MODELING AND PREDICTING CYBER HACKING BREACHES

*A Project Report Submitted in partial fulfillment of the requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

In

## COMPUTER SCIENCE AND ENGINEERING

***Submitted By***

**PITANI VENKATA SURESH 21221A05D4**

**ROSHAN ADHIKARI 21221A05D6**

**UJJWAL TIMILSINA 21221A05D8**

**POTHURAJU REKHA 22225A0509**

**Under the Esteemed Guidance of**

**Mr. B. NARASIMHA RAO**

**M.Tech.,**

**Associate Professor**

****

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE

**(AUTONOMOUS)**

(Approved by A.I.C.T.E, New Delhi & Affiliated to J.N.T.U.K, Kakinada) (Accredited by NBA &NAAC with “A‟ Grade)

**ODALAREVU-533210 2021 – 2025**

## BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE

**(AUTONOMOUS) ODALAREVU-533210**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

****

CERTIFICATE

This is to certify that the project work entitled **“MODELING AND PREDICTING CYBER HACKING BREACHES”** is being submitted for the partial fulfillment of the requirements for the award of the degree of Bachelor Of Technology in **Computer Science and Engineering** to Bonam Venkata Chalamayya Engineering College(A), Odalarevu, is a Bonafide work done by **PITANI VENKATA SURESH (21221A05D4), ROSHAN ADHIKARI (21221A05D6), UJJWAL TIMILSINA**

**(21221A05D8), POTHURAJU REKHA (22225A0509)** under my guidance during the academic year **2021-2025** and it has been found suitable for acceptance according to the requirement of the University.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree.

**Project Guide Head of the Department**

**Mr. B. NARASIMHA RAO Dr. MEERASHARIF SHEIK**

**M.Tech., M.Tech., Ph.D.,**

**Associate Professor Professor & HOD**

## EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

We profoundly grateful to express our deep sense of gratitude and respect towards our guide **Mr. B. NARASIMHA RAO**, **M.Tech., Associate Professor** of CSE Department, Bonam Venkata Chalamayya Engineering College(A), Odalarevu, for his excellent guidance right from selection of the project and his valuable suggestions throughout the project work. His constant encouragement and support has been the cause of our success, in completing this thesis in the college.

We also grateful to the Project Coordinator **Mr. M.V.K. SUBASH, M.Tech(Ph.D.)**, **Assistant Professor** of CSE Department, Bonam Venkata Chalamayya Engineering College(A), Odalarevu, for her excellent guidance right from selection of the project and her valuable suggestions throughout the project work.

We really thankful to **Dr. MEERASHARIF SHEIK, M.Tech., Ph.D., Professor** and **Head of the Department for Computer Science and Engineering** for providing the laboratory facilities to the fullest extent as and when required and also for giving us the opportunity to carry out the project work in the college.

We also thankful to **Dr. MAHESWAR DUTTA**, **M.E., Ph.D., Principal,** Bonam Venkata Chalamayya Engineering College(A), Odalarevu, for his support during and till the completion of the project.

We are also thankful to our **MANAGEMENT,** Bonam Venkata Chalamayya Engineering College(A), Odalarevu, for their support during and till the completion of the project.

We are thankful to all the **TEACHING AND NON-TEACHING STAFF** of Computer Science and Engineering Department, Bonam Venkata Chalamayya Engineering College(A), Odalarevu, and **FRIENDS** for their direct and indirect help provided to us in completing the project.

Last but not least, we own all our success to our **FAMILY MEMBERS, CLASS MATES AND TEACHERS** from our child hood, whose vision, love and inspiration made us to reach out for these glories.

**Project Associates………**

**PITANI VENKATA SURESH 21221A05D4 ROSHAN ADHIKARI 21221A05D6**

**UJJWAL TIMILSINA 21221A05D8**

**POTHURAJU REKHA 22225A0509**

# ABSTRACT

Analyzing cyber incident data sets is an important method for deepening our understanding of the evolution of the threat situation. This is a relatively new research topic, and many studies remain to be done. In this paper, we report a statistical analysis of a breach incident data set corresponding to 12 years (2005-2017) of cyber hacking activities that include malware attacks. We show that, in contrast to the findings reported in the literature, both hacking breach incident inter-arrival times and breach sizes should be modeled by stochastic processes, rather than by distributions because they exhibit autocorrelations. Then, we propose particular stochastic process models to, respectively, fit the inter-arrival times and the breach sizes. We also show that these models can predict the inter-arrival times and the breach sizes. In order to get deeper insights into the evolution of hacking breach incidents, we conduct both qualitative and quantitative trend analyses on the data set. We draw a set of cybersecurity insights, including that the threat of cyber hacks is indeed getting worse in terms of their frequency, but not in terms of the magnitude of their damage.

# TABLE OF CONTENTS

[ABSTRACT 4](#_bookmark0)

[TABLE OF CONTENTS 5](#_bookmark1)

[List of Figures. 7](#_TOC_250000)

[CHAPTER 1](#_bookmark2)

1. [INTRODUCTION 9](#_bookmark3)
   1. [OBJECTIVE OF THE PROJECT 9](#_bookmark4)
   2. [THE EXISTING SYSTEM 9](#_bookmark5)
   3. [THE PROPOSED SYSTEM 1](#_bookmark6)0

[CHAPTER 2](#_bookmark7)

1. [SYSTEM ANALYSIS AND DESIGN 1](#_bookmark8)2
   1. [SRS 1](#_bookmark9)2
      1. [Performance Requirements 1](#_bookmark10)2
      2. [Software Quality Attributes 1](#_bookmark11)2
   2. [Software Design 1](#_bookmark12)3
      1. [Modules 1](#_bookmark13)4
      2. [UML Diagrams 1](#_bookmark14)4

[Types of UML Diagrams 1](#_bookmark15)5

[CHAPTER 3](#_bookmark16)

1. [SCREENS 2](#_bookmark17)0

[CHAPTER 4](#_bookmark18)

1. [TESTING 2](#_bookmark19)6
   1. [Unit Testing 2](#_bookmark20)6
   2. [Integration Testing 2](#_bookmark21)7
   3. [User Acceptance Testing 2](#_bookmark22)8
   4. [System Testing 2](#_bookmark23)8
   5. [Test Cases 2](#_bookmark24)9

[CHAPTER 5](#_bookmark25)

1. [SOFTWARE ENVIRONMENT 3](#_bookmark26)0

[What is Python : 3](#_bookmark27)0

[Advantages of Python : 3](#_bookmark28)0

[Advantages of Python Over Other Languages 3](#_bookmark29)3

[Disadvantages of Python 3](#_bookmark30)3

[History of Python : 3](#_bookmark31)5

[What is Machine Learning : 3](#_bookmark32)5

[Categories Of Machine Leaning : 3](#_bookmark33)6

[Need for Machine Learning 3](#_bookmark34)7

[Challenges in Machines Learning : 3](#_bookmark35)7

[Applications of Machines Learning : 3](#_bookmark36)8

[How to Start Learning Machine Learning? 3](#_bookmark37)9

[How to start learning ML? 3](#_bookmark38)9

[Advantages of Machine learning : 4](#_bookmark39)2

[Disadvantages of Machine Learning : 4](#_bookmark40)3

[Modules Used in Project : 4](#_bookmark41)6

[CHAPTER 6](#_bookmark42)

