referred to as Direct Reinforcement. Moody et al. showed that a differential form of the Sharpe Ratio and Downside Deviation Ratio can be formulated to enable efficient on-line learning with Direct Reinforcement.

David W. Lu used the idea of Direct Reinforcement with an LSTM learning agent to learn how to trade in a Forex and commodity futures market. Du et al. used value function-based algorithm Q Learning for algorithmic trading. They use different forms of value functions like interval profit, Sharp Ratio and derivative Sharp Ratio to evaluate the performance of the approach.

Tang et al. used an actor-critic based portfolio investment method taking into consideration the risks involved in asset investment. The paper uses approximate dynamic programming to setup a Markov Decision model for the multi-time segment portfolio with transaction cost.

Jiang et al. in his one of the first papers, which provides a detailed Deep Reinforcement Learning framework which can be used in the task of Portfolio Management in a cryptocurrency market exchange. They used the concept of a Portfolio Vector Memory to help train the network, which they call the Ensemble of Identical Independent Evaluators (EIIE). They take into consideration market risks and the transaction costs associated with buying and selling assets in a stock exchange.

3. RECURRENT REINFORCEMENT LEARNING

In this approach the decision-making of investment developed by J. Moody and M. Saffell is considered as a stochastic problem and strategies are directly identified. They have an adaptive algorithm for discovering investment policies, called Recurrent Reinforcement Learning (RRL). Dynamic programming and enhancement algorithms like TD-learning and Q-learning are different from direct enforcement approaches, which try to estimate a value function for the control problem. This facilitates the representation of the problem through the RRL Direct Reinforcement Framework and prevents Bellman's dimensionality and offers convincing efficiency benefits. They demonstrate how direct reinforcement can be used to optimize risk-adjusted returns on investment, taking account costs. They use real financial information intra-daily and find that their RRL-based approach produces better trade strategies than Q-learning systems.

Steve Y. Yang and Saud Almahdi are also taking another approach to solving optimal asset allocation problems and a number of trading decision schemes based on methods of enhanced learning. They establish an optimum allocation of variable weights in line with a consistent downside risk measure E(MDD). The Calmar Ratio, specifies their method using the RRL method for both buying and selling signals and asset allocation weights, with a consistent risk-adjusted performance goal. The expected maximum risk downward-focused objective function is shown through the most frequently traded exchange funds portfolio as a higher return than previously proposed RRL functions (i.e. Sharpe or Sterling Ratio), and variable weight portfolios in various scenarios of transactions cost equal portfolios. NOTE: The Calmar ratio represents a comparison between the average annual compound rate of return and the maximum risk attraction for commodity trading consultants and hedge funds. The smallest the Calmar ratio, the worse the investment was carried over the specified period on a risk-based basis, the higher the Calmar ratio, the better it was.

Deep learning (DL) combined with reinforcement learning in the work of Deng et al., introduced a recurrent deep neural network (NN) for real-time financial signal representation and trading. DL automatically detects the dynamic market conditions for informative learning, and the RL module then interacts with deep representations and decides to accumulate ultimate income in an unknown environment. The system of learning is performed in a complex NN with a highly recurring structure. They thus propose a time-based task-back-cutting to tackle the problem of deep training slowdown. The strength of the neural system is confirmed on both the stock and commodity markets under wide-ranging test conditions.

The RRL approach is clearly different from dynamic programs and strengthening algorithms, such as TD-learning and Q-learning, which try to approximate calculate a value function for the control problem. With the RRL framework, simple, elegant problem representation is created, the dimensionality of Bellman is avoided and efficiency offers compelling advantages: Compared to Q-learning when exposed to rowdy data sets, RRL has a more stable performance. Q-learning algorithm is more sensitive to selecting the value (maybe) because of the recursive dynamic optimization property whereas RRL algorithms can choose the objective function and save time.

