**1.INTRODUCTION**

**1.1 Use of Artificial Neural Network to Identify Fake Profile**

In 2017 Facebook reached a total population of 2.46 billion users making it the most popular choice of social media [1]. Social media networks make revenues from the data provided by users. The average user does not know that their rights are given up the moment they use the social media network's service. Social media companies have a lot to gain at the expense of the user. Every time a user shares a new location, new photos, likes, dislikes, and tag other users in content posted, Facebook makes revenue via advertisements and data. More specifically, the average American user generates about $26.76 per quarter [2]. That number adds up quickly when millions of users are involved. In today's digital age, the ever-increasing dependency on computer technology has left the average citizen vulnerable to crimes such as data breaches and possible identity theft. These attacks can occur without notice and often without notification to the victims of a data breach. At this time, there is little incentive for social networks to improve their data security. These breaches often target social media networks such as Facebook and Twitter. They can also target banks and other financial institutions. There seems to be a newsworthy issue involving social media networks getting hacked every day. Recently, Facebook had a data breach which affected about 50 million users [3]. Facebook provides a set of clearly defined provisions that explain what they do with the user's data [4]. The policy does very little to prevent the constant exploitation of security and privacy. Fake profiles seem to slip through Facebook's built-in security features. The other dangers of personal data being obtained for fraudulent purposes is the presence of bots and fake profiles. Bots are programs that can gather information about the user without the user even knowing. This process is known as web scraping. What is worse, is that this action is legal. Bots can be hidden or come in the form of a fake friend request on a social network site to gain access to private information. The solution presented in this paper intends to focus on the dangers of a bot in the form of a fake profile on your social media. This solution would come in the form of an algorithm. The language that we chose to use is Python. The algorithm would be able to determine if a current friend request that a user gets online is an actual person or if it is a bot or it is a fake friend request fishing for information. Our algorithm would work with the help of the social media companies, as we would need a training dataset from them to train our model and later verify if the profiles are fake or not. The algorithm could even work as a traditional layer on the user's web browser as a browser plug-in.

**1.2 Problem Definition**

We use machine learning, namely an artificial neural network to determine what are the chances that Facebook friend request is authentic or not. We also outline the classes and libraries involved. The social network, is a crucial part of our life is plagued by online impersonation and fake accounts. Inthis project, we propose a model that could be used to classify an account is a fake or genuine, this is an automatic detection method, it can be applied easily by social networks which has millions of profiles, whose profiles cannot be examined manually.

**1.3 Objective of project**

Using Artificial Neural Network, we are identifying whether given account details are from genuine or fake users. ANN algorithm will be trained with all previous users fake and genuine account details and then whenever we gave new test data then that ANN train model will be applied on new test data to identity whether given new account details are from genuine or fake users.

Online social networks such as Facebook or Twitter contains users details and some malicious users will hack social network database to steal or breach users information, To protect users data we are using ANN Algorithm. To train ANN algorithm we are using below details from social network.

**1.4 Organization of project**

* + 1. **Data Collection**

Data Collection is the process of collection the data from different sources. We are collection the data from Kaggle, we need real world Facebook, Twitter and other OSN’s dataset, which are publicly unavailable. Anonymized datasets of social groups are available including some profile-based feature but the anonymized form cannot be used. Therefore, in some works data was collected from Facebook API. Twitter API, however it is restricted from the data that can be achieved due to privacy issues. The collected data is in the csv file format. Bellow we have a sample dataset with the attribute names as follows

Account \_Age, Gender, User\_ Age, Link \_Description, Status Count, Friend\_Count, Location, Location\_IP, Status.

* + 1. **Data Pre-processing**

Social media data is highly raw, so there was a need of data cleansing. Data pre-processing involves various steps such as tokenization, stop word removal, stemming and lemmatization.

Along with above steps short word removal, Punctuation mark removal, Numeric and Special character removal, lower case conversion has been performed for better performance of machine learning algorithms.

* + 1. **Model Building**

We use machine learning, namely an artificial neural network to determine what are the chances that Facebook friend request is authentic or not. We also outline the classes and libraries involved, to demonstrate how to build a ANN neural network based image classifier, we shall build a 6 layer neural network that will identify and separate one image from other. To predict image class multiple layers operate on each other to get best match layer and this process continues till no more improvement left. We utilize Microsoft Excel to store old and new fake data profiles. The algorithm then stores the data in a data frame. This collection of data will be divided into a training set and a testing set. We would need a data set from the social media sites to train our model.

* + 1. **Prediction**

Artificial Neural Networks we are identifying whether given account details are from Genuine or Fake users. ANN algorithm will be trained with all previous users fake or genuine account data and then whenever we gave new test data then that ANN train model will be applied on new data to identify whether given new account details are from Genuine or Fake users.

**1.5 Existing System**

Malicious users create fake profiles to phish login information from unsuspecting users. A fake profile will send friend requests to many users with public profiles. These counterfeit profiles bait unsuspecting users with pictures of people that are considered attractive. Once the user accepts the request, the owner of the phony profile will spam friend requests to anyone this user is a friend. The fake profile's contents typically have links that lead to an external website where the damage happens. An unaware curious user clicking the bad link will damage their computer. The cost can be as simple as catching a virus to as bad as installing a rootkit turning the computer into a zombie. While Facebook has a rigorous screening to keep these fake accounts

**1.6 PROPOSED SYSTEM**

In our solution, we use machine learning, namely an artificial neural network to determine what are the chances that a friend request is authentic or not. We utilize Microsoft Excel to store old and new fake data profiles. The algorithm then stores the data in a data frame. This collection of data will be divided into a training set and a testing set. We would need a data set from the social media sites to train our model. • For the training set, the features that we use to determine a fake profile are Account age, Gender, User age, Link in the description.

1.5 Organization

In the rest of the paper, Sect. 2presents the system overview under the points, data

collection, feature selection, feature extraction and classiﬁcation techniques and also

presents the general architecture for identiﬁcation of fake accounts. Section 3repre-

sents related work done in this direction. Section 4is ﬁndings and discussion in which

comparative study of previous researches in done and gaps of existing work are

mentioned, Sect. 5ﬁnally concludes the research subject.

2 System Overview

OSNs attract various kind of malicious and illegal activities and in the past the com-

munity of research has put forward various solutions to the problem [12].

In a general approach for the identiﬁcation of fake accounts in large scale online

social networks following steps are used (Fig. 2):

•Data collection.

•Feature selection.

•Feature extraction.

•Data classiﬁcation/techniques used.

2.1 Data Collection

In this work, we need real-world Facebook, Twitter and other OSN’s datasets, which

are publicly unavailable. Anonymized datasets of social graph are available including

some proﬁle-based feature but the anonymized form cannot be used. Therefore, in

some works data was collected from Facebook API, Twitter API, however it is

restricted from the data that can be achieved due to privacy issues.

In Boshmaf et al. [2] 100 automated fake a

1.5 Organization

In the rest of the paper, Sect. 2presents the system overview under the points, data

collection, feature selection, feature extraction and classiﬁcation techniques and also

presents the general architecture for identiﬁcation of fake accounts. Section 3repre-

sents related work done in this direction. Section 4is ﬁndings and discussion in which

comparative study of previous researches in done and gaps of existing work are

mentioned, Sect. 5ﬁnally concludes the research subject.

2 System Overview

OSNs attract various kind of malicious and illegal activities and in the past the com-

munity of research has put forward various solutions to the problem [12].

In a general approach for the identiﬁcation of fake accounts in large scale online

social networks following steps are used (Fig. 2):

•Data collection.

•Feature selection.

•Feature extraction.

•Data classiﬁcation/techniques used.

2.1 Data Collection

In this work, we need real-world Facebook, Twitter and other OSN’s datasets, which

are publicly unavailable. Anonymized datasets of social graph are available including

some proﬁle-based feature but the anonymized form cannot be used. Therefore, in

some works data was collected from Facebook API, Twitter API, however it is

restricted from the data that can be achieved due to privacy issues.

In Boshmaf et al. [2] 100 automated fake a

1.5 Organization

In the rest of the paper, Sect. 2presents the system overview under the points, data

collection, feature selection, feature extraction and classiﬁcation techniques and also

presents the general architecture for identiﬁcation of fake accounts. Section 3repre-

sents related work done in this direction. Section 4is ﬁndings and discussion in which

comparative study of previous researches in done and gaps of existing work are

mentioned, Sect. 5ﬁnally concludes the research subject.

2 System Overview

OSNs attract various kind of malicious and illegal activities and in the past the com-

munity of research has put forward various solutions to the problem [12].

In a general approach for the identiﬁcation of fake accounts in large scale online

social networks following steps are used (Fig. 2):

•Data collection.

•Feature selection.

•Feature extraction.

•Data classiﬁcation/techniques used.

2.1 Data Collection

In this work, we need real-world Facebook, Twitter and other OSN’s datasets, which

are publicly unavailable. Anonymized datasets of social graph are available including

some proﬁle-based feature but the anonymized form cannot be used. Therefore, in

some works data was collected from Facebook API, Twitter API, however it is

restricted from the data that can be achieved due to privacy issues.

In Boshmaf et al. [2] 100 automated fake a

n this work, we need real-world Facebook, Twitter and other OSN’s datasets, which

are publicly unavailable. Anonymized datasets of social graph are available including

some proﬁle-based feature but the anonymized form cannot be used. Therefore, in

some works data was collected from Facebook API, Twitter API, however it is

restricted from the data that can be achieved due to privacy issues

**2. LITERATURE SURVEY**

This literature review will identify major themes related to the subject of this study. Also, similarities and differences in previous studies will be considered to enrich the project’s content to make it unique and innovative to identify fake profiles.

**2.1 Introduction**

In 2017 Facebook reached a total population of 2.46 billion users making it the most popular choice of social media [1]. Social media networks make revenues from the data provided by users. The average user does not know that their rights are given up the moment they use the social media network's service. Social media companies have a lot to gain at the expense of the user. Every time a user shares a new location, new photos, likes, dislikes, and tag other users in content posted, Facebook makes revenue via advertisements and data.

