Anomaly Detection of Industrial Control Systems Based on Transfer Learning

A Project Report submitted in partial fulfillment of the requirements for the awardof the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Koka Aneesh (221810311013)

Loka Akash Reddy (221810311052)

Ponugoti Vivekananda Reddy (221810311022)

Dronavalli Mourya Sai (221810311048)

Under the esteemed guidance of

Mr. Kondamuri Hanumantha Rao

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GITAM

(Deemed to be University)

HYDERABAD

APRIL 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

DECLARATION

We, hereby declare that the project report entitled "**Anomaly Detection of Industrial Control Systems Based on Transfer Learning**" is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date: 11/04/2022

Registration No(s).	**Name(s)**	**Signature(s)**
221810311013	**Koka Aneesh**	
221810311052	**Loka Akash Reddy**	
221810311022	**Ponugoti Vivekananda Reddy**	
221810311048	**Dronavalli Mourya Sai**	

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

CERTIFICATE

This is to certify that the project report entitled "**Anomaly Detection of Industrial Control Systems Based on Transfer Learning**" is a bonafide record of work carried out by **Koka Aneesh(221810311013), Loka Akash Reddy(221810311052), Ponugoti Vivekananda Reddy(221810311022), Dronavalli Mourya Sai(221810311048)** students submitted in partial fulfillment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

Project Guide	**Head of the Department**
Mr.K.Hanumantha Rao	**Dr.S.Phani Kumar**
Assistant Professor	**Professor**

ACKNOWLEDGMENT

Our project would not have been successful without the help of several people. We would like to thank the personalities who were part of our project in numerous ways, those who gave us outstanding support from the birth of the project.

We are extremely thankful to our honorable Pro-Vice Chancellor, Prof. **N. Siva Prasad** for providing necessary infrastructure and resources for the accomplishment of our project.

We are highly indebted to **Prof. N. Seetharamaiah**, Principal, School of Technology, for his support during the tenure of the project.

We are very much obliged to our beloved **Prof. Phani Kumar**, Head of the Department of Computer Science & Engineering for providing the opportunity to undertake this project and encouragement in completion of this project.

We hereby wish to express our deep sense of gratitude to **Mr. K. Hanumantha Rao, Asst. Professor**, Department of Computer Science and Engineering, School of Technology for the esteemed guidance, moral support and invaluable advice provided by him for the success of the project.

We are also thankful to all the staff members of Computer Science and Engineering department who have co-operated in making our project a success. We would like to thank all our parents and friends who extended their help, encouragement and moral support either directly or indirectly in our project work.

Sincerely,

Koka Aneesh (221810311013)

Loka Akash Reddy (221810311052)

Ponugoti Vivekananda Reddy (221810311022)

Dronavalli Mourya Sai (221810311048)

ABSTRACT

Industrial Control Systems (ICSs) are the lifeline of a country. Therefore, the anomaly detection of ICS traffic is an important endeavor. This paper proposes a model based on a deep residual Convolution Neural Network (CNN) to prevent gradient explosion or gradient disappearance and guarantee accuracy. The developed methodology addresses two limitations: most traditional machine learning methods can only detect known network attacks and deep learning algorithms require a long time to train. The utilization of transfer learning under the modification of the existing residual CNN structure guarantees the detection of unknown attacks. One-dimensional ICS flow data are converted into two-dimensional grayscale images to take full advantage of the features of CNN. Results show that the proposed method achieves a high score and solves the time problem associated with deep learning model training. The model can give reliable predictions for unknown or differently distributed abnormal data through short-term training. Thus, the proposed model ensures the safety of ICSs and verifies the feasibility of transfer learning for ICS anomaly detection.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 MOTIVATION

Modern Industrial Control Systems (ICSs) have higher production efficiency than traditional industrial systems and can well process big data. However, increases in the type and frequency of network attacks and hacking incidents threaten the security of ICSs based on data transmission. The National Institute of Standards and Technology has proposed the main sources of security issues for modern ICSs, which include nonsecure communication protocols, poor network isolation and access controls, and the lack of an ICS anomaly detection system. Intrusion detection technology is an important research direction in the field of network security. The original flows of network equipment and servers have been comprehensively analyzed. When industrial control networks are invaded or traffic data are abnormal, intrusion detection technology can effectively predict and take active defensive measures in a timely manner. Deep learning has shown great research significance in intrusion detection technology.

1.2 PROBLEM DEFINITION

Machine learning algorithms are unsuitable for detecting abnormal traffic in ICSs, and finding abnormal data quickly and implementing active measures with high accuracy are quite challenging. Compared with traditional machine learning, deep learning has a strong generalizability for extracting high dimensional data. Deep learning uses back-propagation algorithms to change and adjust parameters continuously to achieve optimal results.

1.3 OBJECTIVE OF PROJECT

The core aim of traditional machine learning is to map features to the target space. In traditional machine learning algorithms, the recognition rate increases with increasing data size; however, because a bottleneck period is often encountered during processing, these models cannot handle massive amounts of data. Machine learning performs well in intrusion detection in closed environments. However, machine learning will be exposed when entering an open-world scenario with various random traffic or noise, which could adversely affect its availability.

1.4 LIMITATIONS OF PROJECT

This learning method can handle large amounts of data; indeed, the larger the data size, the better the resulting effect. Unfortunately, although deep learning has good generalizability in processing images, it relies on labeled data and cannot handle unknown abnormal data types. In this article, we solve some of the problems of traditional machine learning by using a residual Convolution Neural Network (CNN) structure to model the source dataset and modify the relevant parameters by transfer learning. We then apply the transfer learning algorithm using the relevant information of the source domain and predicting the target domain. Transfer learning is finally employed to train the model quickly and detect differently distributed or unknown datasets.

1.5 STRUCTURE OF PROJECT (SYSTEM ANALYSIS)

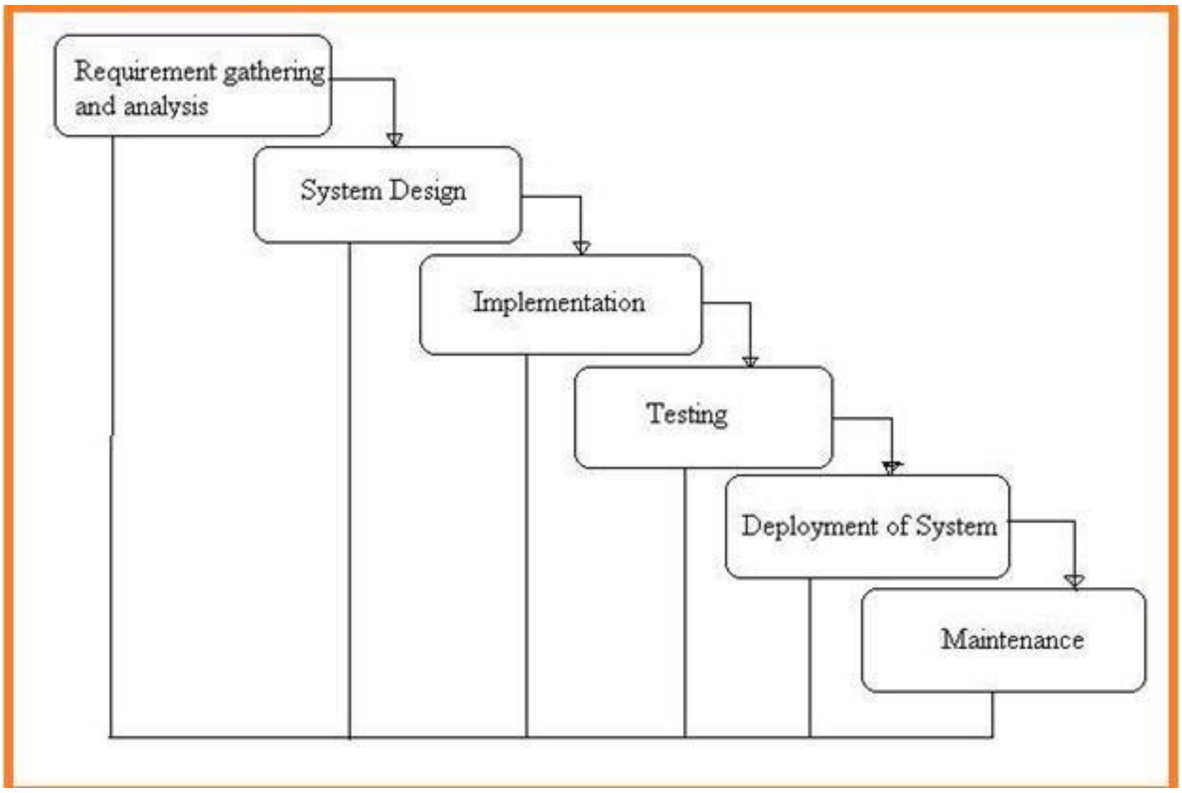


Fig: 1 Project SDLC

- Project Requisites Accumulating and Analysis
- Application System Design
- Practical Implementation
- Manual Testing of My Application
- Application Deployment of System
- Maintenance of the Project

1.5.1 REQUISITES ACCUMULATING AND ANALYSIS

It's the as a matter of first importance phase of the any task as our is a scholarly leave for essentials accumulating we followed of IEEE Journals and Amassed so numerous IEEE Relegated papers and last separated a Paper assigned "Singular web revisitation by setting and substance significance input and for investigation stage we took refs from the paper and did writing review of certain papers and amassed every one of the Requisites of the venture in this stage.

1.5.2 SYSTEM DESIGN

In System Design has separated into three sorts like GUI Designing, UML Designing with benefits being developed of venture in effortless manner with various entertainer and its utilizer case by utilizer case outline, stream of the task using arrangement, Class chart gives data about various class in the undertaking with techniques that must be used in the venture if goes to our task our UML Will utilizable in this manner The third and post import for the undertaking in framework configuration is Data base plan where we attempt to plan information base predicated on the quantity of modules in our task.

1.5.3 IMPLEMENTATION

The Implementation is Phase where we endeavor to give the practical output of the work done in designing stage and most of Coding in Business logic lay comes into action in this stage its main and crucial part of the project

1.6 TESTING

UNIT TESTING

It is done by the developer itself in every stage of the project and fine-tuning the bug and

module predicated additionally done by the developer only here we are going to solve all the

runtime errors.

MANUAL TESTING

As our Project is academic Leave, we can do any automatic testing so we follow manual

testing by endeavor and error methods

1.6.1 DEPLOYMENT OF SYSTEM AND MAINTENANCE

When the undertaking is absolute yare, we will come to sending of customer framework in

truly world as its scholastic leave we did organization I our school lab just with all need

Software's with having Windows OS.

The Maintenance of our Project is one-time process only

1.7 FUNCTIONAL REQUIREMENTS

1. Client Authentication
2. Exchange Keys
3. Authenticate Signature
4. Upload Data
5. Download Data

1.8 NON FUNCTIONAL REQUIREMENTS

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, “how fast does the website load?” Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.

- Usability requirement
- Serviceability requirement
- Manageability requirement

- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Scalability requirement
- Interoperability requirement
- Reliability requirement
- Maintainability requirement
- Regulatory requirement
- Environmental requirement

1.8.1 EXAMPLES OF NON-FUNCTIONAL REQUIREMENTS

Here, are some examples of non-functional requirement:

1.8.1.1 Users must upload dataset

1.8.1.2 The software should be portable. So moving from one OS to other OS does not create any problem.

1.8.1.3 Privacy of information, the export of restricted technologies, intellectual property rights, etc. should be audited.

1.8.2 ADVANTAGES OF NON-FUNCTIONAL REQUIREMENT

Benefits/pros of Non-functional testing are:

- The nonfunctional requirements ensure the software system follow legal and compliance rules.
- They ensure the reliability, availability, and performance of the software system.
- They ensure good user experience and ease of operating the software.
- They help in formulating security policy of the software system.

1.8.3 DISADVANTAGES OF NON-FUNCTIONAL REQUIREMENT

Cons/drawbacks of Non-function requirement are:

- None functional requirement may affect the various high-level software subsystem
- They require special consideration during the software architecture/high-level design phase which increases costs.
- Their implementation does not usually map to the specific software sub-system,
- It is tough to modify non-functional once you pass the architecture phase.

1.8.4 KEY LEARNING

Experimental results demonstrate that the accuracy of the proposed method can outperform most of the state-of-art methods.

2. LITERATURE SURVEY

2.1 SECURITY AND PRIVACY CHALLENGES IN INDUSTRIAL INTERNET OF THINGS

AUTHORS: A. R. Sadeghi, C. Wachsmann, and M. Waidner

ABSTRACT: Today, embedded, mobile, and cyberphysical systems are ubiquitous and used in many applications, from industrial control systems, modern vehicles, to critical infrastructure. Current trends and initiatives, such as “Industrie 4.0” and Internet of Things (IoT), promise innovative business models and novel user experiences through strong connectivity and effective use of next generation of embedded devices. These systems generate, process, and exchange vast amounts of security-critical and privacy-sensitive data, which makes them attractive targets of attacks. Cyberattacks on IoT systems are very critical since they may cause physical damage and even threaten human lives. The complexity of these systems and the potential impact of cyberattacks bring upon new threats. This paper gives an introduction to Industrial IoT systems, the related security and privacy challenges, and an outlook on possible solutions towards a holistic security framework for Industrial IoT systems.