4. ENSEMBLE METHODS FOR DEEP REINFORCEMENT LEARNING

As DQN classes use DNNs to approximate the value function, it has strong generalization ability between similar state inputs. The generalization can cause divergence in the case of repeated bootstrapped temporal difference updates. So we can solve this issue by integrating different versions of the target network.

In contrast to a single classifier, ensemble algorithms in a system have been shown to be more effective. They can lead to a higher accuracy. Bagging, boosting, and Ada Boosting are methods to train multiple classifiers. But in RL, ensemble algorithms are used for representing and learning the value function. They are combined by major voting, Rank Voting, Boltzmann Multiplication, mixture model, and other ensemble methods. If the errors of the single classifiers are

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 03 | Mar 2020 www.irjet.net p-ISSN: 2395-0072

not strongly correlated, this can significantly improve the classification accuracy.

5. THE ENSEMBLE NETWORK ARCHITECTURE

The temporal and target values ensemble algorithm (TEDQN) is an integrated architecture of the value-based DRL algorithms. As shown in previous sections, the ensemble network architecture has two parts to avoid divergence and improve performance.

The architecture of our ensemble algorithm is shown in Figure 1; these two parts are combined together by evaluated network.

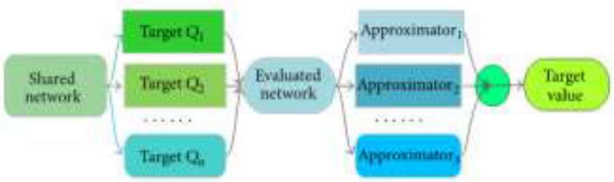

Fig -1: The architecture of the ensemble algorithm

The temporal ensemble stabilizes the training process by reducing the variance of target approximation error [10]. Besides, the ensemble of target values reduces the overestimate and makes better performance by estimating more accurate Q-value. The temporal and target values ensemble algorithm are given by Algorithm 1.

(1) Initialize action-value network Q with random weights θ
(2) Initialize the target neural network buffer $(Q_i)_{i=1}^{L}$
(3) For episode 1, M do
(4) For $t = 1, T$ do
(5) With probability ε select a random action a_t, otherwise $a_t = \mathrm{argmax}_a Q(s_t, a; \theta)$
(6) Execute action a_t in environment and observe reward r_t and next state s_{t+1}, and store transition (s_t, a_t, r_t, s_{t+1}) in D
(7) Sample random *minibatch* of transition (s_t, a_t, r_t, s_{t+1}) from D
(8) set $w_i = \lambda^{i-1} / \sum_{i=1}^{N} \lambda^{i-1}$
(9) Ensemble Q-learner $\widetilde{Q}(s, a; \theta) = \sum_{i=1}^{N} w_i Q_i(s, a; \theta_i)$
(10) set $y_i^{\mathrm{DQN}} = r_t + \gamma \max_a \widetilde{Q}(s_{t+1}, a; \theta_t^-)$
(11) set $y_i^{\mathrm{Sarsa}} = r_t + \gamma \widetilde{Q}(s_{t+1}, a_{t+1}; \theta_t^-)$
(12) set $y_i^{\mathrm{DDQN}} = r_t + \gamma \widetilde{Q}(s_{t+1}, \mathrm{argmax}_{a'} \widetilde{Q}(s_{t+1}, a_{t+1}; \theta_t); \theta_t^-)$
(13) Set $y_j = \{r_j$, if episode terminates at step $j+1$; $\sum_{i=1}^{k} \beta_i y_t^i$, otherwise$\}$
(14) $\theta_i = \underset{\theta}{\mathrm{argmin}}\, E\left[\left(y_{(s,a)}^{j} - Q(s, a; \theta)\right)^2\right]$
(15) Every C steps reset $\widetilde{Q} = Q$
(16) End for
(17) End for

Fig -2: The temporal and target values ensemble algorithm.

As the ensemble network architecture shares the same input-output interface with standard Q-networks and target networks, we can recycle all learning algorithms with Q-networks to train the ensemble architecture.