More specifically, the average American user generates about $26.76 per quarter [2]. That number adds up quickly when millions of users are involved. In today's digital age, the ever-increasing dependency on computer technology has left the average citizen vulnerable to crimes such as data breaches and possible identity theft. These attacks can occur without notice and often without notification to the victims of a data breach. At this time, there is little incentive for social networks to improve their data security. These breaches often target social media networks such as Facebook and Twitter. They can also target banks and other financial institutions. There seems to be a newsworthy issue involving social media networks getting hacked every day. Recently, Facebook had a data breach which affected about 50 million users [3]. Facebook provides a set of clearly defined provisions that explain what they do with the user's data [4]. The policy does very little to prevent the constant exploitation of security and privacy. Fake profiles seem to slip through Facebook's built-in security features. The other dangers of personal data being obtained for fraudulent purposes is the presence of bots and fake profiles. Bots are programs that can gather information about the user without the user even knowing. This process is known as web scraping. What is worse, is that this action is legal. Bots can be hidden or come in the form of a fake friend request on a social network site to gain access to private information. The solution presented in this paper intends to focus on the dangers of a bot in the form of a fake profile on your social media.

**2.2 Summary**

From literature survey we conclude that,use machine learning, namely an artificial neural network to determine what are the chances that Facebook friend request is authentic or not. We also outline the classes and libraries involved to demonstrate how to build a ANN neural network based image classifier, we shall build a 6 layer neural network that will identify and separate one image from other. To predict image class multiple layers, operate on each other to get best match layer and this process continues till no more improvement left.

**3.DOMAIN**

**3.1 MACHINE LEARNING**

Machine Learning is said as a subset of **artificial intelligence** that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by **Arthur Samuel**in**1959.** With the help of sample historical data, which is known as **training data**, machine learning algorithms build a **mathematical model** that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the information, the higher will be the performance.

**3.2 ADVANTAGES OF ML**

**1. Automation**

Machine Learning is one of the **driving forces** behind automation, and it is cutting down time and human workload. Automation can now be seen everywhere, and the complex algorithm does the hard work for the user. Automation is more reliable, efficient, and quick. With the help of machine learning, now advanced computers are being designed. Now this advanced computer can handle several machine-learning models and complex algorithms. However, automation is spreading faster in the industry but, a lot of research and innovation are required in this field.

**2. Scope of Improvement**

Machine Learning is a field where things keep evolving. It gives many opportunities for improvement and can become the leading technology in the future. A lot of research and innovation is happening in this technology, which helps improve software and hardware.

**3. Enhanced Experience in Online Shopping and Quality Education**

Machine Learning is going to be used in the education sector extensively, and it will be going to enhance the quality of education and student experience. It has emerged in China; machine learning has improved student focus. In the e-commerce field, Machine Learning studies your search **feed and give suggestion** based on them. Depending upon search and browsing history, it pushes targeted advertisements and notifications to users.

**4. Wide Range of Applicability**

This technology has a very wide range of applications. Machine learning plays a role in almost every field, **like hospitality, ed-tech, medicine, science, banking, and business**. It creates more opportunities.

**3.3 DISADVANTAGES OF ML**

**1. Data Acquisition**

The whole concept of machine learning is about identifying useful data. The outcome will be incorrect if a credible data source is not provided. The quality of the data is also significant. If the user or institution needs more quality data, wait for it. It will cause delays in providing the output. So, machine learning significantly depends on the data and its quality.

**2. Time and Resources**

The data that machines process remains huge in quantity and differs greatly. Machines require time so that their algorithm can adjust to the environment and learn it. Trials runs are held to check the accuracy and reliability of the machine. It requires massive and expensive resources and high-quality expertise to set up that quality of infrastructure. Trials runs are costly as they would cost in terms of time and expenses.

**3. Results Interpretations**

One of the biggest advantages of Machine learning is that interpreted data that we get from the cannot be hundred percent accurate. It will have some degree of inaccuracy. For a high degree of accuracy, algorithms should be developed so that they give reliable results.

**4. High Error Chances**

The error committed during the initial stages is huge, and if not corrected at that time, it creates havoc. Biasness and wrongness have to be dealt with separately; they are not interconnected. Machine learning depends on two factors, **i.e., data and algorithm**. All the errors are dependent on the two variables. Any incorrectness in any variables would have huge repercussions on the output.

**3.4 APPLICATIONS OF ML**

**1.Human Resource Information Systems**: In short, it is also called an HRIS System, and it is used for identifying the best candidates for an open position using machine learning models to filter the applications

**2.Business Intelligence:** In short, it is called BI. Machine Learning is used by vendors in their software to search potentially important anomalies and patterns of data points.

**3.Customer Relationship Management:**The Machine Learning model used by the CRM software analyzes prompt sales members responding to important messages first and email.

**4.Virtual Assistants:** Smart assistants usually combine unsupervised and supervised learning machine learning models to decipher supply context and natural speech.

**5.Self-Driving cars:** Algorithms based on the Machine Learning model are used to drive the car.

**3.5 TYPES OF ML**

## 1. Supervised Machine Learning

As its name suggests, [Supervised machine learning](https://www.javatpoint.com/supervised-machine-learning) is based on supervision. It means in the supervised learning technique, we train the machines using the "labelled" dataset, and based on the training, the machine predicts the output. Here, the labelled data specifies that some of the inputs are already mapped to the output. More preciously, we can say; first, we train the machine with the input and corresponding output, and then we ask the machine to predict the output using the test dataset.

Let's understand supervised learning with an example. Suppose we have an input dataset of cats and dog images. So, first, we will provide the training to the machine to understand the images, such as the **shape & size of the tail of cat and dog, Shape of eyes, colour, height (dogs are taller, cats are smaller), etc.** After completion of training, we input the picture of a cat and ask the machine to identify the object and predict the output. Now, the machine is well trained, so it will check all the features of the object, such as height, shape, colour, eyes, ears, tail, etc., and find that it's a cat. So, it will put it in the Cat category. This is the process of how the machine identifies the objects in Supervised Learning.

**The main goal of the supervised learning technique is to map the input variable(x) with the output variable(y).**Some real-world applications of supervised learning are**Risk Assessment, Fraud Detection, Spam filtering,**etc.

## 2.Unsupervised Machine Learning

[Unsupervised learnin](https://www.javatpoint.com/unsupervised-machine-learning)g is different from the Supervised learning technique; as its name suggests, there is no need for supervision. It means, in unsupervised machine learning, the machine is trained using the unlabeled dataset, and the machine predicts the output without any supervision.

In unsupervised learning, the models are trained with the data that is neither classified nor labelled, and the model acts on that data without any supervision.

**The main aim of the unsupervised learning algorithm is to group or categories the unsorted dataset according to the similarities, patterns, and differences.** Machines are instructed to find the hidden patterns from the input dataset.

Let's take an example to understand it more preciously; suppose there is a basket of fruit images, and we input it into the machine learning model. The images are totally unknown to the model, and the task of the machine is to find the patterns and categories of the objects.

**3. Semi-Supervised Learning**

Semi-Supervised learning is a type of Machine Learning algorithm that lies between Supervised and Unsupervised machine learning. It represents the intermediate ground between Supervised (With Labeled training data) and Unsupervised learning (with no labeled training data) algorithms and uses the combination of labeled and unlabeled datasets during the training period.

**A**lthough Semi-supervised learning is the middle ground between supervised and unsupervised learning and operates on the data that consists of a few labels, it mostly consists of unlabeled data. As labels are costly, but for corporate purposes, they may have few labels. It is completely different from supervised and unsupervised learning as they are based on the presence & absence of labels.

## 4. Reinforcement Learning

**Reinforcement learning works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, taking action, learning from experiences, and improving its performance.** Agent gets rewarded for each good action and get punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards.

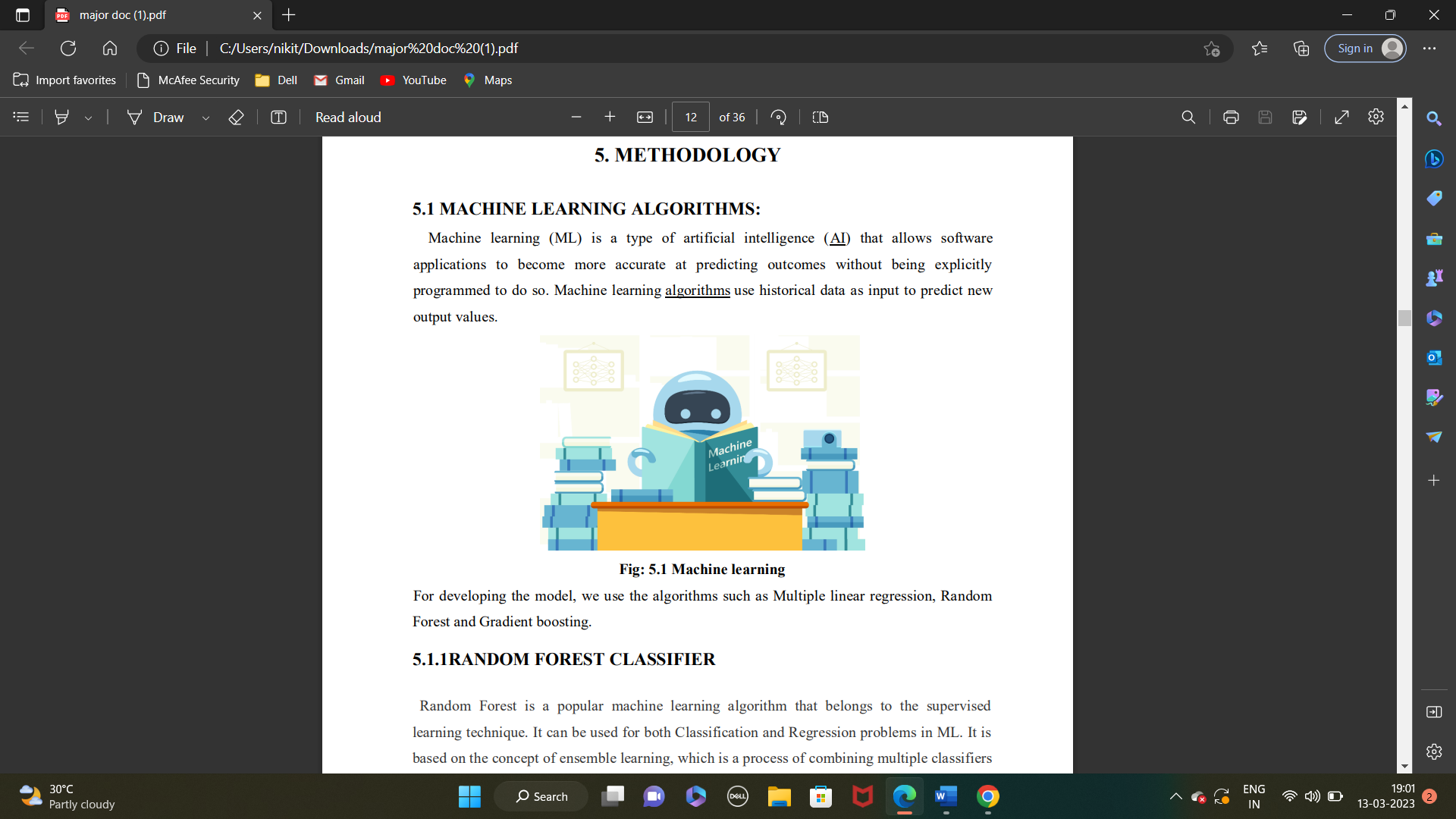
In reinforcement learning, there is no labelled data like supervised learning, and agents learn from their experiences only.