1. [CONCLUSION AND FUTUREWORK 5](#_bookmark43)6
   1. [CONCLUSION 5](#_bookmark44)6
   2. [FUTUREWORK: 5](#_bookmark45)6

[CHAPTER 7](#_bookmark46)

1. [REFERENCE 5](#_bookmark47)7

# List of Figures

Fig 2.1: Use Case Diagram… 16

Fig 2.2: Class Diagram 16

Fig 2.3: Sequence Diagram… 17

Fig 2.4: Activity Diagram… 18

Fig 3.1: User Login… 19

Fig 3.2: User Homepage 19

Fig 3.3: User Analysis… 20

Fig 3.4: Malware Data… 20

Fig 3.5: Unmalware Data 21

Fig 3.6: Breaches Analysis… 21

Fig 3.7: Graphical Analysis 22

Fig 3.8: Bar Chart… 22

Fig 3.9: Column Chart… 23

Fig 3.10: Admin Login… 23

Fig 3.11: User details Analysis… 24

Fig 3.12: Admin Analysis… 24

Fig 5.1: Open Python website to download the python latest version… 48

Fig 5.2: Click on Download Python… 49

Fig 5.3: Select the version what you want… 49

Fig 5.4: Python versions 50

Fig 5.5: Click on OK to install Python… 51

Fig 5.6: Click on Install Now to install… 51

Fig 5.7: Successful Installation of Python… 52

Fig 5.8: Open Command Prompt… 52

Fig 5.9: Successful installation check… 53

Fig 5.10: Click on IDLE (Python 3.7 64-bit) and launch the program… 53

Fig 5.11: Saving the file 54

# CHAPTER 1

# INTRODUCTION

## OBJECTIVE OF THE PROJECT

The present study is motivated by several questions that have not been investigated until now, such as: Are data breaches caused by cyber-attacks increasing, decreasing, or stabilizing? A principled answer to this question will give us a clear insight into the overall situation of cyber threats. This question was not answered by previous studies. Specifically, the dataset analyzed in [7] only covered the time span from 2000 to 2008 and does not necessarily contain the breach incidents that are caused by cyber-attacks; the dataset analyzed in [9] is more recent, but contains two kinds of incidents: negligent breaches (i.e., incidents caused by lost, discarded, stolen devices and other reasons) and malicious breaching. Since negligent breaches represent more human errors than cyber- attacks, we do not consider them in the present study. Because the malicious breaches studied in [9] contain four sub-categories: hacking (including malware), insider, payment card fraud, and unknown, this study will focus on the hacking sub-category (called hacking breach dataset thereafter), while noting that the other three sub- categories are interesting on their own and should be analyzed separately.

## THE EXISTING SYSTEM

The current study is done by several key research questions which have been remained untouched until now in the cyber security community. One of the main questions for the research is: *Are data breaches caused by cyber attacks becoming more frequent, less frequent, or they are staying the same over time?* While answering this question we can get the terrifying insight of the evolution of the cyber threats. So to be prepared for this type of cyber threats the dynamics of these are essential for the effective cyber security policies, managing resources for defense strategies, and raising awareness about long-term trends in malicious digital activities.

The research done previous to it has not fully countered the above raised question in the study span from year 2000 to 2008. The dataset of this time interval is relatively outdated and is very limited in scope. It doesn’t specifically distinguish whether the data breaches included were a direct result of cyber attacks or not. So, the conclusions form these doesn’t reflect the true nature of evolving cyber threat patterns, as the frequency and complicated type of cyber attacks have been increased.

A recent study based on dataset checks a broader and more current set of breaches, classifying them into malicious and harmless breaches. Harmless breaches which are caused by human errors like lost devices or accidental data exposure, are not included in this study as they don’t represent the intentional cyber threats.Within the malicious breach category, four sub-types are identified. This research mainly focus on the hacking breach dataset which focuses modern, deliberate cyber attacks involving methods such as malware, unauthorized access, and vulnerability exploitation. Others sub-categories are also important but in this study they are excluded which allows a more accurate analysis of hacking trends and the development of models to understand and predict future cyber threats.

## THE PROPOSED SYSTEM

In this paper, we make the following three contributions. First, we show that both the hacking breach incident interarrival times (reflecting incident frequency) and breach sizes should be modeled by stochastic processes, rather than by distributions. We find that a particular point process can adequately describe the evolution of the hacking breach incidents inter-arrival times and that a particular ARMA-GARCH model can adequately describe the evolution of the hacking breach sizes, where ARMA is acronym for “AutoRegressive and Moving Average” and GARCH is acronym for “Generalized AutoRegressive Conditional Heteroskedasticity.” We show that these stochastic process models can predict the inter-arrival times and the breach sizes. To the best of our knowledge, this is the first paper showing that stochastic processes, rather than distributions, should be used to model these cyber threat factors. Second, we discover a positive dependence between the incidents inter-arrival times and the breach sizes, and show that this dependence can be adequately described by a particular

copula. We also show that when predicting inter-arrival times and breach sizes, it is necessary to consider the dependence; otherwise, the prediction results are not accurate. To the best of our knowledge, this is the first work showing the existence of this dependence and the consequence of ignoring it. Third, we conduct both qualitative and quantitative trend analyses of the cyber hacking breach incidents. We find that the situation is indeed getting worse in terms of the incidents inter-arrival time because hacking breach incidents become more and more frequent, but the situation is stabilizing in terms of the incident breach size, indicating that the damage of individual hacking breach incidents will not get much worse. We hope the present study will inspire more investigations, which can offer deep insights into alternate risk mitigation approaches. Such insights are useful to insurance companies, government agencies, and regulators because they need to deeply understand the nature of data breach risks.

# CHAPTER 2

# SYSTEM ANALYSIS AND DESIGN

System analysis is the performance management and documentation of activities related to the life cycle phases of any software namely:

* The Study Phase
* The Design Phase
* The Development Phase
* The Implementation Phase
* The Testing Phase

Software Analysis starts with a preliminary analysis and later switches on to a detailed one. During the preliminary analysis the Analyst takes a quick look at what is needed and whether the cost benefits. Detailed analysis studies in depth all the cornered factors, which builds and strengthens the software.

## SRS

SRS (Software Requirement Specification) is a document that completely describes what the proposed should do, without describing how the software does it.

### Performance Requirements

1. The operation time should be small and the throughput should be high..
2. It should produce timely and accurate result.

### Software Quality Attributes

* 1. **Maintainability** – Since it is directly associated with the database, so there is very little maintainability problem with this application.
  2. **Easy to Learn –** Since there are less number of forms, this application is very easy to learn with user-friendly screens.
  3. **Flexibility** – This application is very much flexible for future enhancements.

### Software Design

System design is the second step in the system life cycle, in which overall design of the system is achieved. The functionalities of the system is designed and studied in this phase. The first step is designing of program specification. This determines the various data inputs to the system, data flow and the format in which output is to be obtained.

Design phase is a transmission phase because it is a transition from user oriented document to computer data. The activity in the design phase is the allocation of functions to manual operations, equipment and computer programs. Flow charts are prepared in the study time and is decomposed until all functions in the system perform evidently.