2.2 CHALLENGES OF MACHINE LEARNING BASED MONITORING FOR INDUSTRIAL CONTROL SYSTEM NETWORKS

AUTHORS: M. Mantere, I. Uusitalo, M. Sailio, and S. Noponen

ABSTRACT: Detecting network intrusions and anomalies in industrial control systems is growing in urgency. Such systems used to be isolated but are now being connected to the outside world. Even in the case of isolated networks, privileged users may still present various threats to the system, either accidentally or intentionally. Also malfunctions in devices may cause anomalous traffic. Anomaly detection based network monitoring and intrusion detection systems could be capable of discerning normal and aberrant traffic in industrial control systems, detecting security incidents in an early phase. In this paper we discuss the challenges for such a monitoring system. One of the challenges is which features best differentiate between anomalous and normal behaviour. In the analysis, special focus is placed on this selection.

2.3 DEEP LEARNING ANDD ITS APPLICATIONS TO MACHINE HEALTH MONITORING: A SURVEY

AUTHORS: Rui Zhao, Ruqiang Yan, Zhenghua Chen, Kezhi Mao, Peng Wang, Robert X. Gao

ABSTRACT: Since 2006, deep learning (DL) has become a rapidly growing research direction, redefining state-of-the-art performances in a wide range of areas such as object recognition, image segmentation, speech recognition and machine translation. In modern manufacturing systems, data-driven machine health monitoring is gaining in popularity due to the widespread deployment of low-cost sensors and their connection to the Internet. Meanwhile, deep learning provides useful tools for processing and analyzing these big machinery data. The main purpose of this paper is to review and summarize the emerging research work of deep learning on machine health monitoring. After the brief introduction of deep learning techniques, the applications of deep learning in machine health monitoring systems are reviewed mainly from the following aspects: Auto-encoder (AE) and its variants, Restricted Boltzmann Machines and its variants including Deep Belief Network (DBN) and Deep Boltzmann Machines (DBM), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Finally, some new trends of DL-based machine health monitoring methods are discussed.

2.4 EXPLORING THE LIMITS OF TRANSFER LEARNING WITH A UNIFIED TEXT-TO-TEXT TRANSFORMER

AUTHORS: C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Q. Zhou, W. Li, and P. J. Liu

ABSTRACT: Transfer learning, where a model is first pre-trained on a data-rich task before being fine-tuned on a downstream task, has emerged as a powerful technique in natural language processing (NLP). The effectiveness of transfer learning has given rise to a diversity of approaches, methodology, and practice. In this paper, we explore the landscape of transfer learning techniques for NLP by introducing a unified framework that converts all text-based language problems into a text-to-text format. Our systematic study compares pre-training objectives, architectures, unlabeled data sets, transfer approaches, and other factors on dozens of language understanding tasks. By combining the insights from our exploration with scale and our new ``Colossal Clean Crawled Corpus", we achieve state-of-the-art results on many benchmarks covering summarization, question answering, text classification, and more. To facilitate future work on transfer learning for NLP, we release our data set, pre-trained models, and code.

2.5 INDUSTRIAL ANOMALY DETECTION AND ATTACK CLASSIFICATION METHOD BASED ON CONVOLUTIONAL NEURAL NETWORK

AUTHORS: Y. Lai, J. Zhang, and Z. liu

ABSTRACT: The massive use of information technology has brought certain security risks to the industrial production process. In recent years, cyber-physical attacks against industrial control systems have occurred frequently. Anomaly detection technology is an essential technical means to ensure the safety of industrial control systems. Considering the shortcomings of traditional methods and to facilitate the timely analysis and location of anomalies, this study proposes a solution based on the deep learning method for industrial traffic anomaly detection and attack classification. We use a convolutional neural network deep learning representation model as the detection model. The original one-dimensional data are mapped using the feature mapping method to make them suitable for model processing. The deep learning method can automatically extract critical features and achieve accurate attack classification. We performed a model evaluation using real network attack data from a supervisory control and data acquisition (SCADA) system. The experimental results showed that the proposed method met the anomaly detection and attack classification needs of a SCADA system. The proposed method also promotes the application of deep learning methods in industrial anomaly detection.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

Machine learning algorithms are unsuitable for detecting abnormal traffic in ICSs, and finding abnormal data quickly and implementing active measures with high accuracy are quite challenging. Compared with traditional machine learning, deep learning has a strong generalizability for extracting high dimensional data. Deep learning uses back-propagation algorithms to change and adjust parameters continuously to achieve optimal results.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM:

Finding abnormal data quickly and implementing active measures with high accuracy are quite challenging.

3.2 PROPOSED SYSTEM:

This paper proposes a model based on a deep residual Convolution Neural Network (CNN) to prevent gradient explosion or gradient disappearance and guarantee accuracy. The developed methodology addresses two limitations: most traditional machine learning methods can only detect known network attacks and deep learning algorithms require a long time to train. The utilization of transfer learning under the modification of the existing residual CNN structure guarantees the detection of unknown attacks. One-dimensional ICS flow data are converted into two-dimensional grayscale images to take full advantage of the features of CNN. Results show that the proposed method achieves a high score and solves the time problem associated with deep learning model training. The model can give reliable predictions for unknown or differently distributed abnormal data through short-term training. Thus, the proposed model ensures the safety of ICSs and verifies the feasibility of transfer learning for ICS anomaly detection.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM:

The model can give reliable predictions for unknown or differently distributed abnormal data through short-term training.

3.3 SYSTEM REQUIREMENTS:

3.3.1 SOFTWARE REQUIREMENTS

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- **Python idel 3.7 version (or)**
- **Anaconda 3.7 (or)**
- **Jupiter (or)**
- **Google colab**

3.3.2 HARDWARE REQUIREMENTS

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

• **Operating system**	**: windows, linux**
• **Processor**	**: minimum intel i3**
• **Ram**	**: minimum 4gb**
• **Hard disk**	**: minimum 250gb**

3.4 FUNCTIONAL REQUIREMENTS

1. Data Collection
2. Data Preprocessing
3. Training and Testing
4. Modiling
5. Predicting

3.5 NON FUNCTIONAL REQUIREMENTS

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, “how fast does the website load?” Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.

- Usability requirement

- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Scalability requirement
- Interoperability requirement
- Reliability requirement
- Maintainability requirement
- Regulatory requirement
- Environmental requirement

3.6 SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it.

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE:

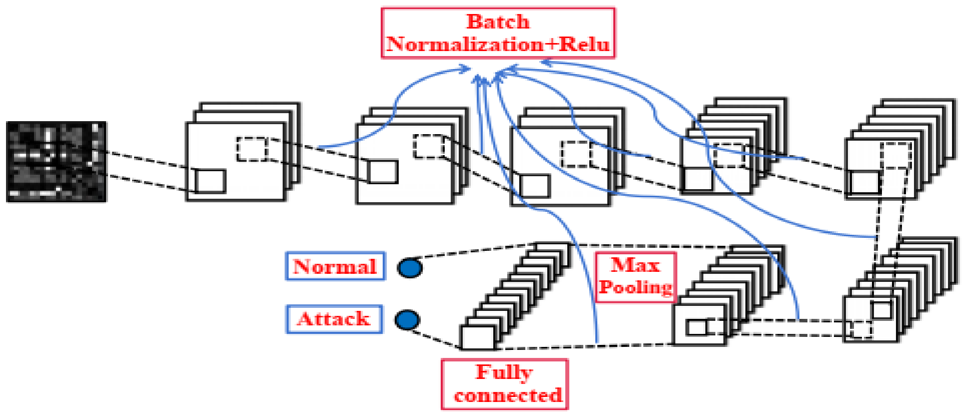


Fig: 2 System Architecture

FLOW CHART:

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.

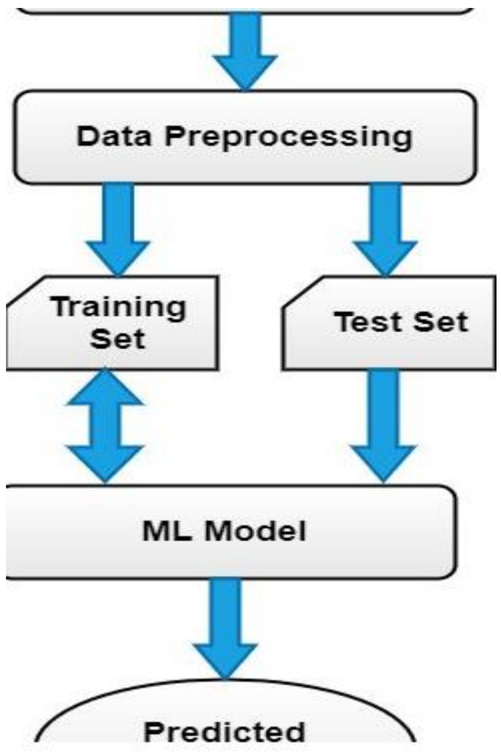


Fig: 3 Flow Chart

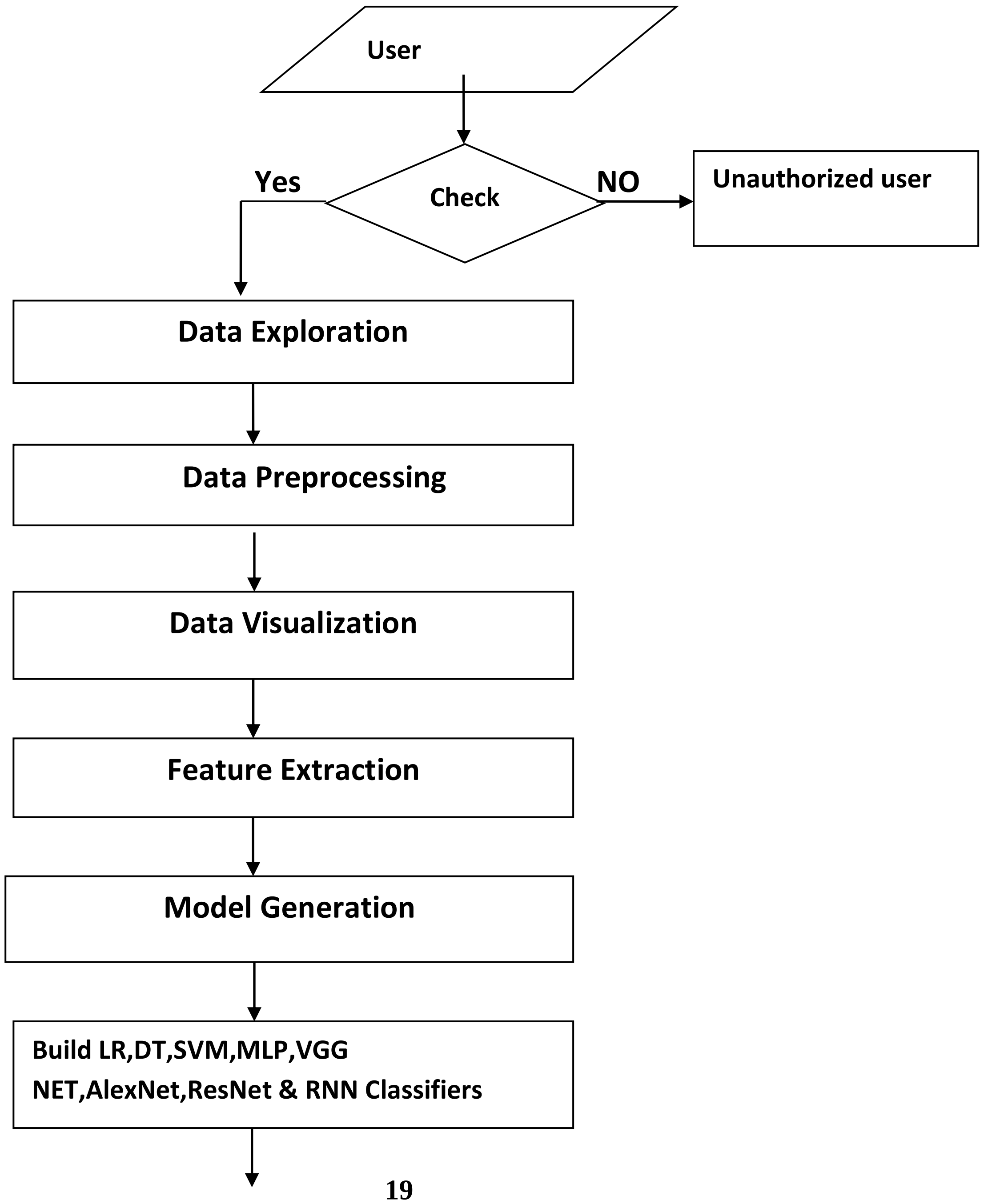
4.2 DATA FLOW DIAGRAM:

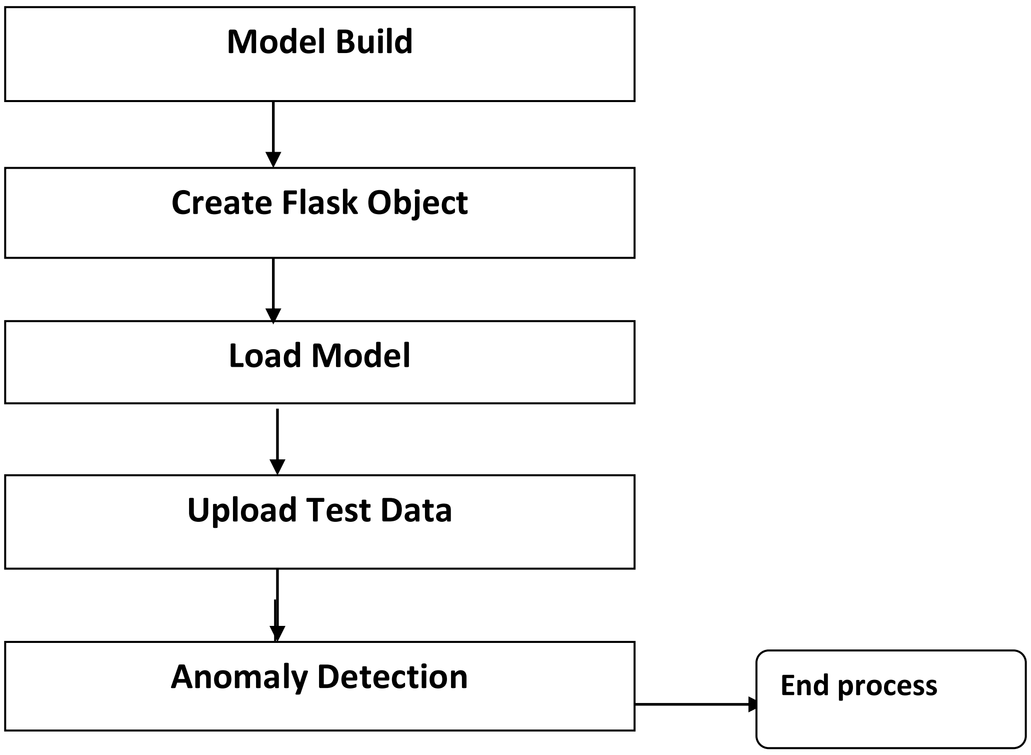
1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.