The [reinforcement learning](https://www.javatpoint.com/reinforcement-learning) process is similar to a human being; for example, a child learns various things by experiences in his day-to-day life. An example of reinforcement learning is to play a game, where the Game is the environment, moves of an agent at each step define states, and the goal of the agent is to get a high score. Agent receives feedback in terms of punishment and rewards.Due to its way of working, reinforcement learning is employed in different fields such as **Game theory, Operation Research, Information theory, multi-agent systems.** A reinforcement learning problem can be formalized using **Markov Decision Process(MDP).** In MDP, the agent constantly interacts with the environment and performs actions; at each action, the environment responds and generates a new state.

**4. METHODOLOGY**

**4.1 Machine Learning**

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.



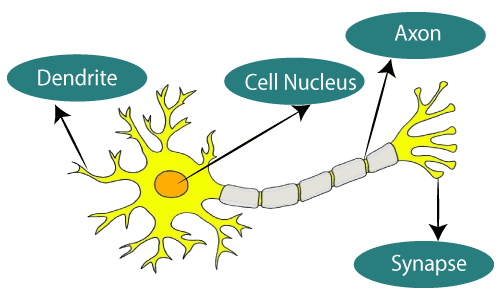
**Fig: 4.1 Machine Learning**

**4.1.1 Artificial Neural Networks**

Artificial Neural Network Tutorial provides basic and advanced concepts of ANNs. Our Artificial Neural Network tutorial is developed for beginners as well as professions.

The term "Artificial neural network" refers to a biologically inspired sub-field of artificial intelligence modelled after the brain. An Artificial neural network is usually a computational network based on biological neural networks that construct the structure of the human brain. Similar to a human brain has neurons interconnected to each other, artificial neural networks also have neurons that are linked to each other in various layers of the networks. These neurons are known as nodes. Artificial neural network tutorial covers all the aspects related to the artificial neural network. In this tutorial, we will discuss ANNs, Adaptive resonance theory, Korhonen self-organizing map, Building blocks, unsupervised learning, Genetic algorithm, etc.

The term "**Artificial Neural Network**" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.

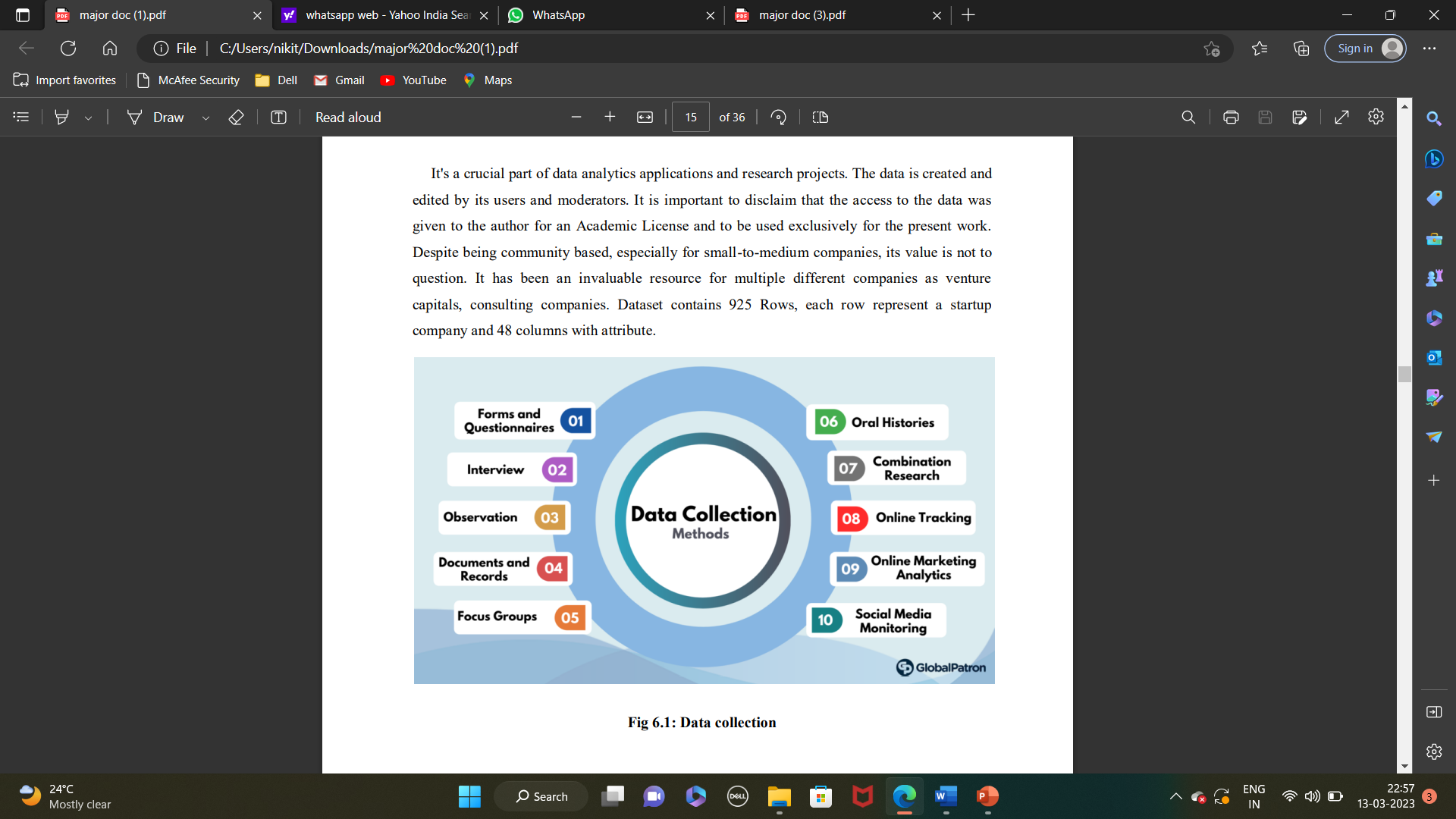


**Fig: 4.1.1 Biological Neural Network**

Dendrites from Biological Neural Network represent inputs in Artificial Neural Networks, cell nucleus represents Nodes, synapse represents Weights, and Axon represents Output.

|  |  |
| --- | --- |
| **Biological Neural Network** | **Artificial Neural Network** |
| Dendrites | Inputs |
| Cell nucleus | Nodes |
| Synapse | Weights |
| Axon | Output |

**FIG 4.1.2:Relationship between Biological neural network and artificial neural network**:

An **Artificial Neural Network** in the field of **Artificial intelligence** where it attempts to mimic the network of neurons makes up a human brain so that computers will have an option to understand things and make decisions in a human-like manner. The artificial neural network is designed by programming computers to behave simply like interconnected brain cells.

There are around 1000 billion neurons in the human brain. Each neuron has an association point somewhere in the range of 1,000 and 100,000. In the human brain, data is stored in such a manner as to be distributed, and we can extract more than one piece of this data when necessary from our memory parallelly. We can say that the human brain is made up of incredibly amazing parallel processors.

We can understand the artificial neural network with an example, consider an example of a digital logic gate that takes an input and gives an output. "OR" gate, which takes two inputs. If one or both the inputs are "On," then we get "On" in output. If both the inputs are "Off," then we get "Off" in output. Here the output depends upon input. Our brain does not perform the same task. The outputs to inputs relationship keep changing because of the neurons in our brain, which are "learning."

**4.1.2 History**

Warren McCulloch and Walter Pitts(1943) opened the subject by creating a computational model for neural networks. In the late 1940s, D. O. Hebbcreated a learning hypothesis based on the mechanism of neural plasticity that became known as Hebbian learning. Farley and Wesley A. Clark(1954) first used computational machines, then called "calculators", to simulate a Hebbian network. In 1958, psychologist Frank Rosenblatt invented the perceptron, the first artificial neural network,funded by the United States Office of Naval Research. The first functional networks with many layers were published by Ivakhnenko and Lapa in 1965, as the Group Method of Data Handling. The basics of continuous backpropagationwere derived in the context of control theory by Kelley in 1960 and by Bryson in 1961, using principles of dynamic programming. Thereafter research stagnated following Minsky and Papert (1969),who discovered that basic perceptron’s were incapable of processing the exclusive-or circuit and that computers lacked sufficient power to process useful neural networks.