Design is a multi-step process that focuses on data structures, software architecture, procedural details and links between the modules. The design process goes through logical and physical stages. In logical design reviews are made linking existing system and specification gathered. The physical plan specifies any hardware and software requirement, which satisfies the local design.

Modularization of task is made in this phase. The success of any integrated system depends on the planning of each and every fundamental module. Usually a project is revised in step by step sequence. Inter-phase management of such module is also important. Software design methodology changes continually as new methods, better analysis and broader understanding evolve.

Various techniques for software design do exit with the availability of criteria for design quality. Software design leads three technical activities-design, code and test.Each activity transforms information, which validates the software. The design

system converts theoretical solution introduced by the feasibility study into a logical reality.

### Modules ADMIN:

In this module Admin will predict malware in this application.

After login admin can view user data prediction, admin analysis and graphical analysis.

## USER:

In this module user will data, Analysis, malware data, unmalware data, branched analysis and graphical Analysis

### UML Diagrams

UML stands for **Unified Modeling Language**. This object-oriented system of notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar Jacobson, and the [Rational Software Corporation](http://www.rational.com/). These renowned computer scientists fused their respective technologies into a single, standardized model. Today, UML is accepted by the [Object Management Group (OMG)](http://www.omg.org/) as the standard for modeling object oriented programs.

There are three classifications of UML diagrams:

* + - * **Behavior diagrams:** A type of diagram that depicts behavioral features of a system or business process. This includes activity, state machine, and use case diagrams as well as the four interaction diagrams.
      * **Interaction diagrams:** A subset of behavior diagrams which emphasize object interactions. This includes communication, interaction overview, sequence, and timing diagrams.
      * **Structure diagrams:** A type of diagram that depicts the elements of a specification that are irrespective of time. This includes class, composite structure, component, deployment, object, and package diagrams.

### Types of UML Diagrams

UML defines nine types of diagrams: class (package), object, use case, sequence, collaboration, state chart, activity, component, and deployment.

### [Class Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Class-Diagrams)

Class diagrams are the backbone of almost every object-oriented method, including UML. They describe the static structure of a system.

### [Package Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Package-Diagrams)

Package diagrams are a subset of class diagrams, but developers sometimes treat them as a separate technique. Package diagrams organize elements of a system into related groups to minimize dependencies between packages.

### [Object Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Object-Diagrams)

Object diagrams describe the static structure of a system at a particular time. They can be used to test class diagrams for accuracy.

### [Use Case Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Use-Case-Diagram)

Use case diagrams model the functionality of system using actors and use cases.

### [Sequence Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Sequence-Diagram)

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.

### [Collaboration Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Collaboration-Diagrams)

Collaboration diagrams represent interactions between objects as a series of sequenced messages. Collaboration diagrams describe both the static structure and the dynamic behavior of a system.

### [Statechart Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Statechart-Diagrams)

Statechart diagrams describe the dynamic behavior of a system in response to external stimuli. Statechart diagrams are especially useful in modeling reactive objects whose states are triggered by specific events.

### [Activity Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Activity-Diagram)

Activity diagrams illustrate the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation.

### [Component Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Component-Diagram)

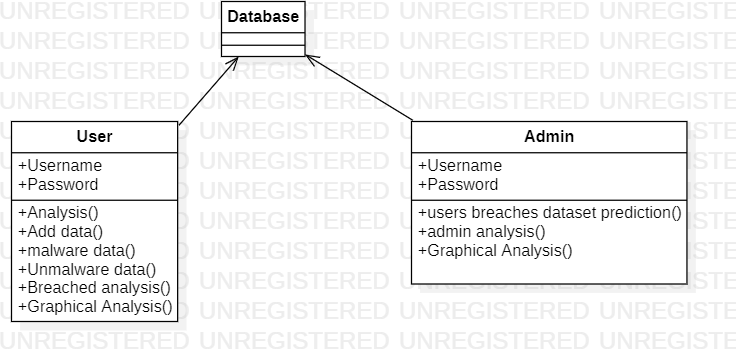
Component diagrams describe the organization of physical software components, including source code, run-time (binary) code, and executables.

### [Deployment Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Deployment-Diagrams)

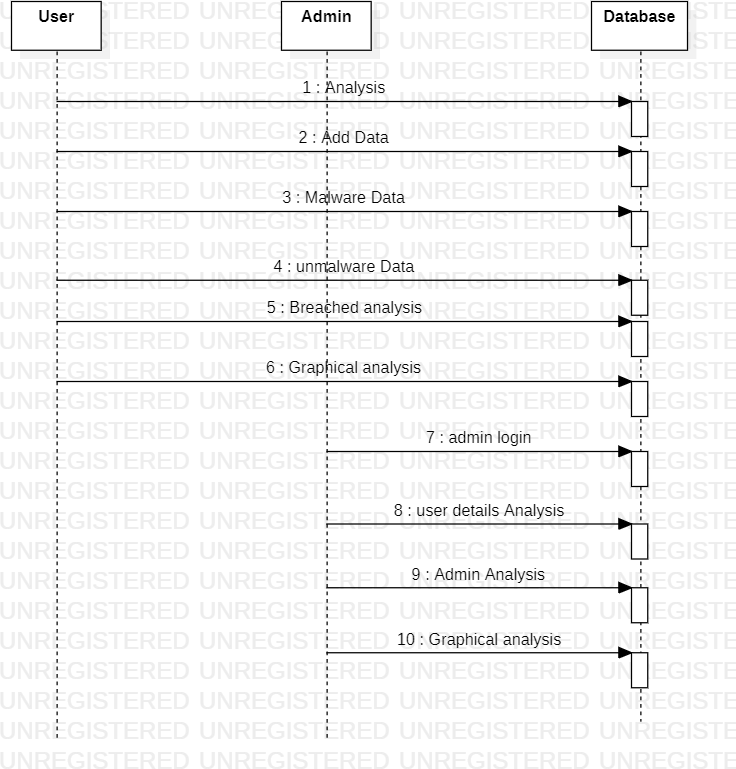
Deployment diagrams depict the physical resources in a system, including nodes, components, and connections.