4.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.

4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

Use case diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

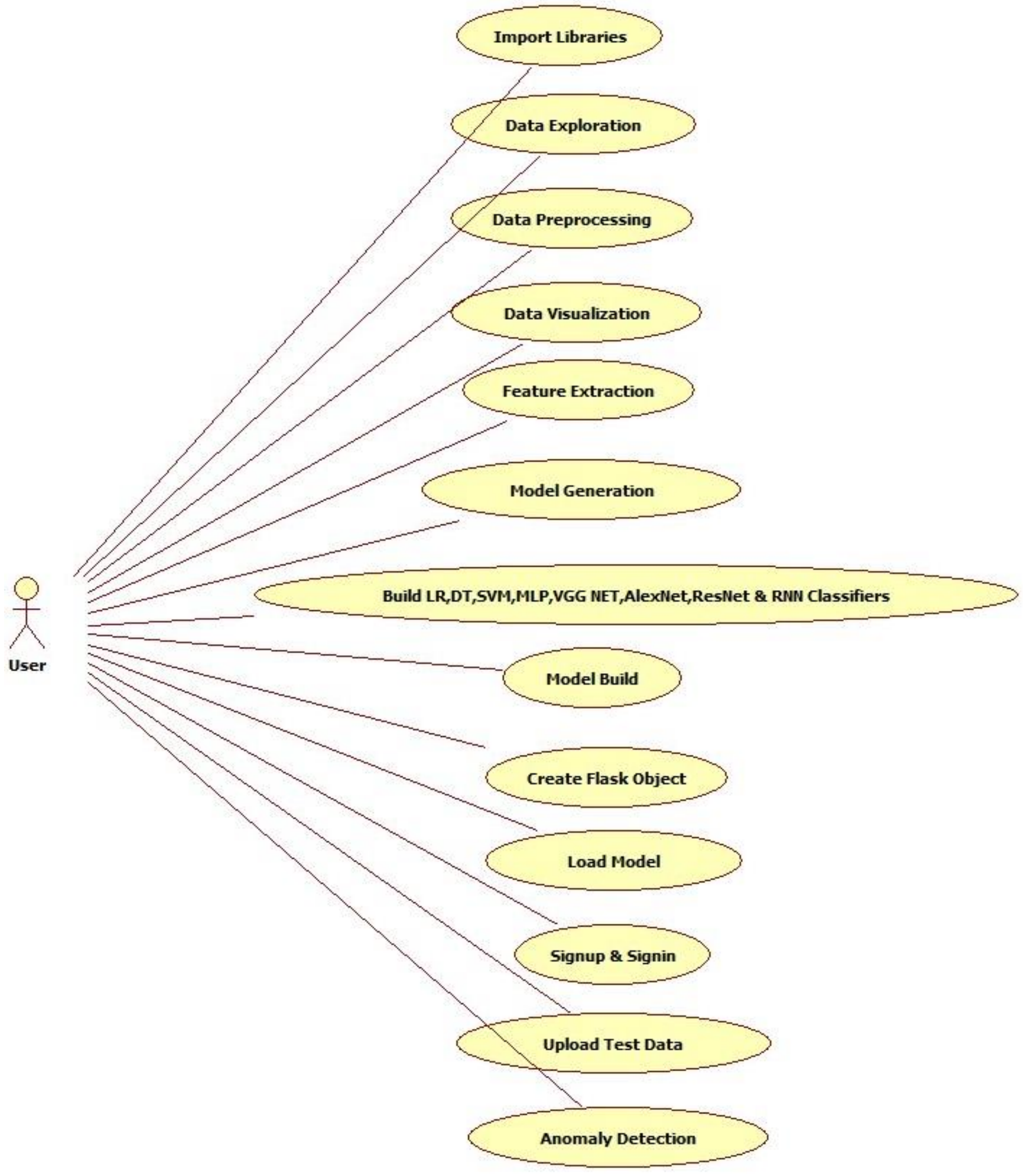
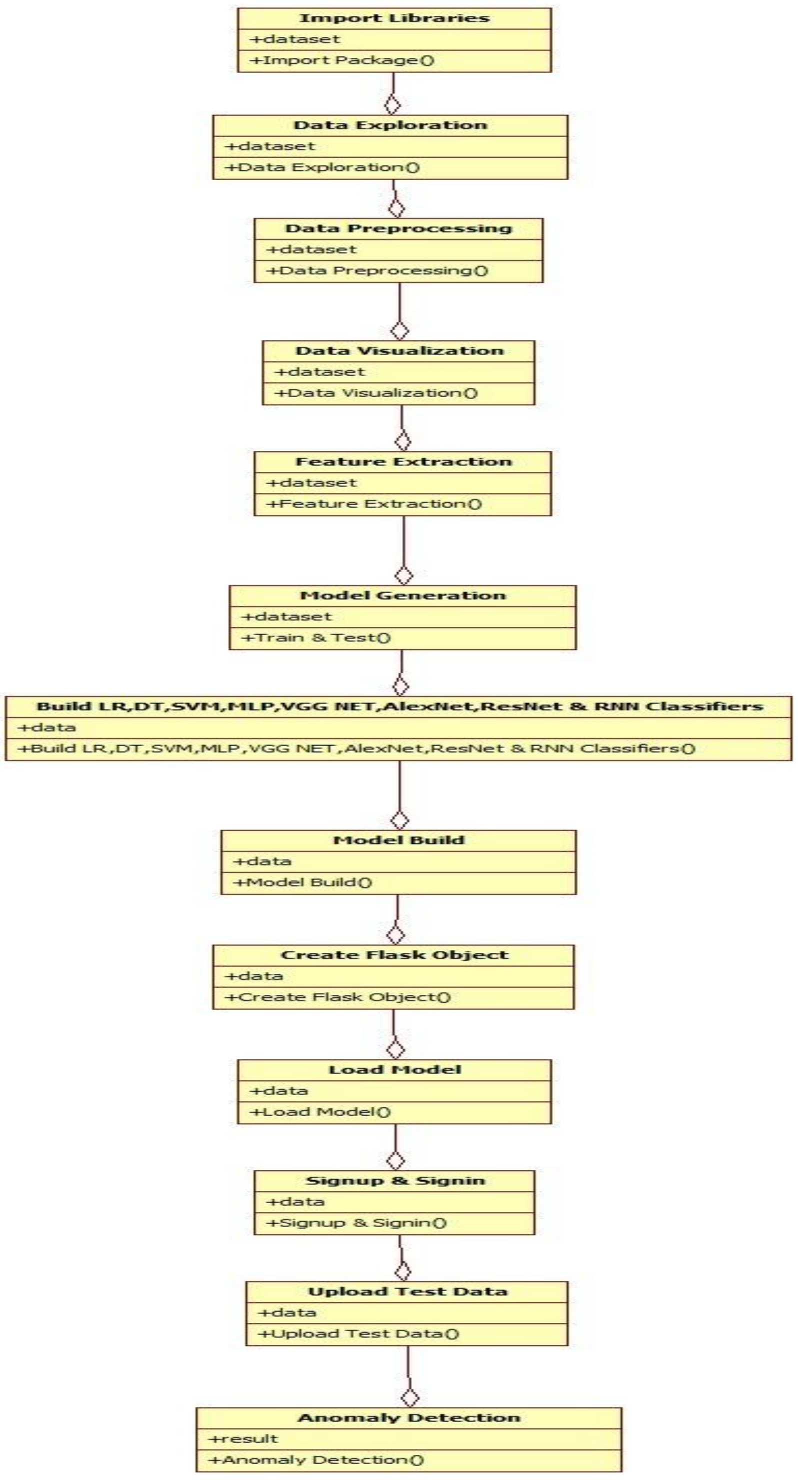


Fig: 4 Use Case Diagram

Class diagram:

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



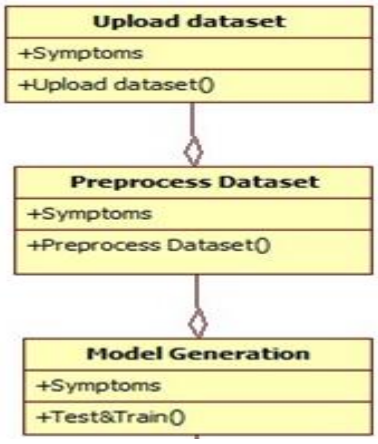


Fig: 5 Class Diagram

Object diagram: The object diagram is a special kind of class diagram. An object is an instance of a class. This essentially means that an object represents the state of a class at a given point of time while the system is running. The object diagram captures the state of different classes in the system and their relationships or associations at a given point of time.

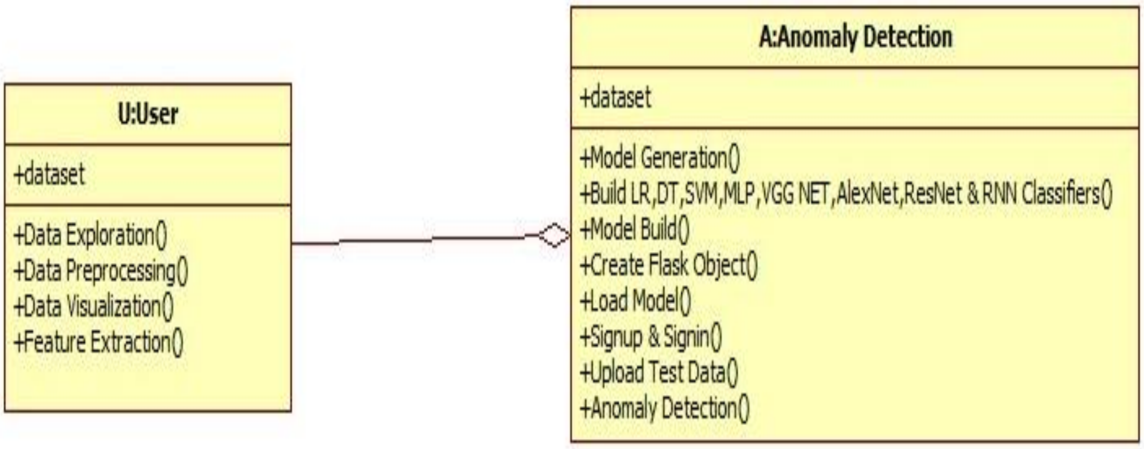


Fig: 6 Object Diagram

State diagram:

A state diagram, as the name suggests, represents the different states that objects in the system undergo during their life cycle. Objects in the system change states in response to events. In addition to this, a state diagram also captures the transition of the object's state from an initial state to a final state in response to events affecting the system.