In 1970, Seppo Linnainmaa published the general method for automatic differentiation (AD) of discrete connected networks of nested differentiable functions. In 1973, Dreyfus used backpropagation to adapt parameters of controllers in proportion to error gradients. WebOS’s (1975) backpropagation algorithm enabled practical training of multi-layer networks. In 1982, he applied Linnainmaa's AD method to neural networks in the way that became widely used.

The development of metal–oxide–semiconductor (MOS) very-large-scale integration (VLSI), in the form of complementary MOS (CMOS) technology, enabled increasing MOS transistor counts in digital electronics. This provided more processing power for the development of practical artificial neural networks in the 1980s.

In 1986 Rumelhart, Hinton and Williams showed that backpropagation learned interesting internal representations of words as feature vectors when trained to predict the next word in a sequence.

From 1988 onward the use of neural networks transformed the field of protein structure prediction, in particular when the first cascading networks were trained on *profiles* (matrices) produced by multiple sequence alignments.

In 1992, max-pooling was introduced to help with least-shift invariance and tolerance to deformation to aid 3D object recognition. Schmid Huber adopted a multi-level hierarchy of networks (1992) pre-trained one level at a time by unsupervised learning and fine-tuned by backpropagation.

Neural networks' early successes included predicting the stock market and in 1995 a (mostly) self-driving car.

Geoffrey Hinton et al. (2006) proposed learning a high-level representation using successive layers of binary or real-valued latent variables with a restricted Boltzmann machineto model each layer. In 2012, Ng and Dean created a network that learned to recognize higher-level concepts, such as cats, only from watching unlabelled images. Unsupervised pre-training and increased computing power from GPUs and distributed computing allowed the use of larger networks, particularly in image and visual recognition problems, which became known as "deep learning".

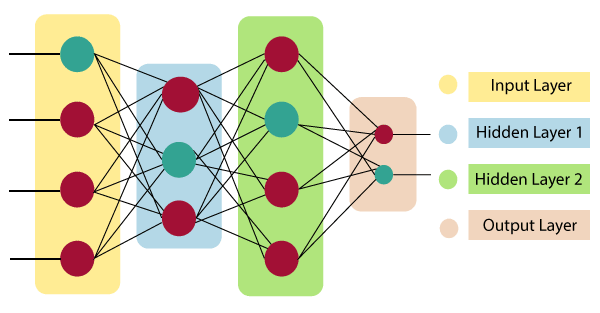
Ciresan and colleagues (2010)showed that despite the vanishing gradient problem, GPUs make backpropagation feasible for many-layered feedforward neural networks. Between 2009 and 2012, ANNs began winning prizes in image recognition contests, approaching human level performance on various tasks, initially in pattern recognition and handwriting recognition. For example, the bi-directional and multi-dimensional long short-term memory (LSTM)of Graves et al. won three competitions in connected handwriting recognition in 2009 without any prior knowledge about the three languages to be learned.

Ciresan and colleagues built the first pattern recognizers to achieve human-competitive/superhuman performanceon benchmarks such as traffic sign recognition (IJCNN 2012).

**4.1.3 Architecture of Artificial Neural Network**

To understand the concept of the architecture of an artificial neural network, we have to understand what a neural network consists of. In order to define a neural network that consists of a large number of artificial neurons, which are termed units arranged in a sequence of layers. Let’s us look at various types of layers available in an artificial neural network.

Artificial Neural Network primarily consists of three layers:



**Fig: 4.1.3 Architecture of Artificial Neural Network**

**Input Layer:**

As the name suggests, it accepts inputs in several different formats provided by the programmer.

**Hidden Layer:**

The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

**Output Layer:**

The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

What is Artificial Neural Network

It determines weighted total is passed as an input to an activation function to produce the output. Activation functions choose whether a node should fire or not. Only those who are fired make it to the output layer. There are distinctive activation functions available that can be applied upon the sort of task we are performing.

Artificial Neural Network can be best represented as a weighted directed graph, where the artificial neurons form the nodes. The association between the neurons outputs and neuron inputs can be viewed as the directed edges with weights. The Artificial Neural Network receives the input signal from the external source in the form of a pattern and image in the form of a vector. These inputs are then mathematically assigned by the notations x(n) for every n number of inputs. Afterward, each of the input is multiplied by its corresponding weights (these weights are the details utilized by the artificial neural networks to solve a specific problem). In general terms, these weights normally represent the strength of the interconnection between neurons inside the artificial neural network. All the weighted inputs are summarized inside the computing unit.

If the weighted sum is equal to zero, then bias is added to make the output non-zero or something else to scale up to the system's response. Bias has the same input, and weight equals to 1. Here the total of weighted inputs can be in the range of 0 to positive infinity. Here, to keep the response in the limits of the desired value, a certain maximum value is benchmarked, and the total of weighted inputs is passed through the activation function.

The activation function refers to the set of transfer functions used to achieve the desired output. There is a different kind of the activation function, but primarily either linear or non-linear sets of functions. Some of the commonly used sets of activation functions are the Binary, linear, and Tan hyperbolic sigmoidal activation functions. Let us take a look at each of them in details:

**Binary:** In binary activation function, the output is either a one or a 0. Here, to accomplish this, there is a threshold value set up. If the net weighted input of neurons is more than 1, then the final output of the activation function is returned as one or else the output is returned as 0.

**Sigmoidal Hyperbolic**: The Sigmoidal Hyperbola function is generally seen as an "**S**" shaped curve. Here the tan hyperbolic function is used to approximate output from the actual net input. The function is defined as:

F(x) = (1/1 + exp(-????x)).

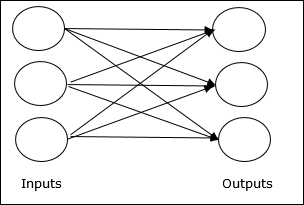
**4.1.4 Types of Artificial Neural Network**

There are various types of Artificial Neural Networks (ANN) depending upon the human brain neuron and network functions, an artificial neural network similarly performs tasks. The majority of the artificial neural networks will have some similarities with a more complex biological partner and are very effective at their expected tasks. For example, segmentation or classification.

### **1.Feedforward Network**

It is a non-recurrent network having processing units/nodes in layers and all the nodes in a layer are connected with the nodes of the previous layers. The connection has different weights upon them. There is no feedback loop means the signal can only flow in one direction, from input to output. It may be divided into the following two types –

* **Single layer feedforward network** − The concept is of feedforward ANN having only one weighted layer. In other words, we can say the input layer is fully connected to the output layer.



**Fig 4.1.4.1 : Feed forward network**

**Multilayer feedforward network** − The concept is of feedforward ANN having more than one weighted layer. As this network has one or more layers between the input and the output layer, it is called hidden layers.

### Multilayer feedforward network

**Fig4.1.4.2: Multilayer feedforward network**

As the name suggests, a feedback network has feedback paths, which means the signal can flow in both directions using loops. This makes it a non-linear dynamic system, which changes continuously until it reaches a state of equilibrium. It may be divided into the following types −

* **Recurrent networks** − They are feedback networks with closed loops. Following are the two types of recurrent networks.
* **Fully recurrent network** − It is the simplest neural network architecture because all nodes are connected to all other nodes and each node works as both input and output**.**
* **Jordan network** − It is a closed loop network in which the output will go to the input again as feedback as shown in the following diagram.

**2. Modular Neural Networks**

In this type of neural network, many independent networks contribute to the results collectively. There are many sub-tasks performed and constructed by each of these neural networks. This provides a set of inputs that are unique when compared with other neural networks. There is no signal exchange or interaction between these neural networks to accomplish any task.

### **3. Radial basis function Neural Network**

There are two layers in the functions of RBF. These are used to consider the distance of a centre with respect to the point. In the first layer, features in the inner layer are united with the Radial Basis Function. In the next step, the output from this layer is considered for computing the same output in the next iteration. One of the applications of Radial Basis function can be seen in Power Restoration Systems. There is a need to restore the power as reliably and quickly as possible after a blackout.

### **4. Kohonen Self Organizing Neural Network**

In this neural network, vectors are input to a discrete map from an arbitrary dimension. Training data of an organization is created by training the map. There might be one or two dimensions on the map. The weight of the neurons may change that depends on the value.

The neuron’s location will not change while training the map and will stay constant. Input vector and small weight are given to every neuron value in the first phase of the self-organization process. A winning neuron is a neuron that is closest to the point. Other neurons will also start to move towards the point along with the winning neuron in the second phase.

The winning neuron will have the least distance, and euclidean distance is used to calculate the distance between neurons and the point. Each neuron represents each kind of cluster, and the clustering of all the points will happen through the iterations.

One of the main applications Kohonen Neural Network is to recognize the data patterns. It is also used in the medical analysis to classify diseases with higher accuracy. Data are clustered into different categories after analysing the trends in the data.

**5. Recurrent Neural Network(RNN)**

The principle of Recurrent Neural Network is to feedback the output of a layer back to the input again. This principle helps to predict the outcome of the layer. In the Computation process, Each neuron will act as a memory cell. The neuron will retain some information as it goes to the next time step.

It is called a recurrent neural network process. The data to be used later will be remembered and work for the next step will go on in the process. The prediction will improve by error correction. In error correction, some changes are made to create the right prediction output. The learning rate is the rate of how fast the network can make the correct prediction from the wrong prediction.

There is much application of Recurrent Neural Networks, and one of them is the model of converting text to speech. The recurrent neural network was designed for supervised learning without any requirement of teaching signal.

### **6. Long / Short Term Memory**

Schmidhuber and Hochreiter in 1997 built a neural network which is called long short term memory networks (LSTMs). Its main goal is to remember things for a long time in a memory cell that is explicitly defined. Previous values are stored in the memory cell unless told to forget the values by “forget gate”.

New stuff is added through the “input gate” to the memory cell, and it is passed to the next hidden state from the cell along the vectors which is decided by the “output gate”. Composition of primitive music, writing like Shakespeare, or learning complex sequences are some of the applications of LSTMs.