**Fig 2.1: Use Case Diagram**

****

**Fig 2.2: Class Diagram**



**Fig 2.3: Sequence Diagram**



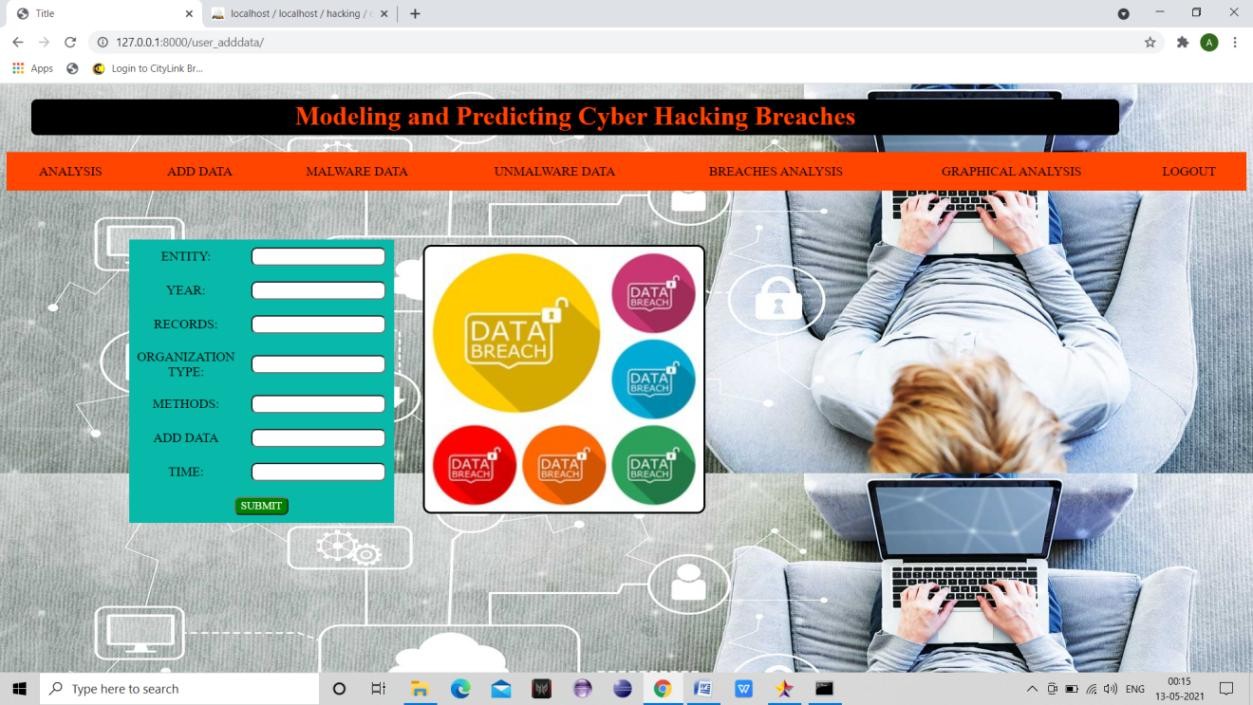
**Fig 2.4: Activity Diagram**

# CHAPTER 3

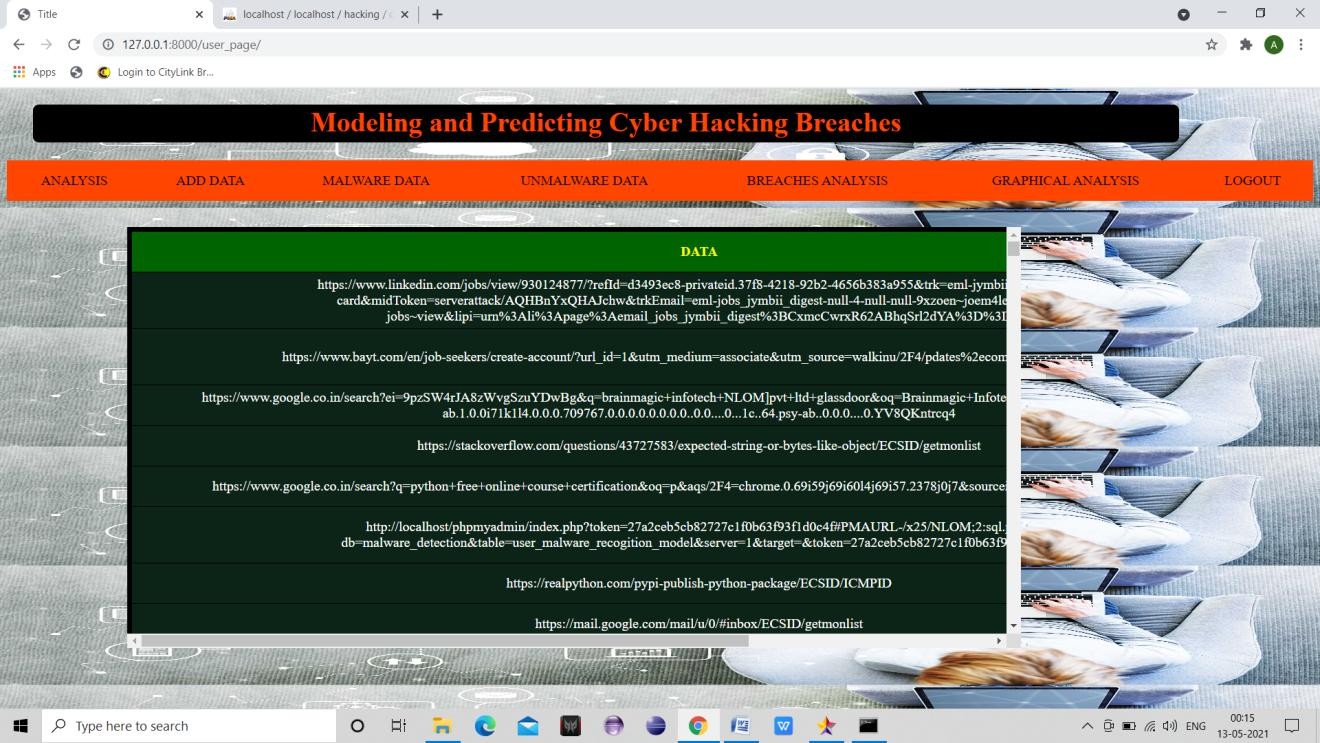
# SCREENS

****

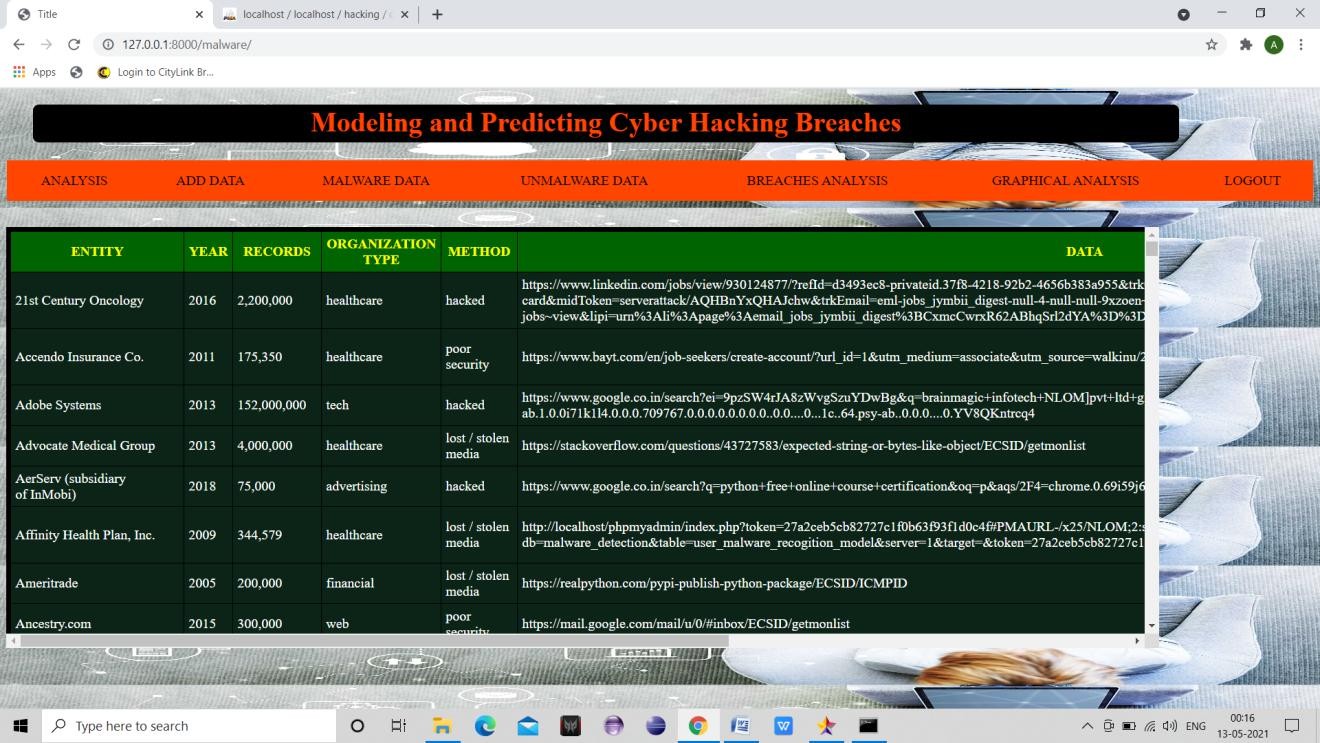
**Fig 3.1: User Login**

****

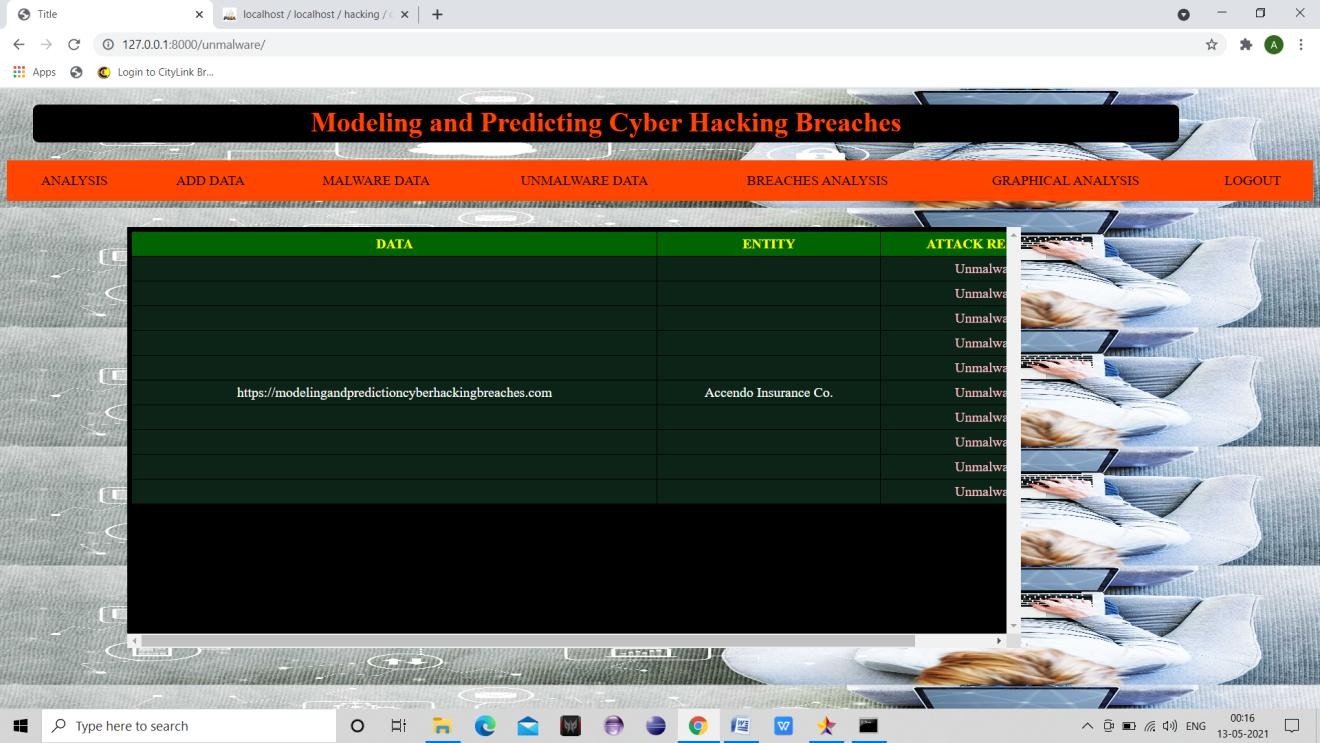
**Fig 3.2: User Home Page**



**Fig 3.3: User Analysis**

****

**Fig 3.4: Malware Data**



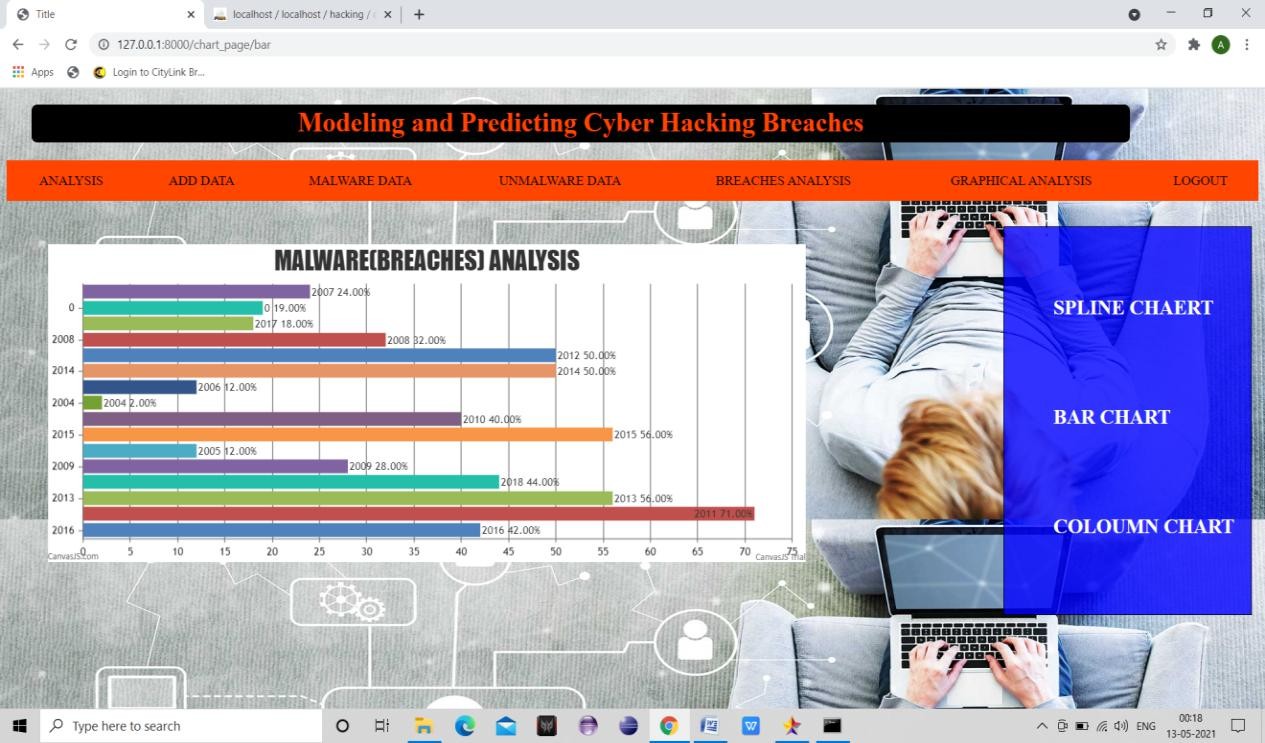
**Fig 3.5: Unmalware Data**

****

**Fig 3.6: Breaches Analysis**



**Fig 3.7: Graphical Analysis**

****

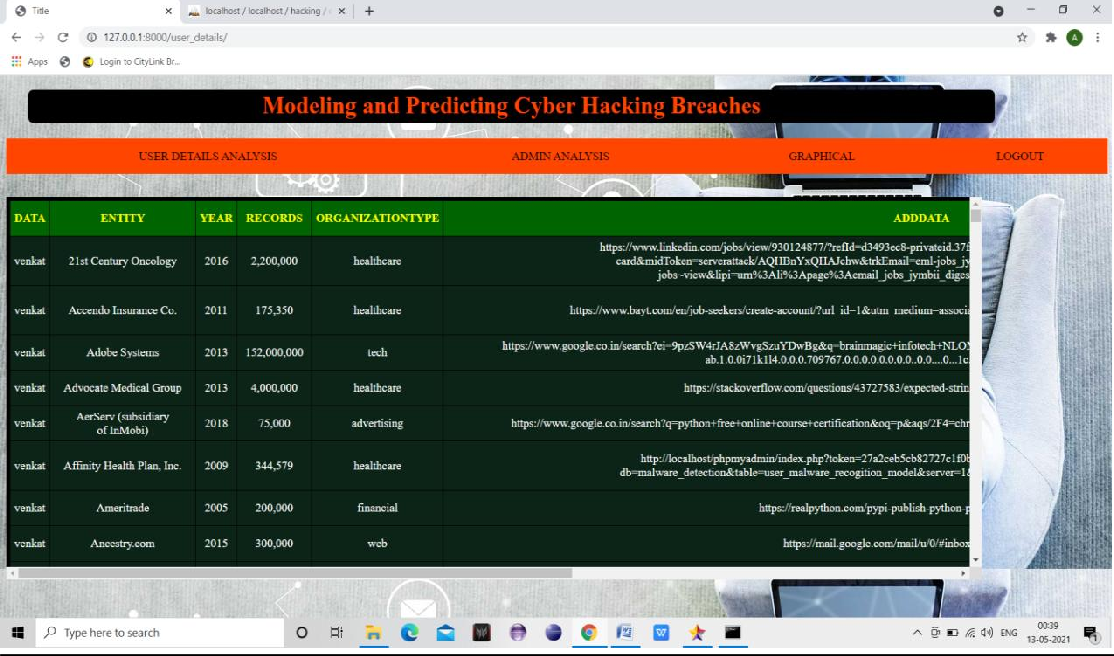
**Fig 3.8: Bar Chart**



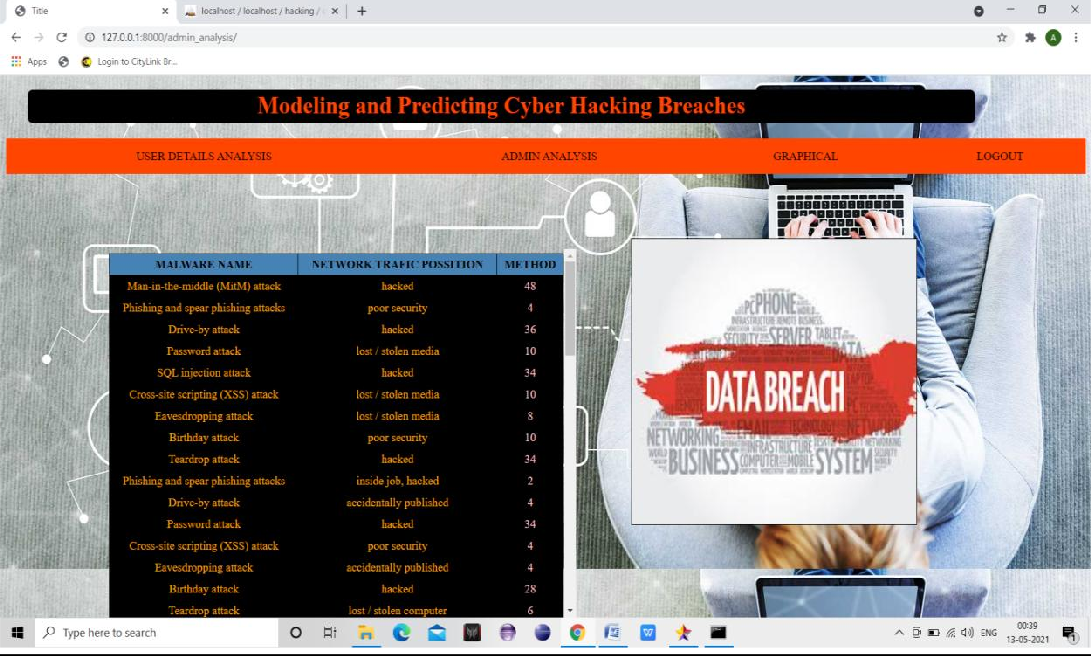
**Fig 3.9: Column Chart**

****

**Fig 3.10: Admin Login**



**Fig 3.11: User details Analysis**

****

**Fig 3.12: Admin Analysis**

# CHAPTER 4

# TESTING

Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification, design and coding. Testing presents an interesting anomaly of the software. During earlier definition and development phases, it was attempted to build software from abstract concept to a tangible implementation.