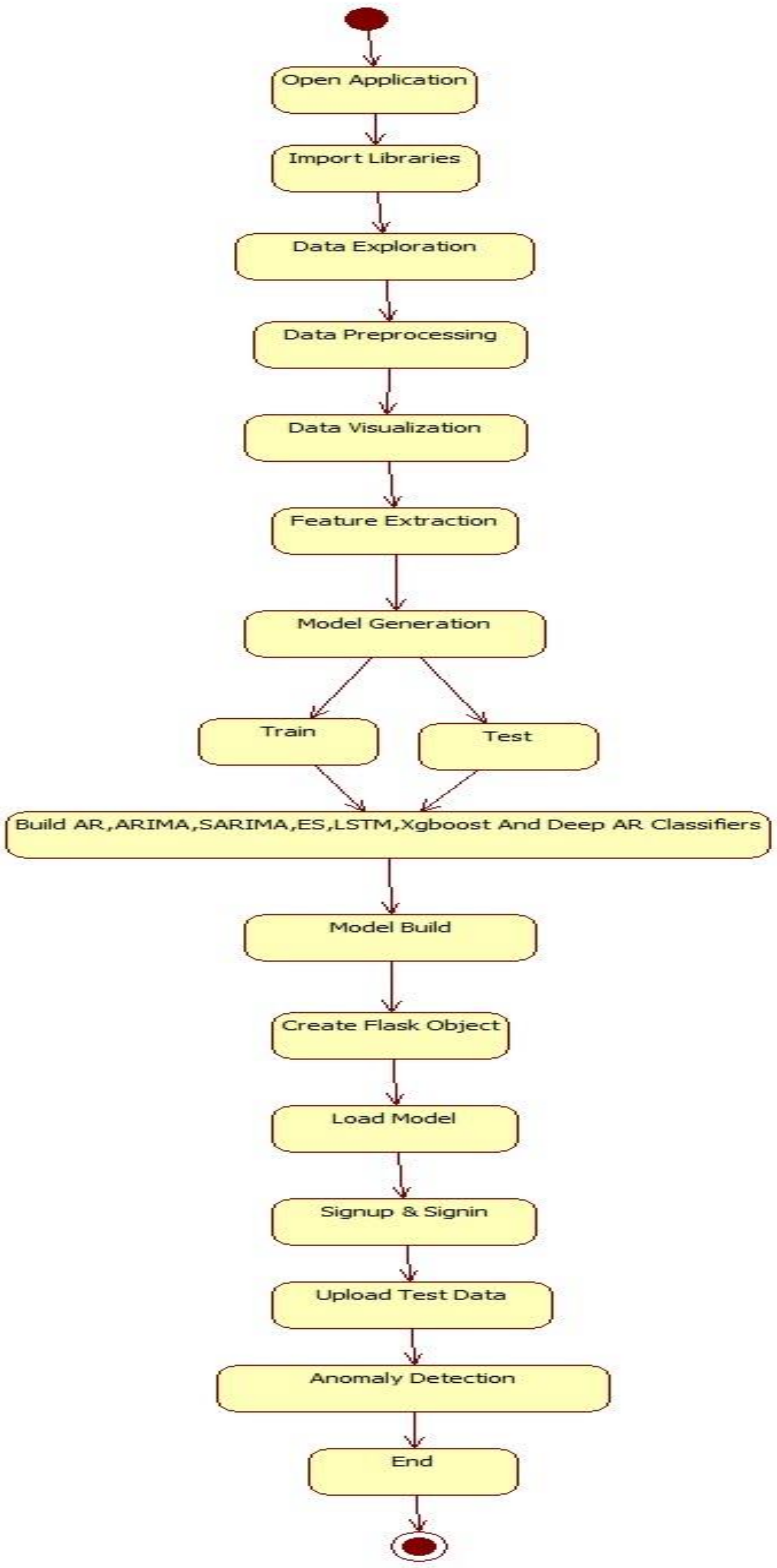


Fig: 7 State Diagram

Activity diagram:

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

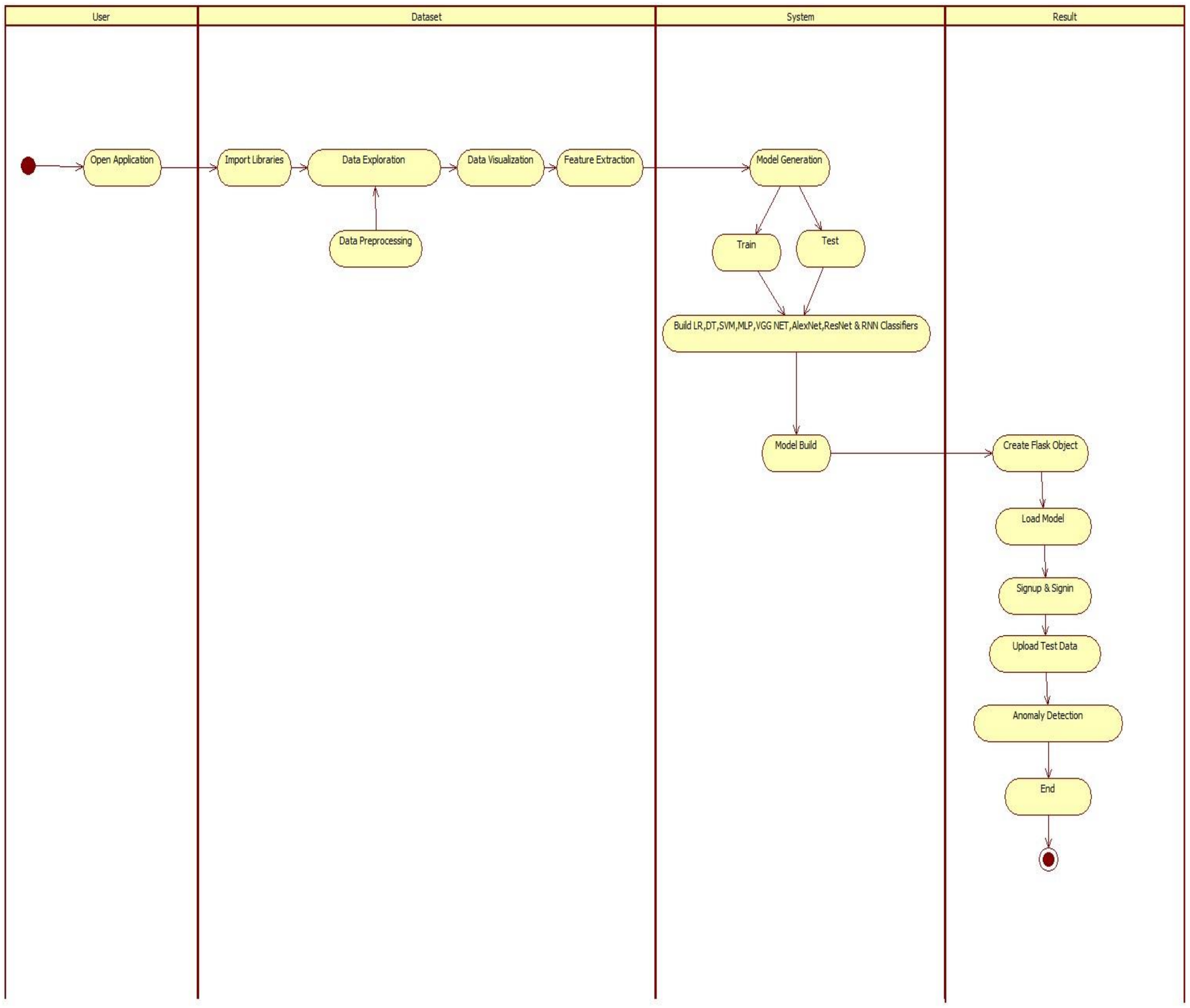


Fig: 8 Activity Diagram

Sequence diagram:

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

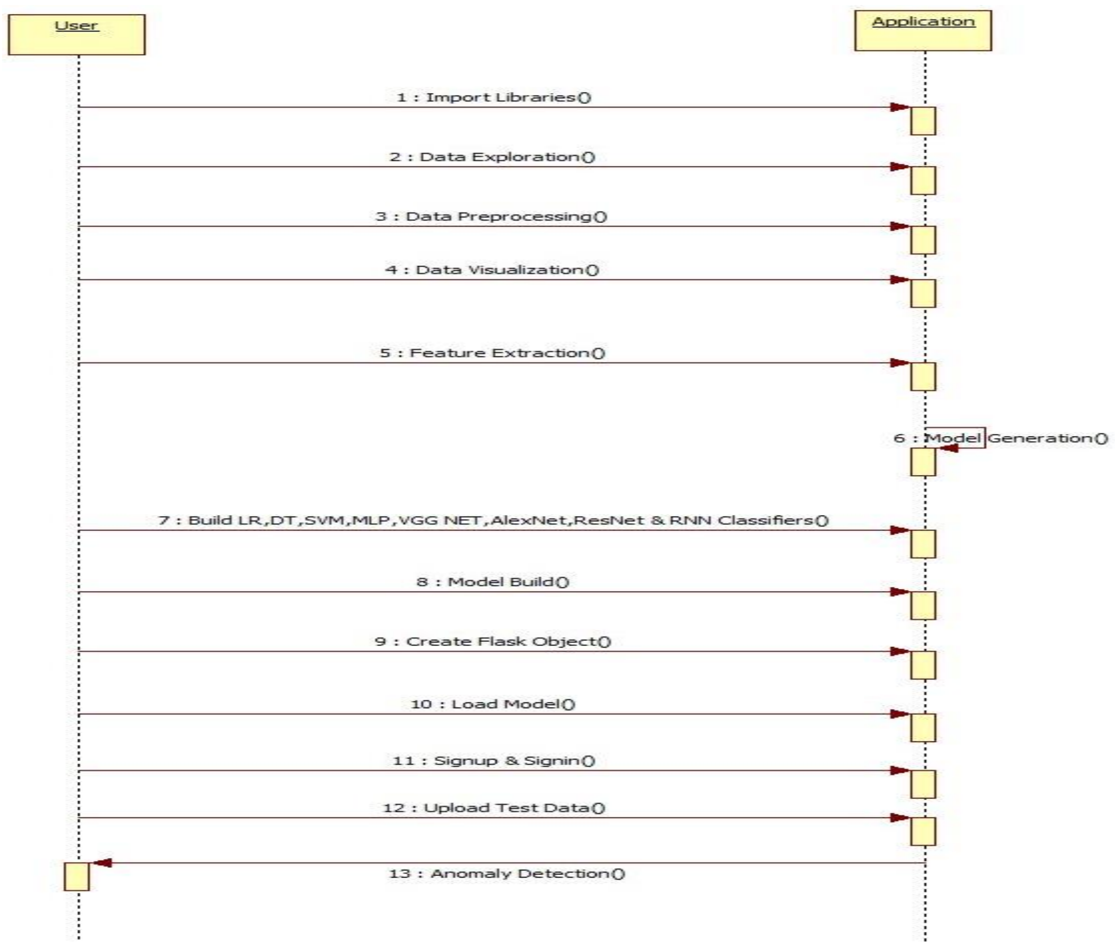


Fig: 9 Sequence Diagram

Collaboration diagram:

A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.

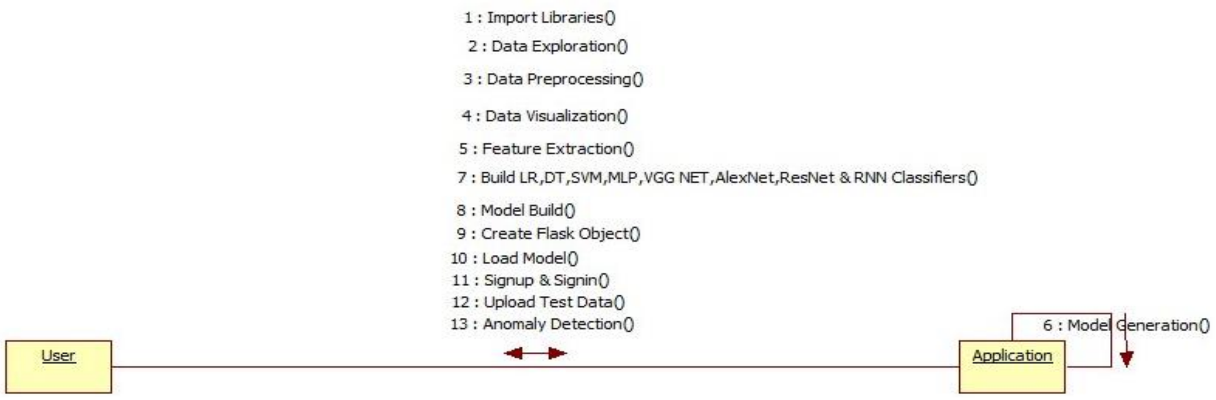


Fig:

Fig: 10 Collaboration Diagram

Component diagram:

The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.

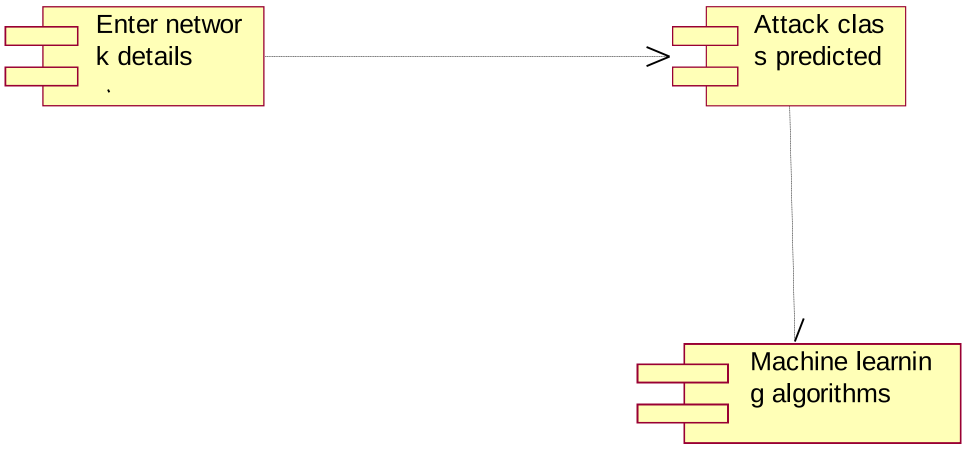


Fig: 11 Component Diagram

Deployment diagram:

The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.