### **7.  Convolutional Neural Network**

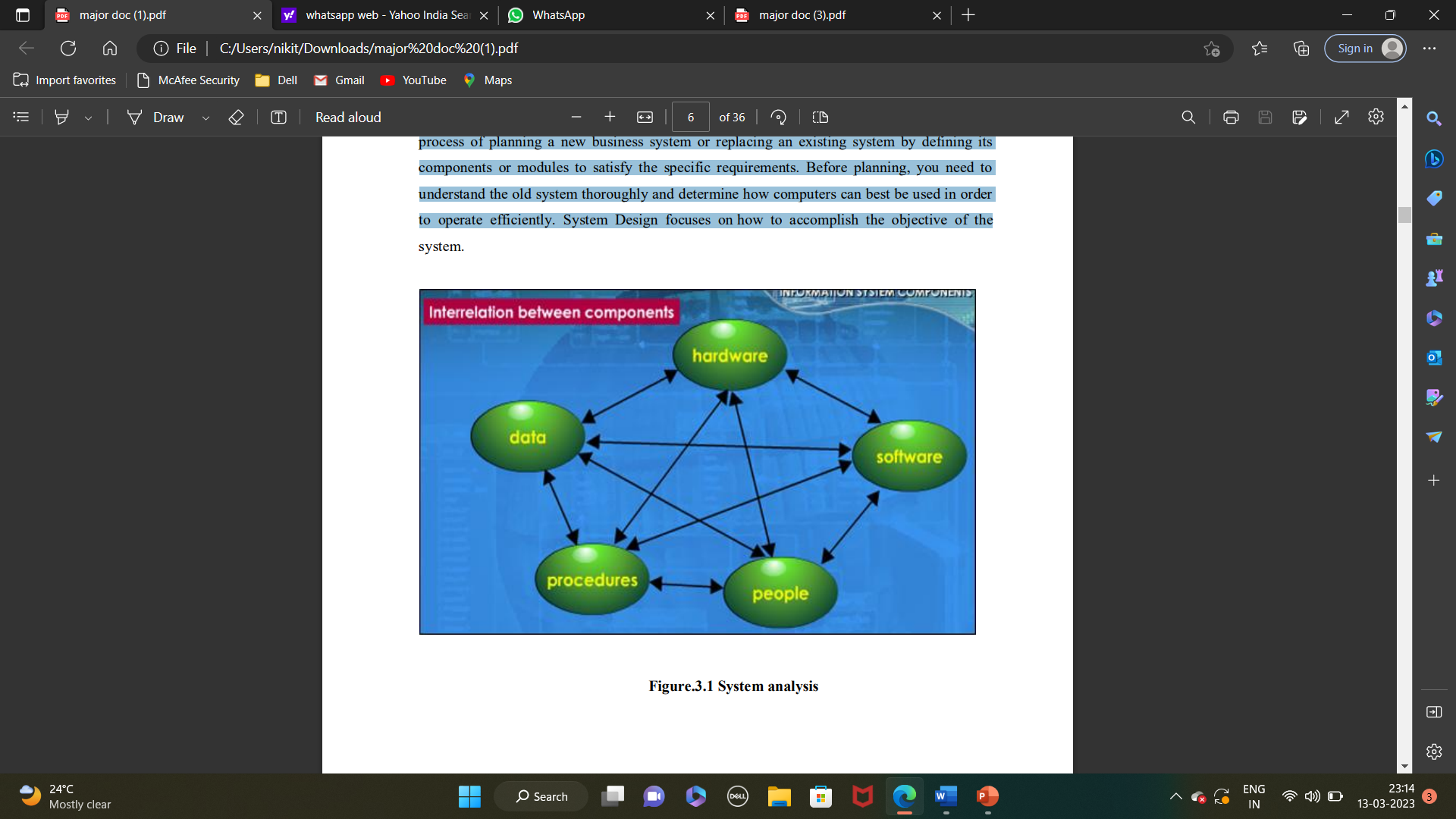
In this type of neural network, Learn-able biases and weights are given to the neurons initially. Image processing and signal processing are some of its applications in the computer vision field. It has taken over OpenCV.

The images are remembered in parts to help the network in computing operations. The photos are recognized by taking the input features batch-wise. In the computing process, image is converted to Grayscale from HSI or RGB scale. The classification of images is done into various categories after the image is transformed. Edges are detected by finding out the pixel value change.

The technique of Image classification and signal processing are used in ConvNet. For image classification, Convolutional Neural Networks have a very high level of accuracy. That is also the reason why convolutional neural networks are dominating the computer vision techniques. Prediction of yield and growth in the future of a land area are other applications of convolutional neural networks in weather and agriculture features.

**5.SYSTEM ANALYSIS**

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do. It is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently. System Design focuses on how to accomplish the objective of the system.

****

**Fig 5.1: System Analysis**

**5.1 SOFTWARE REQUIREMENTS**

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed. Platform – In computing, a platform describes some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer’s architecture, operating system, or programming languages and their runtime libraries. Operating system is one of the first requirements mentioned when defining system requirements (software). Software may not be compatible with different versions of same line of operating systems, although some measure of backward compatibility is often maintained. For example, most software designed for Microsoft Windows XP does not run on Microsoft Windows 98, although the converse is not always true. Similarly, software designed using newer features of Linux Kernel v2.6 generally does not run or compile properly (or at all) on Linux distributions using Kernel v2.2 or v2.4. APIs and drivers – Software making extensive use of special hardware devices, like highend display adapters, needs special API or newer device drivers. A good example is DirectX, which is a collection of APIs for handling tasks related to multimedia, especially game programming, on Microsoft platforms. Web browser – Most web applications and software depending heavily on Internet technologies make use of the default browser installed on system. Microsoft Internet Explorer is a frequent choice of software running on Microsoft Windows, which makes use of ActiveX controls, despite their vulnerabilities.

1)Visual Studio Community Version

2)Nodejs (Version 12.3.1)

3)Python IDEL (Python 3.7)

**5.2 HARDWARE REQUIREMENTS**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements. Architecture – All computer operating systems are designed for a particular computer architecture. Most software applications are limited to particular operating systems running on particular architectures. Although architecture-independent operating systems and applications exist, most need to be recompiled to run on a new architecture. See also a list of common operating systems and their supporting architectures. Processing power – The power of the central processing unit (CPU) is a fundamental system requirement for any software. Most software running on x86 architecture define processing power as the model and the clock speed of the CPU. Many other features of a CPU that influence its speed and power, like bus speed, cache, and MIPS are often ignored. This definition of power is often erroneous, as AMD Athlon and Intel Pentium CPUs at similar clock speed often have different throughput speeds. Intel Pentium CPUs have enjoyed a considerable degree of popularity, and are often mentioned in this category. Memory – All software, when run, resides in the random access memory (RAM) of a computer. Memory requirements are defined after considering demands of the application, operating system, supporting software and files, and other running processes. Optimal performance of other unrelated software running on a multi-tasking computer system is also considered when defining this requirement. Secondary storage – Hard-disk requirements vary, depending on the size of software installation, temporary files created and maintained while installing or running the software, and possible use of swap space (if RAM is insufficient).

**Display adapter**

Software requiring a better than average computer graphics display, like graphics editors and high-end games, often define high-end display adapters in the system requirements. **Peripherals**

Some software applications need to make extensive and/or special use of some peripherals, demanding the higher performance or functionality of such peripherals. Such peripherals include CD-ROM drives, keyboards, pointing devices, network devices, etc.

1) Operating System : Windows Only

2) Processor : i3 and above

3) Ram : 4gb and above

4) Hard Disk : 50 GB

**5.3 FUNCTIONAL REQUIREMENTS**

1.Data Collection

2.Data Pre-processing

3.Training And Testing 4.Modeling

5.Predicting

**5.4 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. Nonfunctional 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.

**6. MODULE DESIGN**

Module Division is the process of dividing collection of source files required in the project into discrete units of functionality. Each module can be independently built, tested and debugged.

**6.1 Data Collection**

Data collection is the process of gathering relevant data and arranging it to create data sets for use in business decision-making, strategic planning and other purposes. It's a crucial part of data analytics applications and research projects. The data is created and edited by its users and moderators. It is important to disclaim that the access to the data was given to the author for an Academic License and to be used exclusively for the present work. Despite being community based, especially for small-to-medium companies, its value is not to question. It has been an invaluable resource for multiple different companies as venture capitals, consulting companies. Dataset contains 925 Rows, each row represent a startup company and 48 columns with attribute.



**Fig: 6.1 Data Collection**

**6.2 Data pre-processing**

Data pre-processing refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. It is the first and crucial step while creating a machine learning model. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model. Pre-processing includes a number of techniques and actions:

• Data cleaning. These techniques, manual and automated, remove data incorrectly added or classified.