The testing phase involves the testing of the developed system using various set data. Presentation of test data plays a vital role in system testing. After preparing the test data the system under study was tested using test data. While testing the system by using test data errors were found and corrected. A series of tests were performed for the proposed system before the system was ready for implementation. The various types of testing done on the system are:

* Unit Testing
* Integration Testing
* User Acceptance Testing
* System Testing

### Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. It comprises the set of test performed by the programmer prior to integration of the unit into larger system. The testing was carried out during the coding stage itself. In this step each module is found to be working satisfactorily as regards to the expected output from the module**.**

Each form is treated as a unit and tested thoroughly for bugs. The following is a list of some of the test cases :

1. In the login form, if a member does not enter a value for userId and password, then the user is prompted with the error message “userId and password should not be blank”.
2. In the login form, if a member enters wrong values for userId and password, then the user is prompted with the error message “Invalid userId and password. Try again.”.
3. In book Entry screen and new student, teacher screen, all the fields should have a value. Otherwise, the user is prompted with an appropriate error messages.
4. In book transactions form, member id, book no,. issue date, and return date are mandatory. If not provided, then the system will prompt the user with the error message “Fields should not be blank”.

### Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover error associated within the interface. The objective is to take unit tested modules and build a program structure that has been dictated by design. All modules are combined in this step. The entire program is tested as whole. And chaos in interfaces may usually result. A set of errors is encountered in such a case.

The integration testing can be carried out using two methodologies: # Top Down Integration

# Bottom Up Integration

The first one is done where integration is carried out by addition of major modules to minor modules. While Bottom Up integration follows combination of smaller ones to larger one. Here, Bottom Up Integration is followed. Even though correction was difficult because the isolation of causes is complicated by the vastness of the entire program, all the errors found in the system were corrected and then forwarded to the next testing steps.