User
System

Fig: 12 Deployment Diagram

4.4 IMPLEMENTATION:

MODULES:

1. DATA COLLECTION

The dataset used in human speech emotion. And the composition of the dataset understand the relationship among different features. A plot of the core features and the entire dataset. The dataset is further split into 2/3 for training and 1/3 for testing the algorithms. Furthermore, in order to obtain a representative sample, each class in the full dataset is represented in about the right proportion in both the training and testing datasets.

2. DATA PREPROCESSING

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency o the algorithm. The outliers have to be removed and also variable conversion need to be done. In order to overcoming these issues we use map function.

3. MODEL SELECTION

Machine learning is about predicting and recognizing patterns and generate suitable results after understanding them. ML algorithms study patterns in data and learn from them. An ML model will learn and improve on each attempt. To gauge the effectiveness of a model, it’s vital to split the data into training and test sets first. So before training our models, we split the data into Training set which was 70% of the whole dataset and Test set which was the remaining 30%.

4. PREDICT THE RESULTS

The designed system is tested with test set and the performance is assured. Evolution analysis refers to the description and model regularities or trends for objects whose behavior changes over time. Common metrics calculated from the confusion matrix are Precision; Accuracy. The mot important features since these features are to develop a predictive model using ordinary SVM and CNN model.

4.5 ALGORITHM:

SUPPORT VECTOR MACHINE (SVM):

Machine learning involves predicting and classifying data and to do so we employ various machine learning algorithms according to the dataset. SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyper plane which separates the data into classes. In machine learning, the radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms.

Intuitively, the further from the hyper plane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as far away from the hyper plane as possible, while still being on the correct side of it.

So when new testing data is added, whatever side of the hyperplane it lands will decide the class that we assign to it.

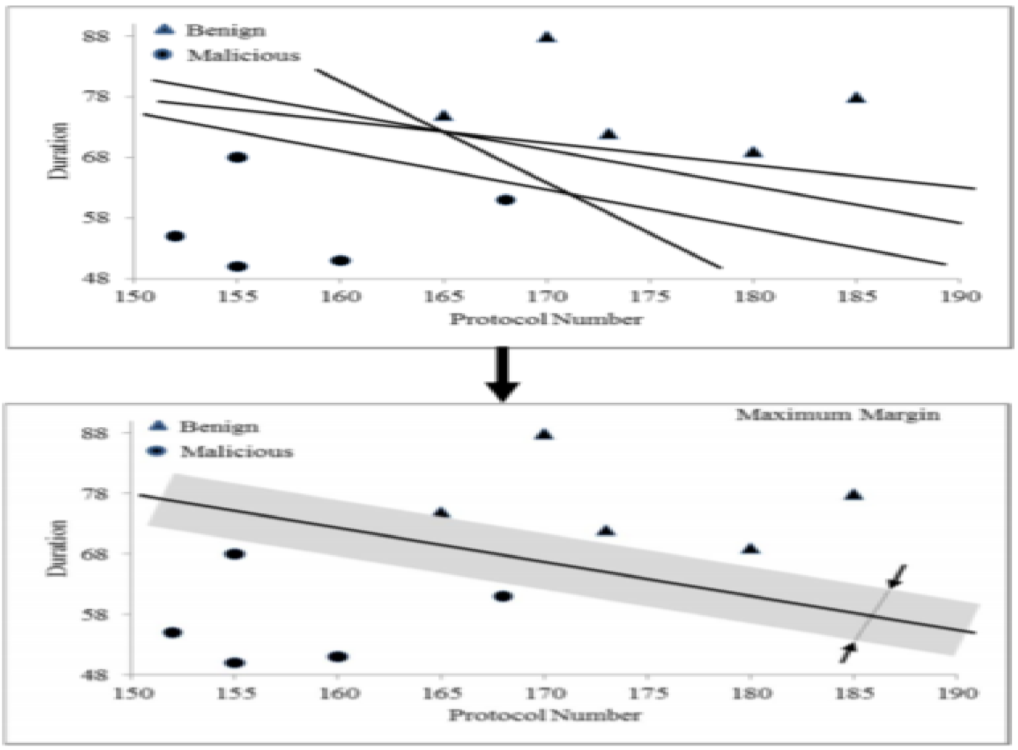


Fig: 13 Support Vector Machine

VGG16

VGG16 is a convolution neural net (CNN) architecture which was used to win ILSVR(Imagenet) competition in 2014. It is considered to be one of the excellent vision model architecture till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx) parameters. I am going to implement full VGG16 from scratch in Keras.

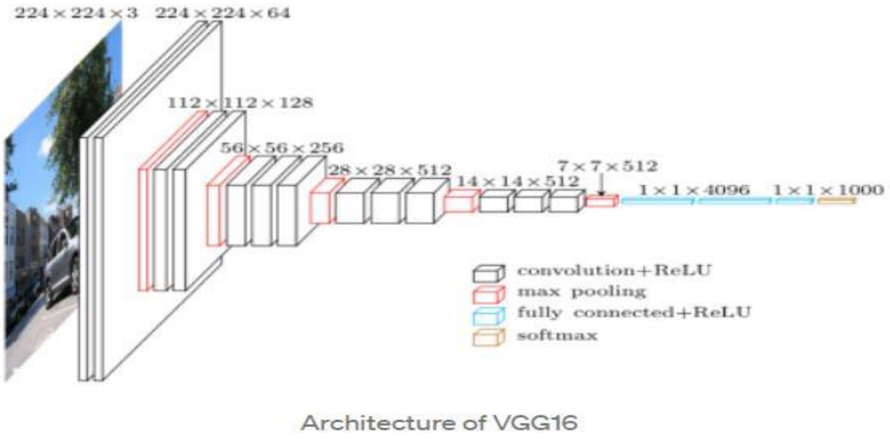


Fig: 14 Architecture of VGG16

RANDOM FOREST ALGORITHM

Random Forest algorithm is a supervised classification algorithm. We can see it from its name, which is to create a forest by some way and make it random. There is a direct relationship between the number of trees in the forest and the results it can get: the larger the number of trees, the more accurate the result. But one thing to note is that creating the forest is not the same as constructing the decision with information gain or gain index approach. The decision tree is a decision support tool. It uses a tree-like graph to show the possible consequences. If you input a training dataset with targets and features into the decision tree, it will formulate some set of rules. These rules can be used to perform predictions. When we have our dataset categorized into 3 category so now Random forest helps to make classes from the dataset. Random forest is clusters of decision trees all together, if you input a training dataset with features and labels into a decision tree.

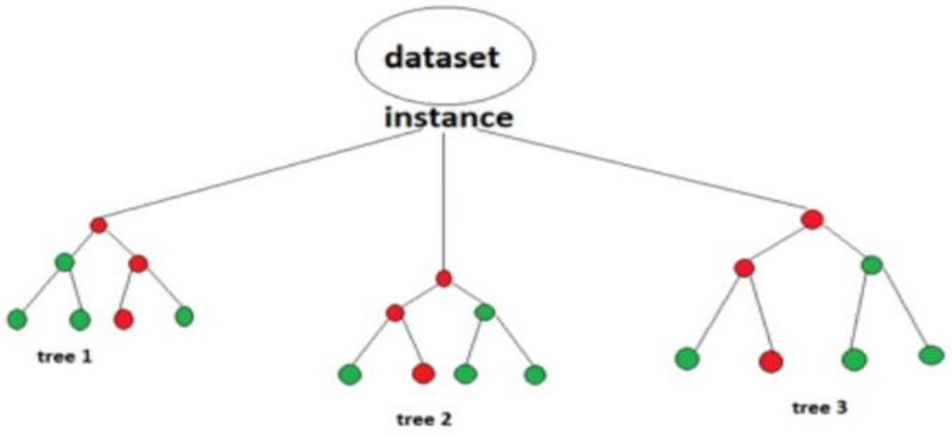


Fig: 15 Random Forest Algorithm

Decision Tree: A decision tree is a tree-like graph with nodes representing the place where we pick an attribute and ask a question; edges represent the answers to the question; and the leaves represent the actual output or class label. They are used in non-linear decision making with a simple linear decision surface.

Logistic Regression: The logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

MULTI-LAYER ANN

Deep Learning deals with training multi-layer artificial neural networks, also called Deep Neural Networks. After Rosenblatt perceptron was developed in the 1950s, there was a lack of interest in neural networks until 1986, when Dr.Hinton and his colleagues developed the backpropagation algorithm to train a multilayer neural network. Today it is a hot topic with many leading firms like Google, Facebook, and Microsoft which invest heavily in applications using deep neural networks.

Multi-layer ANN

A fully connected multi-layer neural network is called a Multilayer Perceptron (MLP).

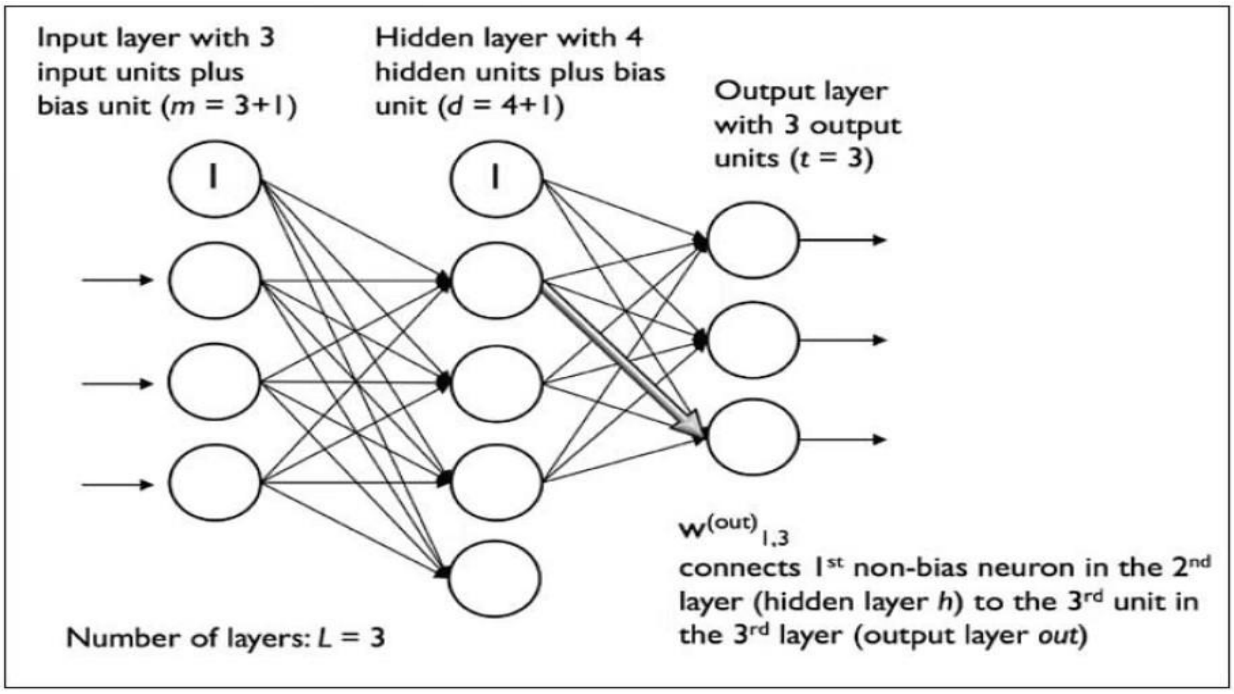

Input layer with 3 input units plus bias unit (m = 3+1)
Hidden layer with 4 hidden units plus bias unit (d = 4+1)
Output layer with 3 output units (t = 3)
1
1
w(out) 1,3 connects 1st non-bias neuron in the 2nd layer (hidden layer h) to the 3rd unit in the 3rd layer (output layer out)
Number of layers: L = 3

Fig: 16 Multi Layer Perceptron

It has 3 layers including one hidden layer. If it has more than 1 hidden layer, it is called a deep ANN. An MLP is a typical example of a feedforward artificial neural network. In this figure, the ith activation unit in the lth layer is denoted as ai(l).

The number of layers and the number of neurons are referred to as hyper parameters of a neural network, and these need tuning. Cross-validation techniques must be used to find ideal values for these.

The weight adjustment training is done via back propagation. Deeper neural networks are better at processing data. However, deeper layers can lead to vanishing gradient problems. Special algorithms are required to solve this issue.

RESNET:

A residual neural network (ResNet) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or shortcuts to jump over some layers. Typical ResNet models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between. An additional weight matrix may be used to learn the skip weights; these models are known as HighwayNets. Models with several parallel skips are referred to as DenseNets. In the context of residual neural networks, a non-residual network may be described as a plain network. A reconstruction of a pyramidal cell. Soma and dendrites are labeled in red, axon arbor in blue. (1) Soma, (2) Basal dendrite, (3) Apical dendrite, (4) Axon, (5) Collateral axon. There are two main reasons to add skip connections: to avoid the problem of vanishing gradients, or to mitigate the Degradation (accuracy saturation) problem; where adding more layers to a suitably deep model leads to higher training error. During training, the weights adapt to mute the upstream layer[clarification needed], and amplify the previously-skipped layer. In the simplest case, only the weights for the adjacent layer's connection are adapted, with no explicit weights for the upstream layer. This works best when a single nonlinear layer is stepped over, or when the intermediate layers are all linear. If not, then an explicit weight matrix should be learned for the skipped connection (a HighwayNet should be used). Skipping effectively simplifies the network, using fewer layers in the initial training stages[clarification needed]. This speeds learning by reducing the impact of vanishing gradients, as there are fewer layers to propagate through. The network then gradually restores the skipped layers as it learns the feature space. Towards the end of training, when all layers are expanded, it stays closer to the manifold[clarification needed] and thus learns faster.