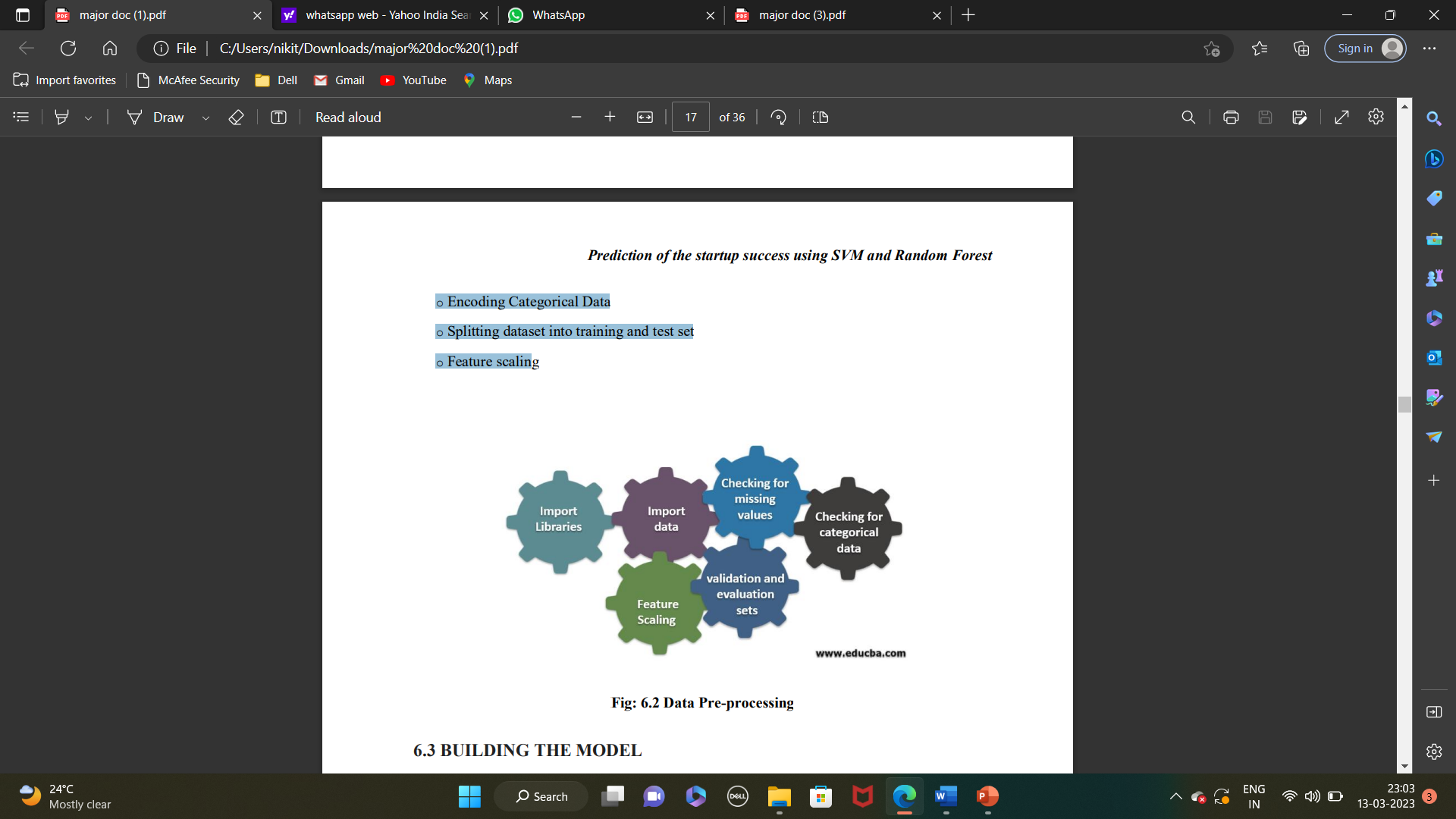
• Data imputations. Most ML frameworks include methods and APIs for balancing or filling in missing data.

• Oversampling. Bias or imbalance in the dataset can be corrected by generating more observations/samples with methods like repetition, bootstrapping and then adding them to the under-represented classes.

• Data integration. Combining multiple datasets to get a large corpus can overcome incompleteness in a single dataset.

• Data normalization. The size of a dataset affects the memory and processing required for iterations during training. Normalization reduces the size by reducing the order and magnitude of data. It involves below steps: o Getting the dataset of Importing libraries of Importing datasets of Finding Missing Data, noisy data, mismatched data.

* Encoding Categorical Data
* Splitting dataset into training and test set
* Feature scaling.

****

**Fig: 6.2 Data Pre-Processing**

**6.3 Model Building**

We use machine learning, namely an artificial neural network to determine what are the chances that Facebook friend request is authentic or not. We also outline the classes and libraries involved. To demonstrate how to build a ANN neural network based image classifier, we shall build a 6 layer neural network that will identify and separate one image from other. To predict image class multiple layers, operate on each other to get best match layer and this process continues till no more improvement left. We utilize Microsoft Excel to store old and new fake data profiles.

The algorithm then stores the data in a data frame. This collection of data will be divided into a training set and a testing set. We would need a data set from the social media sites to train our model. We use a training data set by Facebook or other social networks. This would allow the presented deep learning algorithm to learn the patterns of bot behavior by backpropagation, minimizing the final cost function and adjusting each neuron's weight and bias. In this paper, we outline the classes and libraries involved. We also discuss the sigmoid function and how are the weights determined and used. We also consider the parameters of the social network page which are the most important to our solution.

**6.4 Prediction**

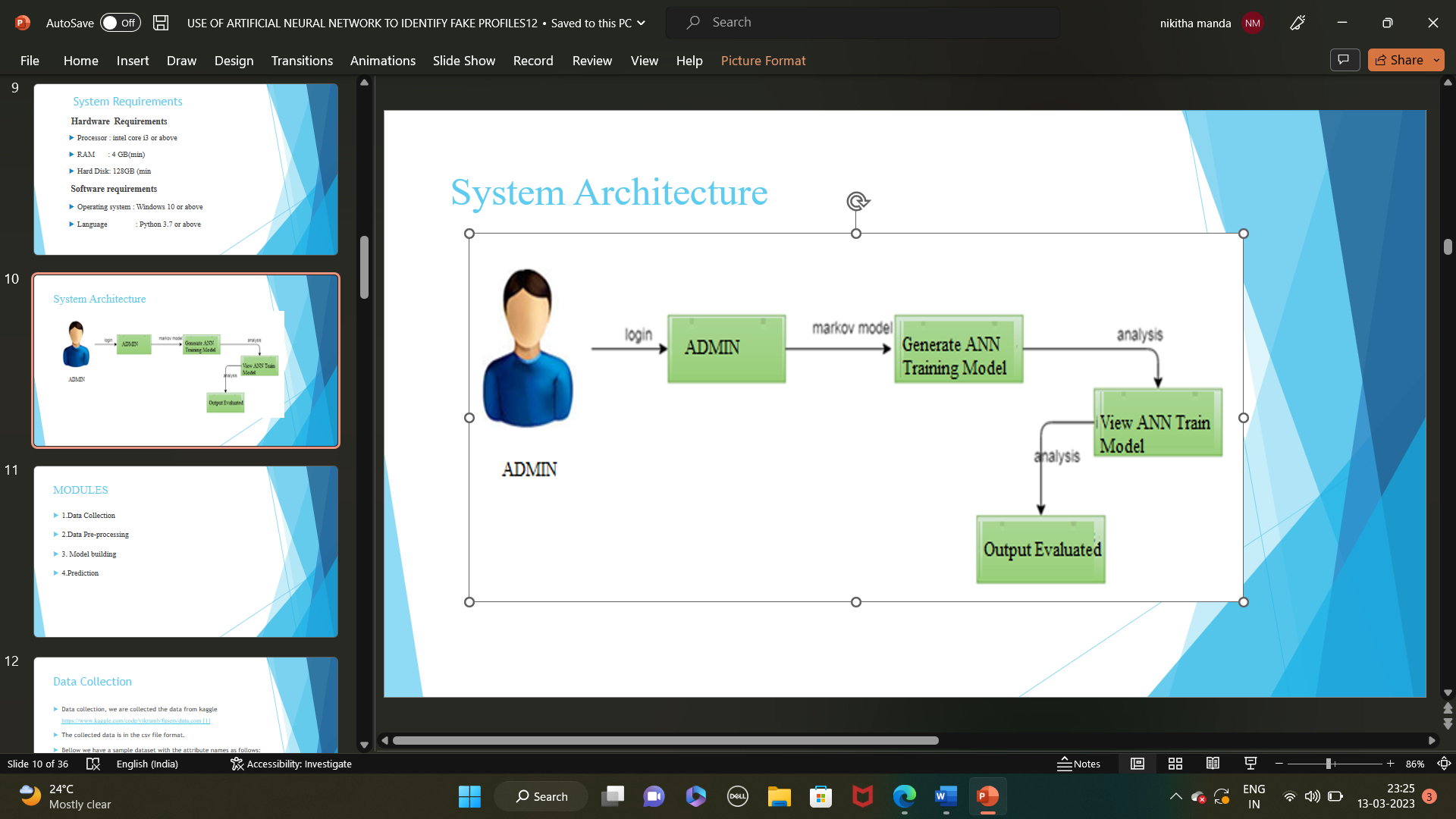
“Prediction” refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome, such as whether or not a customer will churn in 30 days. The algorithm will generate probable values for an unknown variable for each record in the new data, allowing the model builder to identify what that value will most likely be. Machine learning model predictions allow businesses to make highly accurate guesses as to the likely outcomes of a question based on historical data, which can be about all kinds of things- customer churn likelihood, possible fraudulent activity and more. These provide the business with insights that result in tangible business value.

**7.SYSTEM DESIGN**

**7.1 SYSTEM ARCHITECTURE**

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization.

Admin will login to application by using username and password and will upload profile dataset to ANN algorithm to build train model. This train model can be used to predict fake or genuine account by taking new account data, next admin can view all dataset used to train model.



UML

**Fig: 7.1 System Architecture**

**7.2 UML DIAGRAMS**

### **7.2.1 An Overview of UML**

The UML is a language for:

* Visualizing
* Specifying
* Constructing
* Documenting

These are the artifacts of a software system-intensive system.

### **7.2.2 Conceptual Model of UML**

The Three major elements of UML are:

* The UML’s basic building blocks
* The rules that dictate how those building blocks may be put together
* Some common mechanism that applies throughout the UML

### **7.2.3 Basic Building Blocks Of the UML**

The building blocks of UML can be defined as:

* Things: Things are the abstractions that are first-class citizens in model.
* Relationships: Relationships tie these things together.
* Diagrams: Diagrams group the interesting collection of things.

### **7.2.4 Things in the UML**

Things are the most important building blocks of UML. Things can be:

* Structural things
* Behavioural things
* Grouping things
* A notational thing

### **7.2.5 Relationships in the UML**

Relationship is another most important building block of UML.

It shows how elements are associated with each other and this association describes the functionality of an application.

There are four kinds of relationships available.

* Dependency
* Association
* Generalization
* Realization

#### **Dependency**

Dependency is a relationship between two things in which change in one element also affects the other one.