The navigation among all the screens have been thoroughly verified so that the user of the system can move from one form to another form.The connectivity between the forms and the database has been checked. In case of any malfunctions, the user will be informed about the problem.

### User Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration was tested for users acceptance by constantly keeping in

touch with the perspective system user at the time of developing and making changes wherever required. This is done with the regards to the following points:

A system may be defined as a set of instructions combined in the same form and directed to some purpose.

Before any development is undertaken certain specifications are prepared which objectively describe the application system. The System specifications are made after consulting the end user managers of the relevant departments.

Software to be developed is planned on the basis of requirement of the user. The problem definition statement description of present situation and goal to be achieved by news system.

The success of system depends on how accurately a problem is defined, thoroughly investigated carried out through choice of solution. User need identification and analysis that are concerned with what the uses needs rather than what he/she wants. System explains how to perform specific activities or task, which does what and what.

### System Testing

Testing the behavior of the whole software/system as defined in software requirements specification(SRS) is known as system testing, its main focus is to verify that the customer requirements are fulfilled.

System testing is done after integration testing is complete. System testing should test functional and non functional requirements of the software. The test types followed in system testing differ from organization to organization.

The project is executed and tested on the machines that satisfy the given hardware and software requirements. It was executed successfully with the specified hardware and operating system.

### Test Cases

**Test case for Login form:**

|  |  |
| --- | --- |
| **FUNCTION:** | **LOGIN** |
| **EXPECTED RESULTS:** | Should Validate the user and check his existence in database |
| **ACTUAL RESULTS:** | Validate the user and checking the user against the database |
| **LOW PRIORITY** | **No** |
| **HIGH PRIORITY** | **Yes** |

**Test case for User Registration form:**

|  |  |
| --- | --- |
| **FUNCTION:** | **USER REGISTRATION** |
| **EXPECTED RESULTS:** | Should check if all the fields are filled by the user and saving the user to database. |
| **ACTUAL RESULTS:** | Checking whether all the fields are field by user or not through validations and saving user. |
| **LOW PRIORITY** | **No** |
| **HIGH PRIORITY** | **Yes** |

# CHAPTER 5

# SOFTWARE ENVIRONMENT

### What is Python :-

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
* GUI Applications (like Kivy, Tkinter, PyQt etc. )
* Web frameworks like Django (used by YouTube, Instagram, Dropbox)
* Image processing (like Opencv, Pillow)
* Web scraping (like Scrapy, BeautifulSoup, Selenium)
* Test frameworks
* Multimedia

### Advantages of Python :-

Let’s see how Python dominates over other languages.

### Extensive Libraries

Python downloads with an extensive library and it *contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more.* So, we don’t have to write the complete code for that manually.

### Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

### Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

### Improved Productivity

The language’s simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

### IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

### Simple and Easy

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

### Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

### Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

### Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you [**download**](https://data-flair.training/blogs/install-python-windows/)[**Python**](https://data-flair.training/blogs/install-python-windows/)for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

### Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

### Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

### Advantages of Python Over Other Languages

1. **Less Coding**

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

### Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 GitHub annual survey showed us that Python has overtaken Java in the most popular programming language category.

### Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine**](https://data-flair.training/blogs/machine-learning-tutorials-home/)[**learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

### Disadvantages of Python

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

### Speed Limitations

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

### Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

### Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

### Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

### Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

### History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

### What is Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of *building models of data*.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

### Categories Of Machine Leaning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

*Supervised learning* involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and *regression* tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

*Unsupervised learning* involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as *clustering* and *dimensionality reduction.* Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

### Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data- driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programming logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

### Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long

way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

### Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real- world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

### How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed,](http://blog.indeed.com/2019/03/14/best-jobs-2019/) Machine Learning Engineer Is The Best Job of 2019 with a *344%* growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

### How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

### Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

### Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

### Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!

Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

### Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/)! While there are other languages you can use for Machine

Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Keras](https://keras.io/), [TensorFlow,](https://www.tensorflow.org/) [Scikit-learn](https://scikit-learn.org/stable/), etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as [**Fork Python**](https://practice.geeksforgeeks.org/courses/fork-python)available Free on GeeksforGeeks.

### Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

### Terminologies of Machine Learning

* **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
* **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
* **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training –** The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

### Types of Machine Learning

* **Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
* **Unsupervised Learning –** This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
* **Semi-supervised Learning –** This involves using unlabeled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
* **Reinforcement Learning –** This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

### Advantages of Machine learning :-

1. **Easily identifies trends and patterns -**

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

### No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

### Continuous Improvement

As [**ML algorithms**](https://data-flair.training/blogs/machine-learning-algorithms/)gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

### Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

### Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

### Disadvantages of Machine Learning :-

1. **Data Acquisition**

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

### Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

### Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

[**Machine Learning**](https://en.wikipedia.org/wiki/Machine_learning)is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

### Python Development Steps : -

Guido Van Rossum published the first version of Python code at alt sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python

2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* + Print is now a function
  + Views and iterators instead of lists
  + The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
  + There is only one integer type left, i.e. int. long is int as well.
  + The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
  + Text Vs. Data Instead Of Unicode Vs. 8-bit

### Purpose :-

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

### Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* + Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
  + Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and

several of them have later been patched and updated by people with no Python background - without breaking.

### Modules Used in Project :- Tensorflow

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and](https://en.wikipedia.org/wiki/Library_(computing)) [differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google.](https://en.wikipedia.org/wiki/Google)

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

### NumPy

NumPy is a general-purpose array-processing package. It provides a high- performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi- dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

### Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

### Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail](https://matplotlib.org/gallery/index.html) [gallery.](https://matplotlib.org/gallery/index.html)

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

### Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has

a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

### Install Python Step-by-Step in Windows and Mac :

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

### How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices. Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version

3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet](https://myelearninghub.com/python-cheat-sheet/) [here.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

### Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [**https://www.python.org**](https://www.python.org/)

****

**Fig 5.1: Open Python website to download the python latest version**

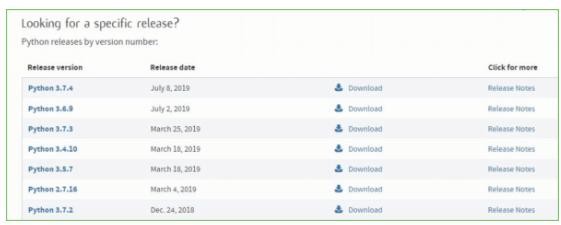
Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.