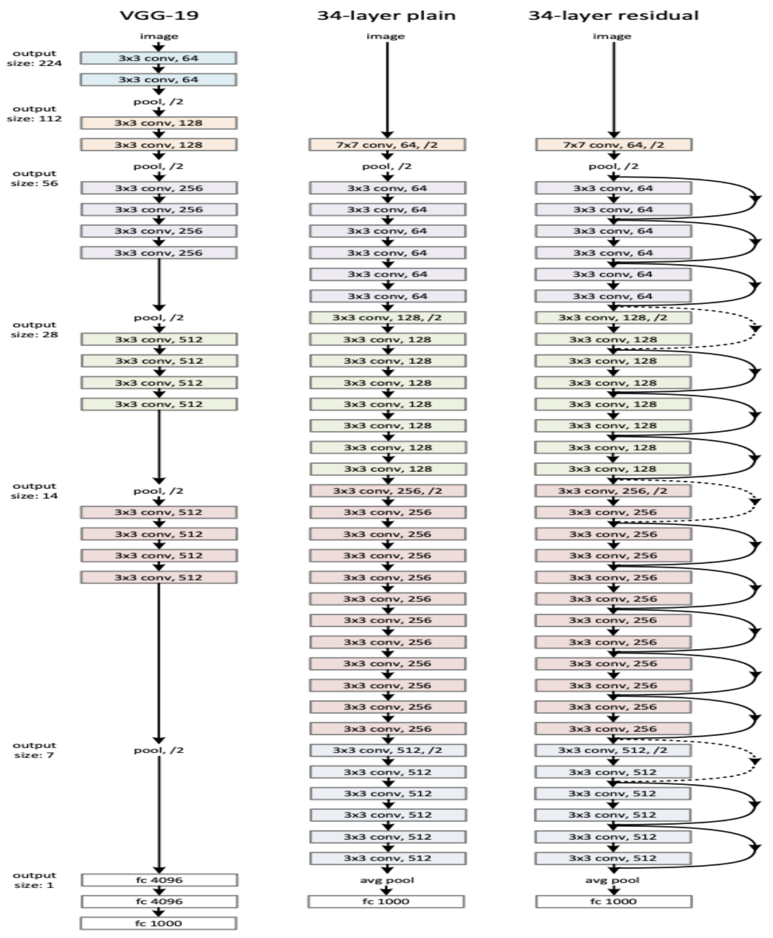


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

Fig: 17 ResNet

5. SOFTWARE ENVIRONMENT

5.1 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.

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

- 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.

1. 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.

2. 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.

3. 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.

4. 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.

5. Free and Open-Source:-

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it.

6. 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.

5.1.1 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.

2. 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.

3. 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 learning**, automate things, do web scraping and also build games and powerful visualizations.

5.1.2 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.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python 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.

2. 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.

3. 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**.

4. 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.

5. 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.

5.1.3 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

Venners, 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.

5.2 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.

5.2.1 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.

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.

5.2.2 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, 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!

(a) 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.

(b) 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.

(c) 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! While there are other languages you can use for Machine Learning like R, Scala, etc. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, 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** 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. 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:

(a) 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).

(b) 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 unlabelled 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.

5.2.3 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.

2. 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.

3. Continuous Improvement

As **ML 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.

4. 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.

5. 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.

5.2.4 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.

2. 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.

3. 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.

4. High error-susceptibility

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.

5.3 Python Development Steps: -

Guido Van Rossum published the first version of Python code (version 0.9.0) 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.

5.4 Modules Used in Project:-

Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 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. 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.

MatplotlibL

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 shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits.

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.

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.

How to Install Python on Windows and Mac :

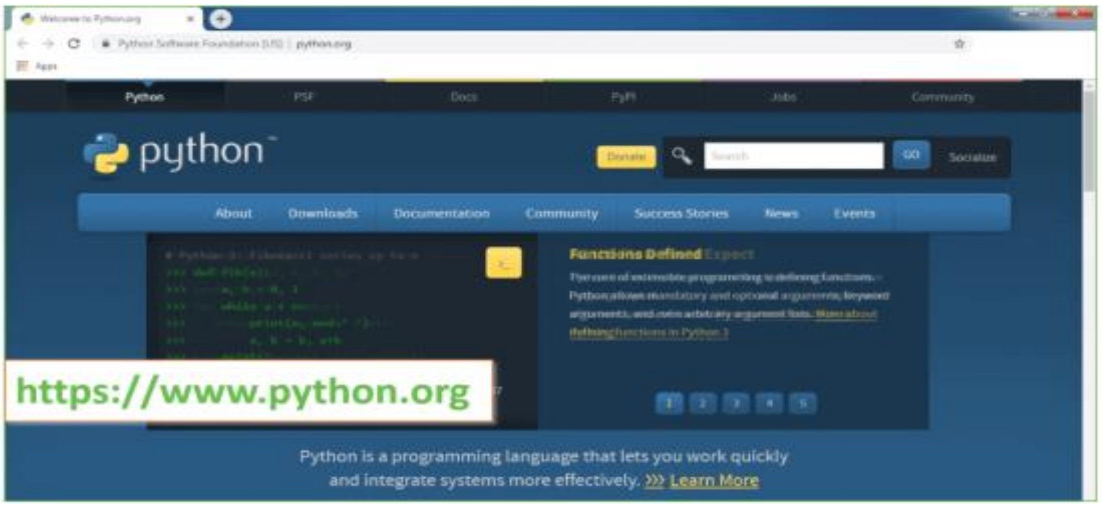
There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other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

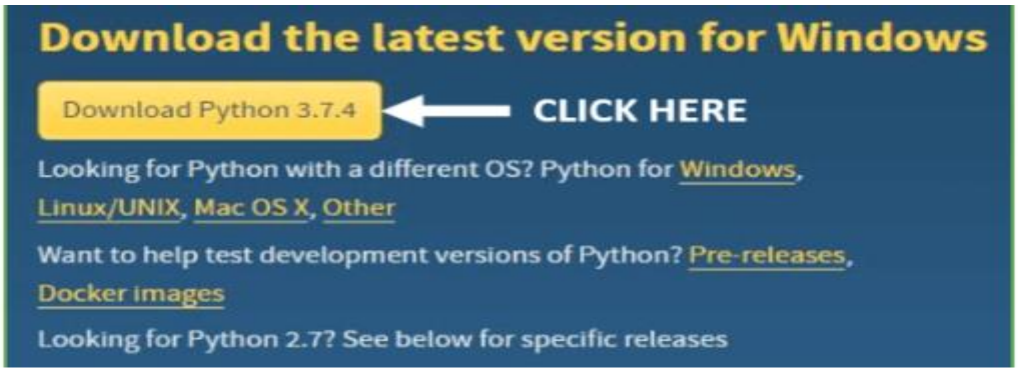
Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: **https://www.python.org**



Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

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

Step 2: Click on the Download Tab.

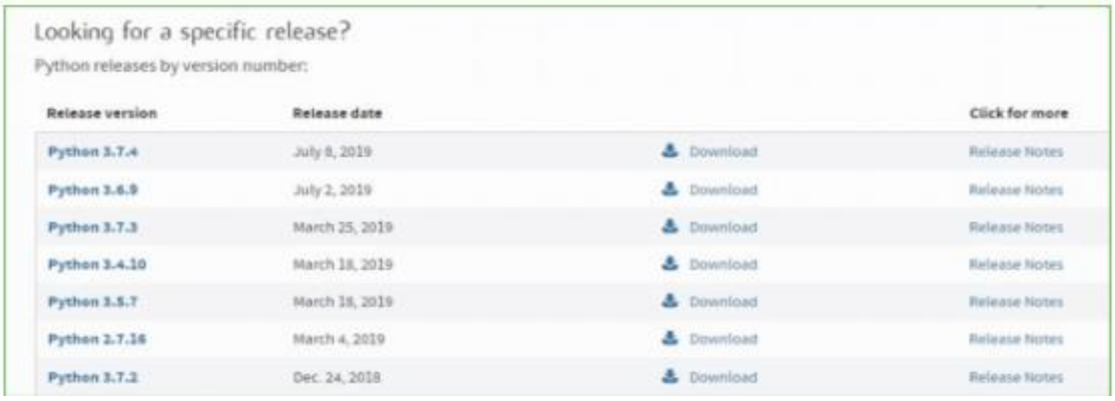

Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

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

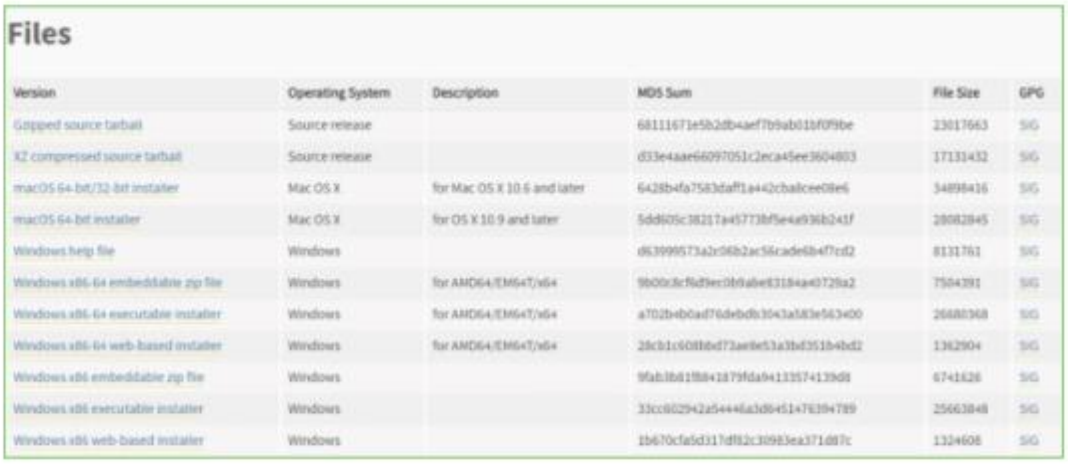

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

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.


Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

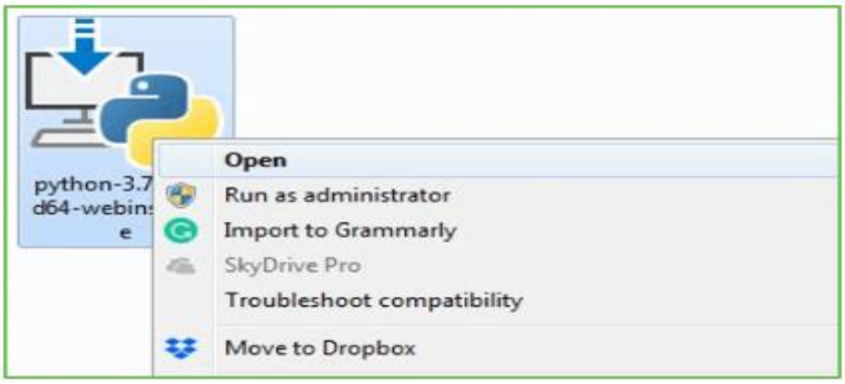
- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file,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

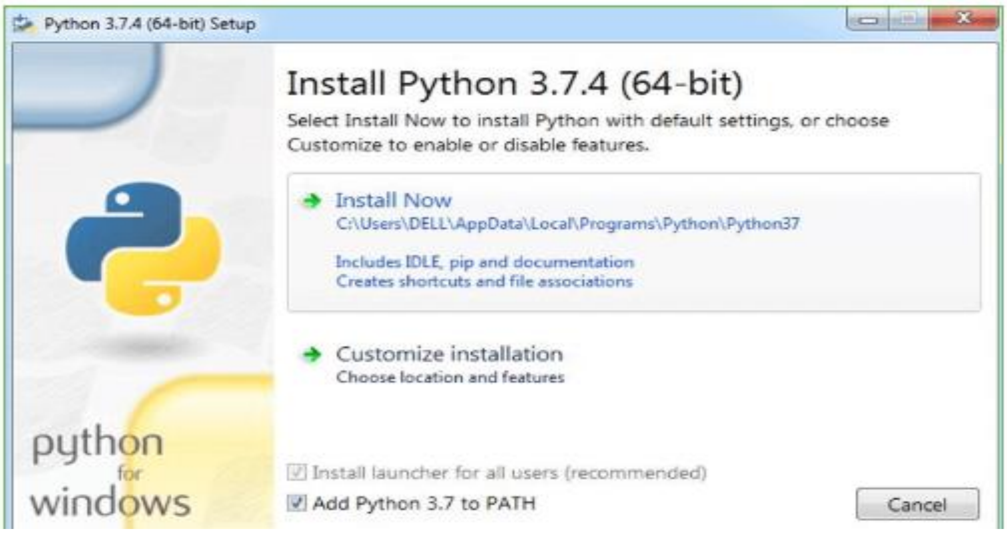
Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