#### **Association**

Association is basically a set of links that connects elements of an UML model. It also describes how many objects are taking part in that relationship.

#### **Generalization**

Generalization can be defined as a relationship which connects a specialized element with generalized element. It basically describes inheritance relationship in the world of objects.

#### **Realization**

Realization can be defined as a relationship in which two elements are connected. One element describes some responsibility which is not implemented and the other one implementsthem. This relationship exists in case of interfaces.

UML diagrams are the ultimate output of the entire discussion. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it a complete one.

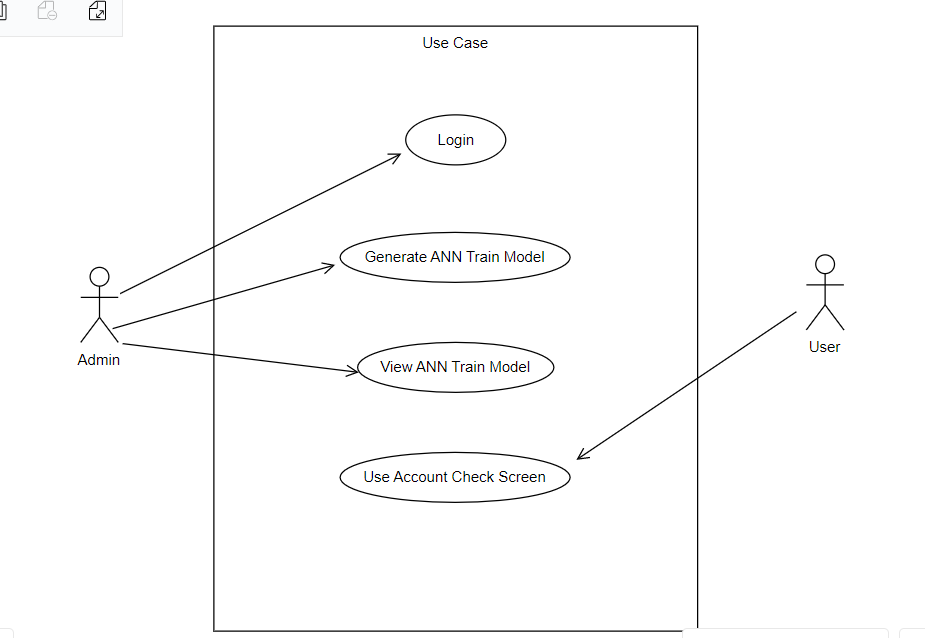
UML includes the following diagrams:

* Class diagram
* Object diagram
* Use case diagram
* Sequence diagram
* Collaboration diagram
* Activity diagram

**7.3 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.

Admin will login to application by using username and password and admin will upload profile dataset to ANN algorithm to build train model, and admin can view all dataset used train ANN model, any user can use this application and enter test data of new account and call ANN algorithm. ANN algorithm will take new test data and applied train model to predict whether given test data contains fake or genuine details.



**FIG 7.3 USE CASE DIAGRAM**

**7.4 Class Diagram**

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code.

It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

We have three classes namely Database , Admin and User .

Admin class contains the attributes of username and password and perform the operations of login,Generate the Ann model and view the trained dataset

User class contains the attributes of Account\_Age, Gender, User\_Age, Link\_Desc, Status\_Count, Friend\_Count, Location, Location\_IP, Status and perform the operation account is fake or genuine.

****

**Fig: 7.4 Class Diagram**

**7.5 Sequence Diagram**

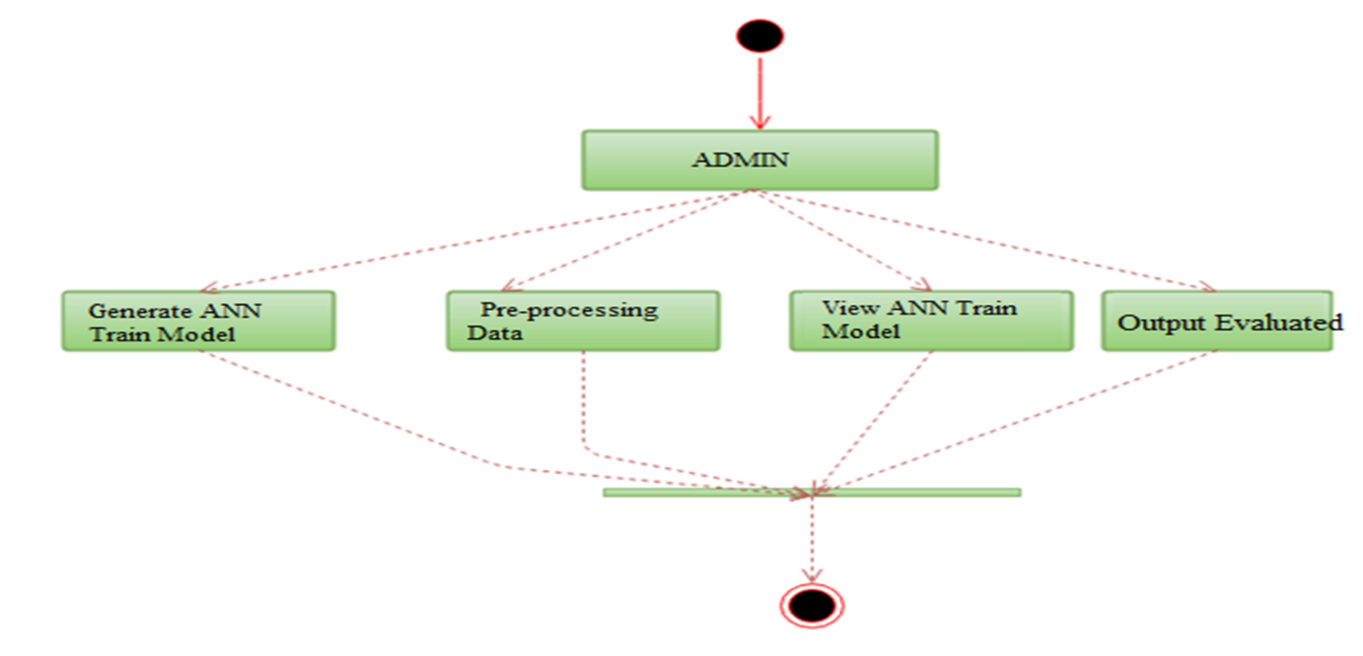
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

Admin will login to application by using username and password and admin will upload profile dataset to ANN algorithm to build train model, and admin can view all dataset used train ANN model.Admin logout from database, any user can use this application and enter test data of new account and call ANN algorithm. ANN algorithm will take new test data and applied train model to predict whether given test data contains fake or genuine details.



**Fig: 7.5 Sequence Diagram**

**7.6 Activity Diagram**



## Fig: 7.5 Activity Diagram

**Fig 7.6 Activity Diagram**

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

Activites we do while identifying a Fake Profile are Admin will login to the application and perform the set of operations are Data preprocessing Generate ANN model and view the Trained dataset. After that output will evaluated that account is a Fake or genuine.

**8. IMPLEMENATION**

**8.1 SOURCE CODE**

***Importing necessary packages***

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponse

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from keras.models import Sequential

from keras.layers.core import Dense,Activation,Dropout

from keras.callbacks import EarlyStopping

from sklearn.preprocessing import OneHotEncoder

from keras.optimizers import Adam

import pickle

classifier = pickle.load(open('model.pkl','rb'))

global model

***// reading data in get method***

def index(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def User(request):

if request.method == 'GET':

return render(request, 'User.html', {})

**// *render user to user.html page***

def Admin(request):

if request.method == 'GET':

return render(request, 'Admin.html', {})

***// admin login***

def AdminLogin(request):

if request.method == 'POST':

username = request.POST.get('username', False)

password = request.POST.get('password', False)

if username == 'admin' and password == 'admin':

context= {'data':'welcome '+username}

return render(request, 'AdminScreen.html', context) ***/\*Admin success full login if details correct \*/***

else:

context= {'data':'login failed'} ***// Admin login failed message if details wrong***

return render(request, 'Admin.html', context)

def importdata():

balance\_data = pd.read\_csv('C:/Users/nikit/OneDrive/Desktop/Use of Artificial Neural Networks to Identify FakeProfiles/Profile/dataset/dataset.txt')

balance\_data = balance\_data.abs()

rows = balance\_data.shape[0] # gives number of row count

cols = balance\_data.shape[1] # gives number of col count

return balance\_data

***SPLITTING DATA SET***

def splitdataset(balance\_data):

X = balance\_data.values[:, 0:8]

y\_= balance\_data.values[:, 8]

y\_ = y\_.reshape(-1, 1)

encoder = OneHotEncoder(sparse=False)

Y = encoder.fit\_transform(y\_)

print(Y)

train\_x, test\_x, train\_y, test\_y = train\_test\_split(X, Y, test\_size=0.2)

return train\_x, test\_x, train\_y, test\_y

def UserCheck(request):

if request.method == 'POST':

data = request.POST.get('t1', False)

input = 'Account\_Age,Gender,User\_Age,Link\_Desc,Status\_Count,Friend\_Count,Location,Location\_IP\n';

input+=data+"\n"

f = open("test.txt", "w")

f.write(input)

f.close()

test = pd.read\_csv('test.txt')

test = test.values[:, 0:8]

print(test)

predict = classifier.predict(test)

print(predict[0])

msg = ''

if str(predict[0]) == '0':

msg = "Given Account Details Predicted As Genuine"

if str(predict[0]) == '1':

msg = "Given Account Details Predicted As Fake"

context= {'data':msg}

return render(request, 'User.html', context)