6. CONCLUSIONS

We introduced a new learning architecture, making temporal extension and the ensemble of target values for deep learning algorithms, while sharing a common learning module. The new ensemble architecture, in combination with some algorithmic improvements, leads to dramatic improvements over existing approaches for deep RL in the challenging classical control issues. In practice, this ensemble architecture can be very convenient to integrate the RL methods based on the approximate value function.

Although the ensemble algorithms are superior to a single reinforcement learning algorithm, it is noted that the computational complexity is higher. The experiments also show that the temporal ensemble makes the training process more stable, and the ensemble of a variety of algorithms makes the estimation of the -value more accurate. The combination of the two ways enables the training to achieve a stable convergence. This is due to the fact that ensembles improve independent algorithms most if the algorithms predictions are less correlated. So that the output of the -network based on the choice of action can achieve balance between exploration and exploitation.

In fact, the independence of the ensemble algorithms and their elements is very important on the performance for ensemble algorithms. In further works, we want to analyze the role of each algorithm and each -network in different stages, so as to further enhance the performance of the ensemble algorithm.

REFERENCES

[1] S. Mozer and M. Hasselmo, “Reinforcement learning: an introduction,” IEEE Transactions on Neural Networks and Learning Systems, vol. 16, no. 1, pp. 285-286, 2005. View at Publisher View at Google Scholar

[2] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: a survey,” Journal of Artificial Intelligence Research, vol. 4, pp. 237–285, 1996. View at Google Scholar · View at ScopusR. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[3] V. Mnih, K. Kavukcuoglu, D. Silver et al., “Playing Atari with deep reinforcement learning [EB/OL],” https://arxiv.org/abs/1312.5602.

[4] M. A. Wiering and H. van Hasselt, “Ensemble algorithms in reinforcement learning,” IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 38, no. 4, pp. 930–936, 2008. View at Publisher View at Google Scholar View at Scopus

[5] S. Whiteson and P. Stone, “Evolutionary function approximation for reinforcement learning,” Journal of Machine Learning Research (JMLR), vol. 7, pp. 877–917, 2006. View at Google Scholar View at MathSciNet

[6] P. Preux, S. Girgin, and M. Loth, “Feature discovery in approximate dynamic programming,” in Proceedings of the 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, ADPRL 2009, pp. 109–116, April 2009. View at Publisher View at Google Scholar View at Scopus

[7] T. Degris, P. M. Pilarski, and R. S. Sutton, “Model-Free reinforcement learning with continuous action in

practice," in Proceedings of the 2012 American Control Conference, ACC 2012, pp. 2177–2182, June 2012. View at Scopus

[8] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015. View at Publisher View at Google Scholar View at Scopus

[9] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-Learning," in Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016, pp. 2094–2100, February 2016. View at Scopus

[10] O. Anschel, N. Baram, N. Shimkin et al., "Averaged-DQN: Variance Reduction and Stabilization for Deep Reinforcement Learning [EB/OL]," https://arxiv.org/abs/1611.01929.

[11] I. Osband, C. Blundell, A. Pritzel et al., "Deep Exploration via Bootstrapped DQN [EB/OL]," https://arxiv.org/abs/1602.04621.

[12] S. Faußer and F. Schwenker, "Ensemble Methods for Reinforcement Learning with Function Approximation," in Multiple Classifier Systems, pp. 56–65, Springer, Berlin, Germany, 2011. View at Google Scholar

[13] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: a review," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 4–37, 2000. View at Publisher View at Google Scholar View at Scopus

[14] T. Schaul, J. Quan, I. Antonoglou et al., "Prioritized Experience Replay [EB/OL]," https://arxiv.org/abs/1511.05952.

[15] I. Zamora, N. G. Lopez, V. M. Vilches et al., "Extending the OpenAI Gym for robotics: a toolkit for reinforcement learning using ROS and Gazebo [EB/OL]," https://arxiv.org/abs/1608.05742.

[16] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," Journal of Machine Learning Research (JMLR), vol. 6, no. 2, pp. 503–556, 2005. View at Google Scholar View at MathSciNet