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



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


Python 3.7.4 (64-bit) Setup

Install Python 3.7.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

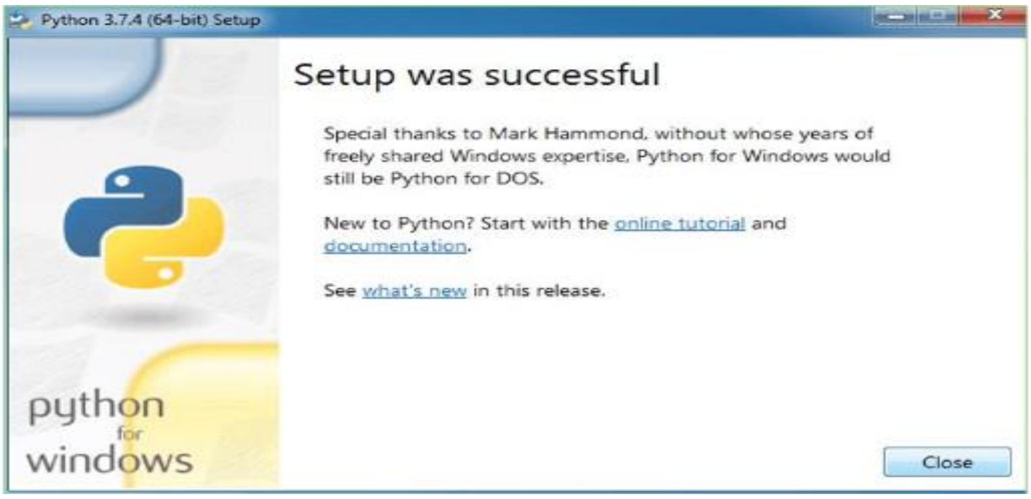
Install Now
C:\Users\DELL\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

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


Python 3.7.4 (64-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

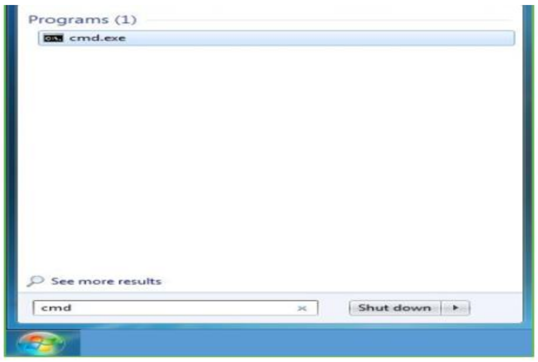
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

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

Verify the Python Installation

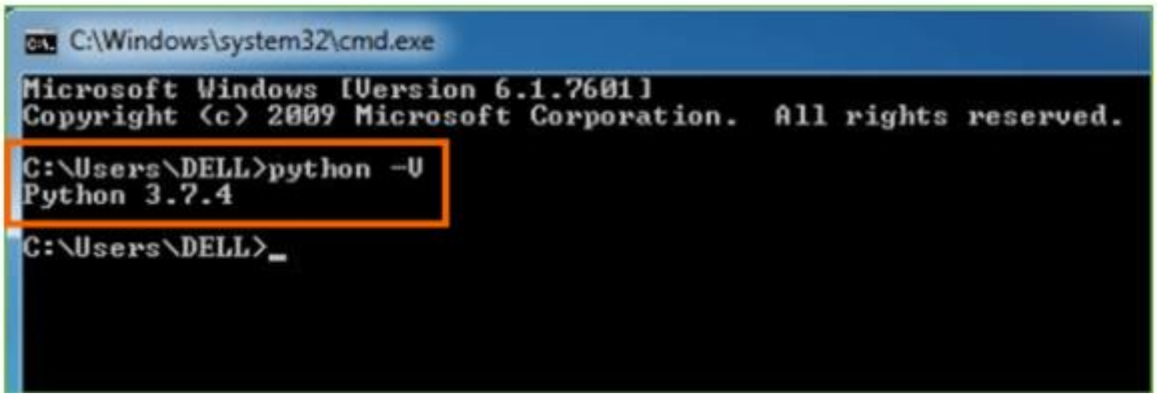
Step 1: Click on Start

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



Step 3: Open the Command prompt option.

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



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

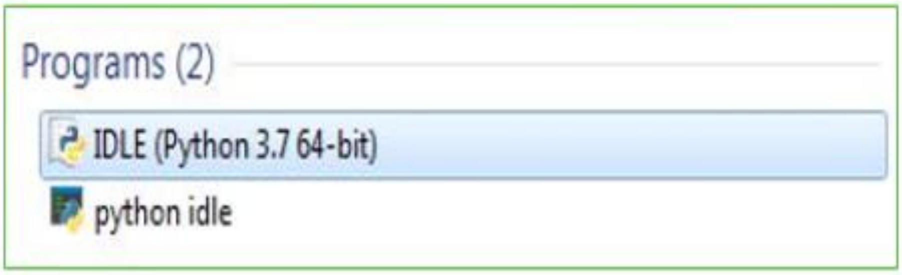
Step 5: You will get the answer as 3.7.4

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

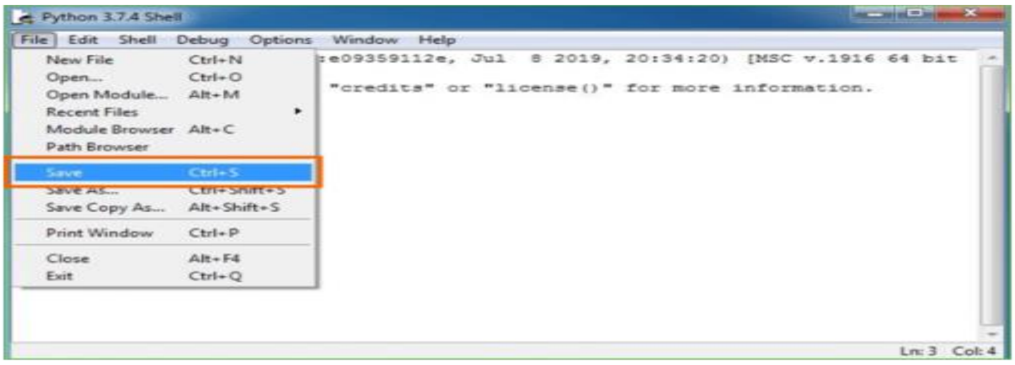
Step 1: Click on Start

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



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**



```
:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit
"credits" or "license()" for more information.
```

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 print.**

6. SYSTEM TESTING

6.1 TESTING STRATEGIES

6.1.1 Unit Testing

Unit testing, a testing technique using which individual modules are tested to determine if there are issues by the developer himself.. it is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

Unit Testing Techniques:

Black Box Testing - Using which the user interface, input and output are tested.

White Box Testing –Used to test each one of those functions behavior is tested.

6.1.2 Data Flow Testing

Data flow testing is a family of testing strategies based on selecting paths through the program’s control flow in order to explore sequence of events related to the status of Variables or data object. Dataflow Testing focuses on the points at which variables receive and the points at which these values are used.

6.1.3 Integration Testing

Integration Testing done upon completion of unit testing, the units or modules are to be integrated which gives raise too integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

6.1.4 Big Bang Integration Testing

Big Bang Integration Testing is an integration testing Strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

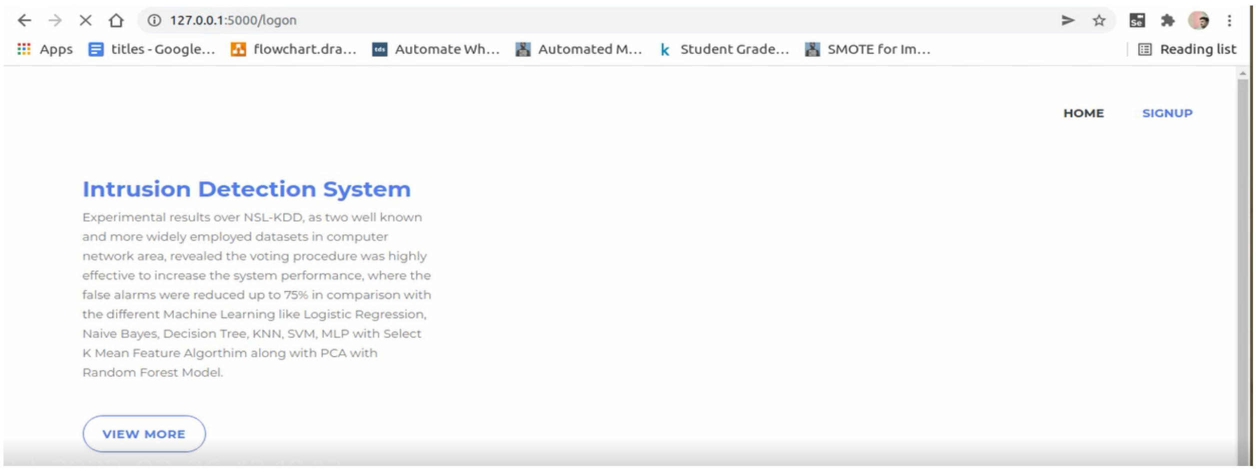
6.1.5 User Interface Testing

User interface testing, a testing technique used to identify the presence of defects is a product/software under test by Graphical User interface [GUI].

6.2 TEST CASES:

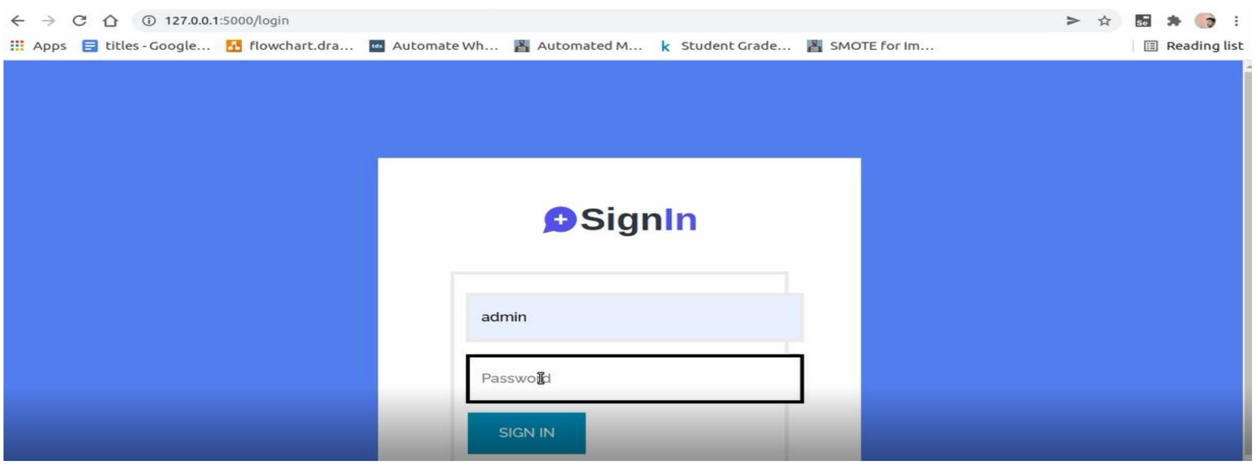
S.NO	INPUT	If available	If not available
1	Upload data	Data loaded	There is no process
2	Model generation	All algorithms performed on that data	There is no process
3	Predict attack class	Attack class predicted	There is no process

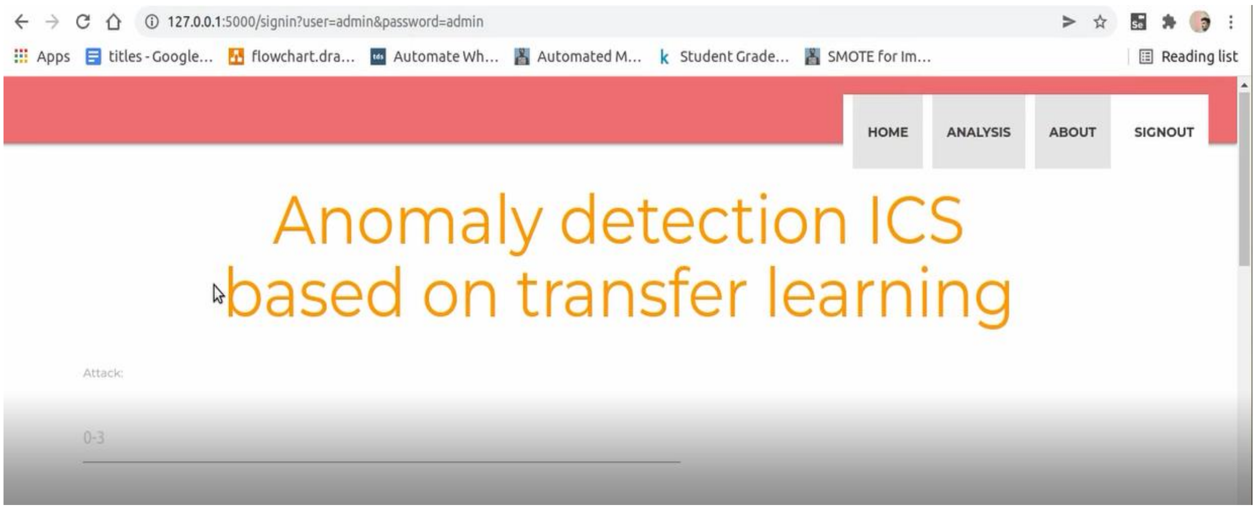
7. RESULT ANALYSIS



Intrusion Detection System

Experimental results over NSL-KDD, as two well known and more widely employed datasets in computer network area, revealed the voting procedure was highly effective to increase the system performance, where the false alarms were reduced up to 75% in comparison with the different Machine Learning like Logistic Regression, Naive Bayes, Decision Tree, KNN, SVM, MLP with Select K Mean Feature Algorthim along with PCA with Random Forest Model.