**Fig 5.2: Click on Download Python**

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



**Fig 5.3: Select the version what you want Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



**Fig 5.4: Python versions**

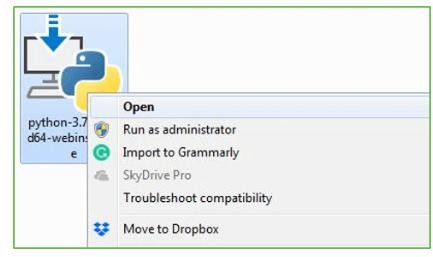
* To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
* To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Fig 5.5: Click on OK to install Python**

**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Fig 5.6: Click on Install Now to install**

**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



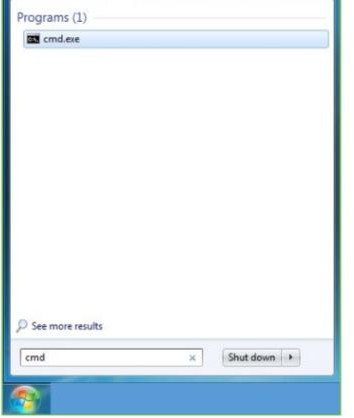
**Fig 5.7: Successful Installation of Python**

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

**Verify the Python Installation Step 1:** Click on Start

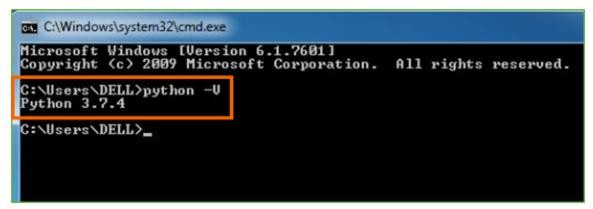
**Step 2:** In the Windows Run Command, type “cmd”.



**Fig 5.8: Open Command Prompt**

**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



**Fig 5.9: Successful installation check**

**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

### Check how the Python IDLE works

**Step 1:** Click on Start

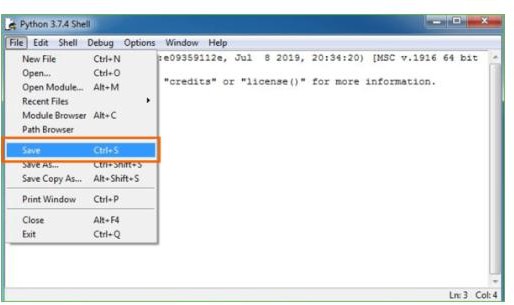
**Step 2:** In the Windows Run command, type “python idle”.



**Fig 5.10: Click on IDLE (Python 3.7 64-bit) and launch the program**

**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Fig 5.11: Saving the file**

**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter prin****t**

# CHAPTER 6

# CONCLUSION AND FUTUREWORK

## CONCLUSION

We analyzed a hacking breach dataset from the points of view of the incidents inter-arrival time and the breach size, and showed that they both should be modeled by stochastic processes rather than distributions. The statistical models developed in this paper show satisfactory fitting and prediction accuracies. In particular, we propose using a copula-based approach to predict the joint probability that an incident with a certain magnitude of breach size will occur during a future period of time. Statistical tests show that the methodologies proposed in this paper are better than those which are presented in the literature, because the latter ignored both the temporal correlations and the dependence between the incidents inter-arrival times and the breach sizes. We conducted qualitative and quantitative analyses to draw further insights. We drew a set of cybersecurity insights, including that the threat of cyber hacking breach incidents is indeed getting worse in terms of their frequency, but not the magnitude of their damage. The methodology presented in this paper can be adopted or adapted to analyze datasets of a similar nature.

## FUTUREWORK:

There are many open problems that are left for future research. For example, it is both interesting and challenging to investigate how to predict the extremely large values and how to deal with missing data (i.e., breach incidents that are not reported). It is also worthwhile to estimate the exact occurring times of breach incidents. Finally, more research needs to be conducted towards understanding the predictability of breach incidents (i.e., the upper bound of prediction accuracy).

# CHAPTER 7

# REFERENCE

1. K. K. Bagchi and G. Udo, “An analysis of the growth of computer and Internet security breaches,” Commun. Assoc. Inf. Syst., vol. 12, no. 1, p. 46, 2003.
2. R. Böhme and G. Kataria, “Models and measures for correlation in cyber- insurance,” in Proc. Workshop Econ. Inf. Secur. (WEIS), 2006, pp. 1–26.
3. E. Condon, A. He, and M. Cukier, “Analysis of computer security incident data using time series models,” in Proc. 19th Int. Symp. Softw. Rel. Eng. (ISSRE), Nov. 2008, pp. 77–86.
4. T. Maillart and D. Sornette, “Heavy-tailed distribution of cyber-risks,” Eur. Phys.

J. B, vol. 75, no. 3, pp. 357–364, 2010.

1. H. Herath and T. Herath, “Copula-based actuarial model for pricing cyber-insurance policies,” Insurance Markets Companies: Anal. Actuarial Comput., vol. 2, no. 1, pp. 7– 20, 2011.
2. P. Embrechts, C. Klüppelberg, and T. Mikosch, Modelling Extremal Events: For Insurance and Finance, vol. 33. Berlin, Germany: Springer-Verlag, 2013.
3. A. Mukhopadhyay, S. Chatterjee, D. Saha, A. Mahanti, and S. K. Sadhukhan, “Cyber-risk decision models: To insure it or not?” Decision Support Syst., vol. 56, pp. 11–26, Dec. 2013.
4. M. Eling and W. Schnell, “What do we know about cyber risk and cyber risk insurance?” J. Risk Finance, vol. 17, no. 5, pp. 474–491, 2016.
5. B. Edwards, S. Hofmeyr, and S. Forrest, “Hype and heavy tails: A closer look at data breaches,” J. Cybersecur., vol. 2, no. 1, pp. 3–14, 2016.
6. S. Wheatley, T. Maillart, and D. Sornette, “The extreme risk of personal data breaches and the erosion of privacy,” Eur. Phys. J. B, vol. 89, no. 1, p. 7, 2016.
7. C. R. Center. Cybersecurity Incidents. Accessed: Nov. 2017. [Online]. Available: <https://www.opm.gov/cybersecurity/cybersecurity-incidents>
8. ITR Center. Data Breaches Increase 40 Percent in 2016, Finds New Report From Identity Theft Resource Center and CyberScout. Accessed: Nov. 2017. [Online].

Available: <http://www.idtheftcenter.org/> 2016databreaches.html

1. P. R. Clearinghouse. Privacy Rights Clearinghouse’s Chronology of Data Breaches. Accessed: Nov. 2017. [Online]. Available: <https://www.privacyrights.org/data-breaches>
2. IBM Security. Accessed: Nov. 2017. [Online]. Available: <https://www.ibm.com/security/data-breach/index.html>
3. NetDiligence. The 2016 Cyber Claims Study. Accessed: Nov. 2017. [Online]. Available: https://netdiligence.com/wp-content/uploads/2016/ 10/P02\_NetDiligence- 2016-Cyber-Claims-Study-ONLINE.pdf
4. R. B. Security. Datalossdb. Accessed: Nov. 2017. [Online]. Available: [https://blog.datalossdb.org](https://blog.datalossdb.org/)
5. M. Xu and L. Hua. (2017). Cybersecurity Insurance: Modeling and Pricing. [Online]. Available: https:/[/www.soa.org/research-reports/](http://www.soa.org/research-reports/) 2017/cybersecurity- insurance
6. M. Eling and N. Loperfido, “Data breaches: Goodness of fit, pricing, and risk measurement,” Insurance, Math. Econ., vol. 75, pp. 126–136, Jul. 2017.
7. M. Xu, L. Hua, and S. Xu, “A vine copula model for predicting the effectiveness of cyber defense early-warning,” Technometrics, vol. 59, no. 4, pp. 508–520, 2017
8. C. Peng, M. Xu, S. Xu, and T. Hu, “Modeling multivariate cybersecurity risks,”

J. Appl. Stat., pp. 1–23, 2018.