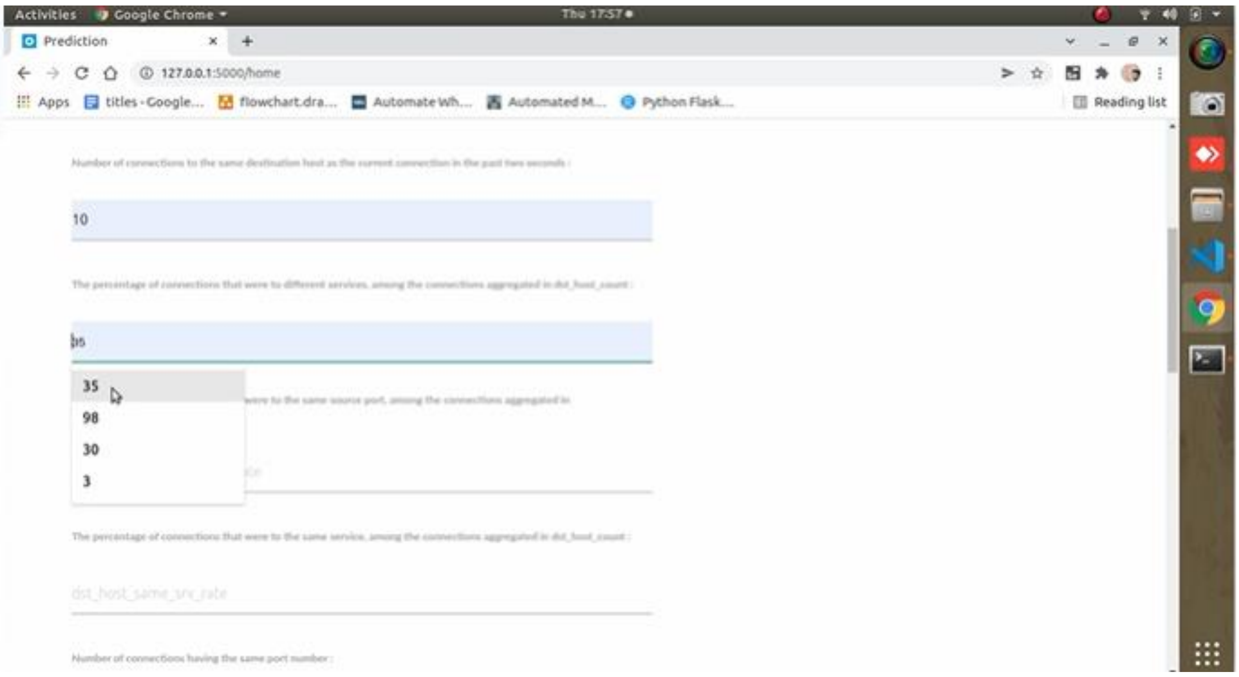



127.0.0.1:5000/signin?user=admin&password=admin

Anomaly detection ICS based on transfer learning

Attack:

0-3


Prediction

127.0.0.1:5000/home

Number of connections to the same destination host as the current connection in the past two seconds :

10

The percentage of connections that were to different services, among the connections aggregated in dst_host_count :

35

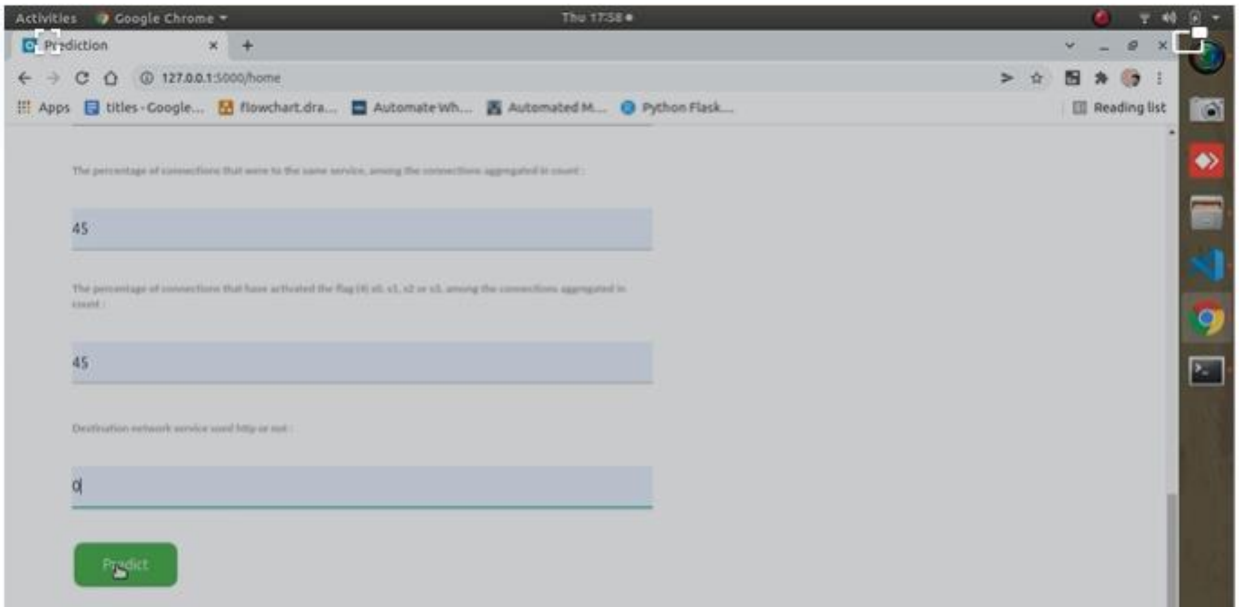
35
98
30
3

were to the same source port, among the connections aggregated in

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count :

dst_host_same_srv_rate

Number of connections having the same port number :


Prediction

127.0.0.1:5000/home

The percentage of connections that were to the same service, among the connections aggregated in count :

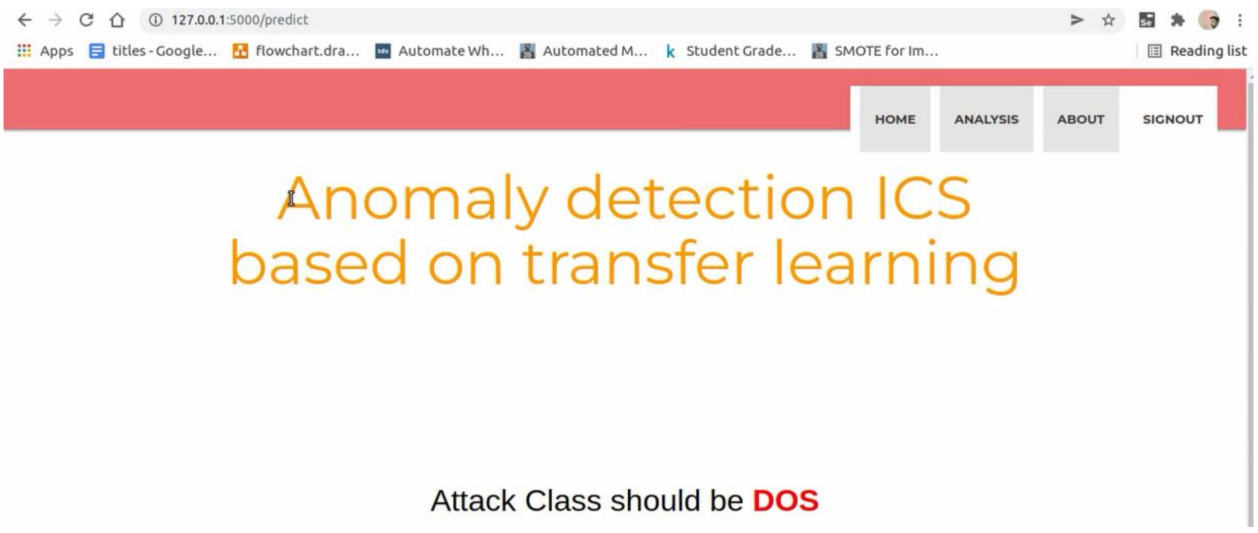
45

The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :

45

Destination network service used http or not :

0



Attack Class should be **DOS**

8. CONCLUSION

Network security is a popular and important topic. The network security of ICSs is of great importance for a country. This paper uses data visualization to convert flow data into images. Specifically, we build an eight-layer residual neural network and use fine-tuning technology for transfer learning to detect abnormal datasets of ICSs. Experimental results show that transfer learning for residual CNNs is effective in this field. The depth of the model also ensures that it has a certain generalizability. The residual structure effectively prevents gradient explosion or gradient disappearance. The model can provide reliable predictions for unknown or differently distributed abnormal data through short-term training by transfer learning. Compared with other anomaly detection algorithms, the algorithm proposed in this paper results in superior indicators. The method we proposed not only solves the problem associated with training time for deep learning models by transfer learning, but also meets the requirements of ICSs in terms of evaluation indicators. At present, the model we constructed solves the two classification problem, but a refined classification of abnormal traffic data is still desirable. In the future work, we will perform multi classification of abnormal traffic data, track the characteristics of different abnormal data types, and then reliably classify them to further ensure network security in ICSs.

9. REFERENCES

Textbooks:

1. Programming Python, Mark Lutz
2. Head First Python, Paul Barry
3. Core Python Programming, R. Nageswara Rao
4. Learning with Python, Allen B. Downey

Journals:

[1] A. R. Sadeghi, C. Wachsmann, and M. Waidner, Security and privacy challenges in industrial Internet of Things, in Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, **2015, pp. 1–6.**

[2] L. Obergon, InfoSec reading room secure architecture for industrial control systems, SANS Institute InfoSec, GIAC(GSEC) Gold Certification, **Vol. 1, pp. 1–27, 2014.**

[3] C. Markman, A. Wool, and A. A. Cardenas, A new burstDFA model for SCADA anomaly detection, in Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy, Dallas, TX, USA, **2017, pp. 1–12.**

[4] M. Mantere, I. Uusitalo, M. Sailio, and S. Noponen, Challenges of machine learning based monitoring for industrial control system networks, in Proceedings of the 2012 26th International Conference on Advanced Information Networking and Applications Workshops, Fukuoka, Japan, **2012, pp. 968–972.**

[5] R. Zhao, R. Q. Yan, Z. H. Chen, K. Z. Mao, P. Wang, and R. X. Gao, Deep learning and its applications to machine health monitoring: A survey, Mechanical System and Signal Processing, **Vol. 115, pp. 213–237, 2019.**

[6] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Q. Zhou, W. Li, and P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research, **Vol. 21, No. 140, pp. 1–67, 2020.**

[7] S. N. Shirazi, A. Gouglidis, K. N. Syeda, S. Simpson, A. Mauthe, I. M. Stephanakis, and D. Hutchison, Evaluation of anomaly detection techniques for SCADA communication resilience, in Proceedings of the 2016 Resilience Week (RWSr), Chicago, IL, USA, 2016, **pp. 140–145.**

[8] Y. Lai, J. Zhang, and Z. liu,, Industrial anomaly detection and attack classification method based on convolutional neural network, Security and Communication Networks, **doi: 10.1155/2019/8124254.**

[9] J. Hurley, A. Munoz, and S. Sezer, ITACA: Flexible, scalable network analysis, in Proceedings of the 2012 IEEE International Conference on Communications (ICC), Ottawa, Canada, 2012, **pp. 1069–1073.**

[10] G. Thatte, U. Mitra, and J. Heidemann, Parametric methods for anomaly detection in aggregate traffic, IEEE/ACM Transactions On Networking, **Vol. 19, No. 2, pp. 512–525, 2010.**

[11] A. Terai, S. Abe, K. Shoya, Y. Takano, and I. Koshijima, Cyber-attack detection for industrial control system monitoring with support vector machine based on communication profile, in Proceedings of the 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Paris, France, **2017, pp. 132–138.**

[12] C. Zhou, S. Huang, N. Xiong, S. Yang, H. Li, Y. Qin, and X. Li, Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation, IEEE Transactions on Systems, Man, and Cybernetics: Systems, **Vol. 45, No. 10, pp. 1345–1360, 2015.**

[13] M. Zhang, B. Y. Xu, and J. Gong, An anomaly detection model based on one-class SVM to detect network intrusions, in Proceedings of the 2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN), Shenzhen, China, **2015, pp. 102–107.**

[14] S. C. Zhang, X. Y. Xie, and Y. Xu, Intrusion detection method based on a deep convolutional neural network, Tsinghua Science and Technology, **Vol. 59, No. 1, pp. 44–52, 2019.**

[15] A. Almalawi, X. H. Yu, Z. Tari, A. Fahad, and I. Khalil, An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems, Computers & Security, **Vol. 46, pp. 94–110, 2014.**

Websites:

- https://www.w3schools.com/python/
- https://www.tutorialspoint.com/python/index.htm
- https://www.javatpoint.com/python-tutorial
- https://www.learnpython.org/
- https://www.pythontutorial.net/