***GENERATE MODEL***

def GenerateModel(request):

global model

data = importdata()

train\_x, test\_x, train\_y, test\_y = splitdataset(data)

model = Sequential()

model.add(Dense(200, input\_shape=(8,), activation='relu', name='fc1'))

model.add(Dense(200, activation='relu', name='fc2'))

model.add(Dense(2, activation='softmax', name='output'))

optimizer = Adam(lr=0.001)

model.compile(optimizer, loss='categorical\_crossentropy', metrics=['accuracy'])

print('CNN Neural Network Model Summary: ')

print(model.summary())

model.fit(train\_x, train\_y, verbose=2, batch\_size=5, epochs=200)

results = model.evaluate(test\_x, test\_y)

ann\_acc = results[1] \* 100

context= {'data':'ANN Accuracy : '+str(ann\_acc)}

return render(request, 'AdminScreen.html', context)

***VIEWTRAIN MODEL***

def ViewTrain(request):

if request.method == 'GET':

strdata = '<table border=1 align=center width=100%><tr><th><font size=4 color=white>Account Age</th><th><font size=4 color=white>Gender</th><th><font size=4 color=white>User Age</th><th><font size=4 color=white>Link Description</th> <th><font size=4 color=white>Status Count</th><th><font size=4 color=white>Friend Count</th><th><font size=4 color=white>Location</th><th><font size=4 color=white>Location IP</th><th><font size=4 color=white>Profile Status</th></tr><tr>'

data = pd.read\_csv('C:/Users/nikit/OneDrive/Desktop/Use of Artificial Neural Networks to Identify FakeProfiles/Profile/dataset/dataset.txt')

rows = data.shape[0] # gives number of row count

cols = data.shape[1] # gives number of col count

for i in range(rows):

for j in range(cols):

strdata+='<td><font size=3 color=white>'+str(data.iloc[i,j])+'</font></td>'

strdata+='</tr><tr>'

context= {'data':strdata} return render(request, 'ViewData.html', context)

**9. OUTPUT SCREENS**

All fake users main intention is to send friend request to normal users to hack their machine or to steal their data and never they will have many number of posts or have many following friends and their account age also will have less number of years. By analysing this features Facebook will mark whether user profile is fake or genuine.

**Account\_Age, Gender, User\_Age, Link\_Desc, Status\_Count, Friend\_Count, Location, Location\_IP, Status**

10, 1, 22, 0, 1073, 237, 0, 0, 0

10, 0, 33, 0, 127, 152, 0, 0, 0

10, 1, 46, 0, 1601, 405, 0, 0, 0

10, 0, 25, 0, 704, 380, 0, 0, 0

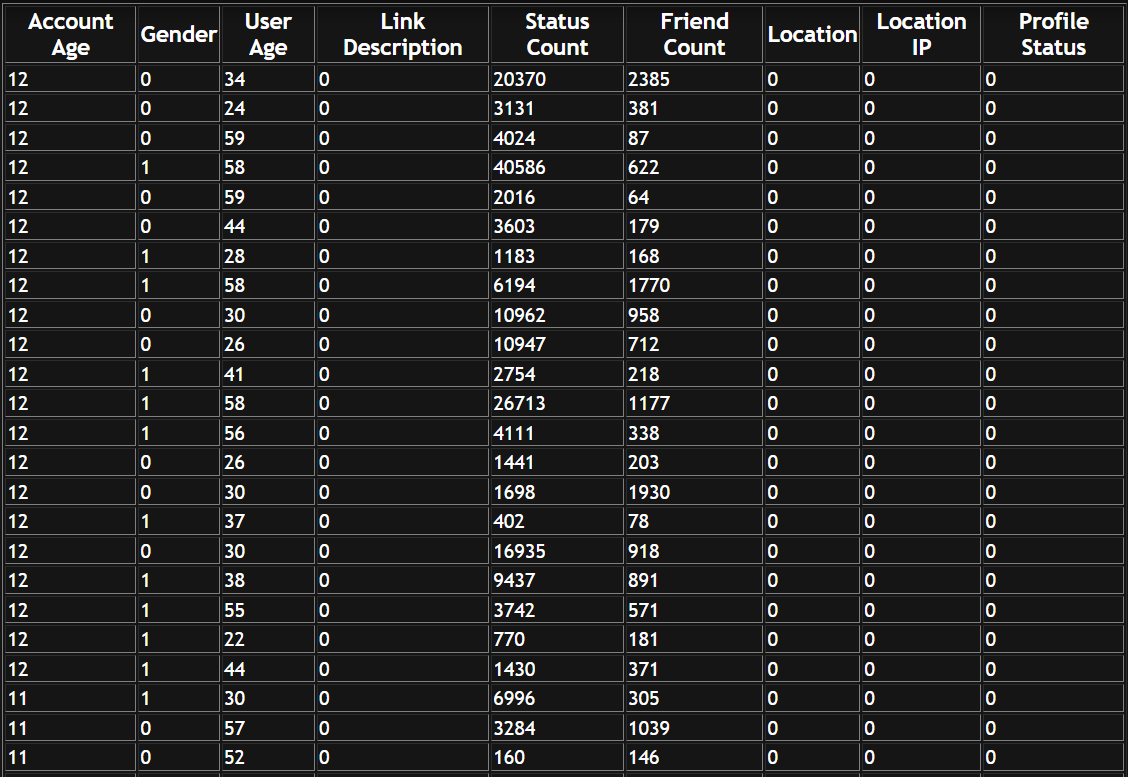
7, 1, 34, 1, 64, 721, 1, 1, 1

7, 1, 30, 1, 69, 587, 1, 1, 1

7, 1, 36, 1, 61, 782, 1, 1, 1

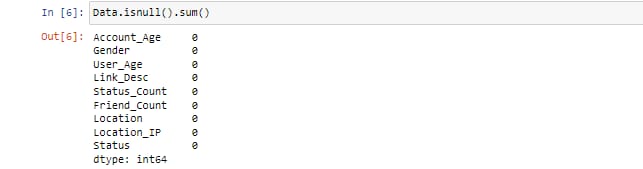
7, 1, 52, 1, 96, 827, 1, 1, 1

In bellow dataset all bold names are the dataset column names and all integer values are the dataset values. In bellow dataset last column give us information of fake or genuine account if last column contains value 0 then account is genuine otherwise fake.

****

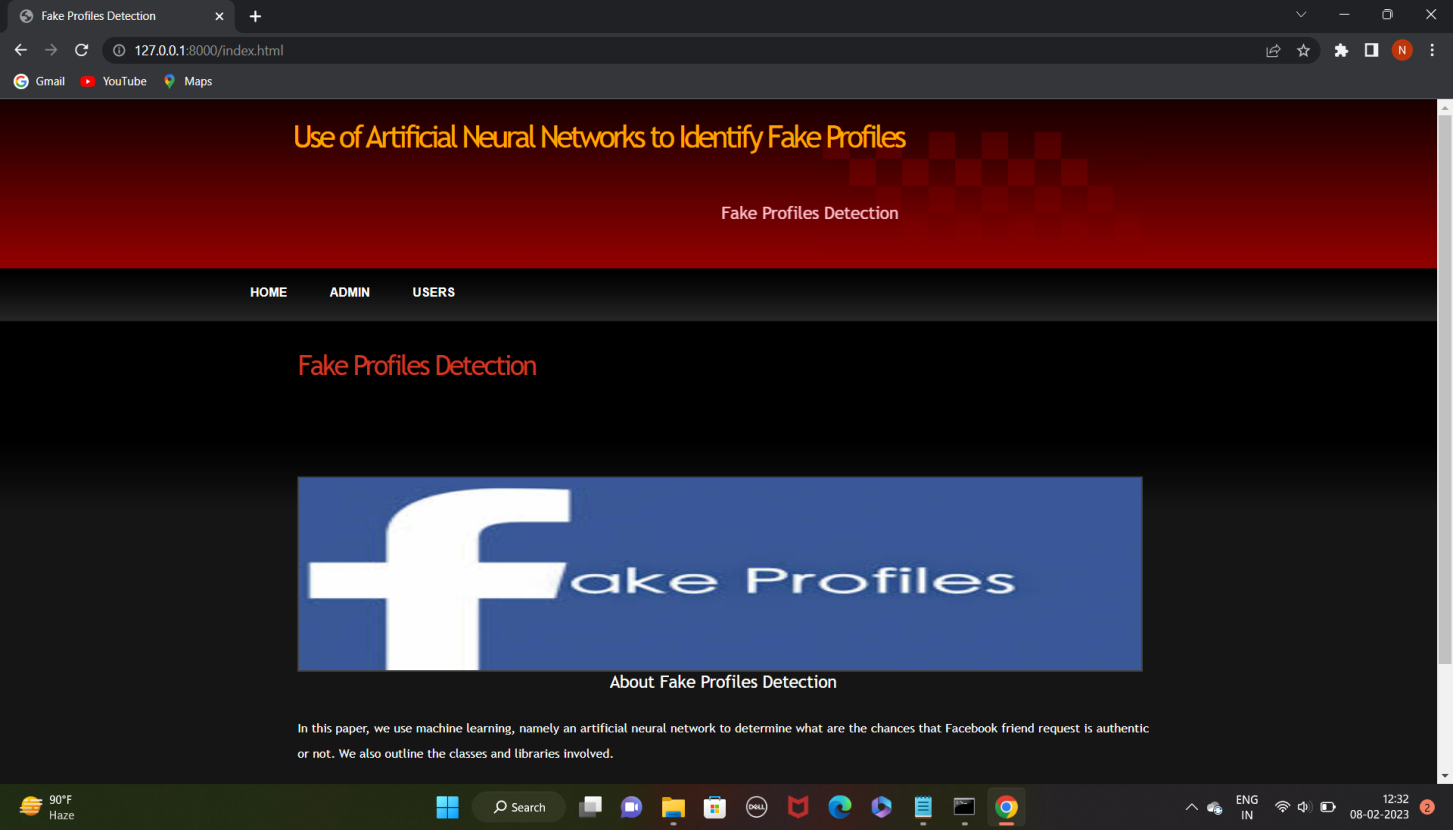
**FIG9.1:DATA COLLECTION**

Data pre-processing refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. In this step we check for null values in dataset. In our collected dataset,there is no null values. If any null values occurred fill it with Mean,Mode or Median.



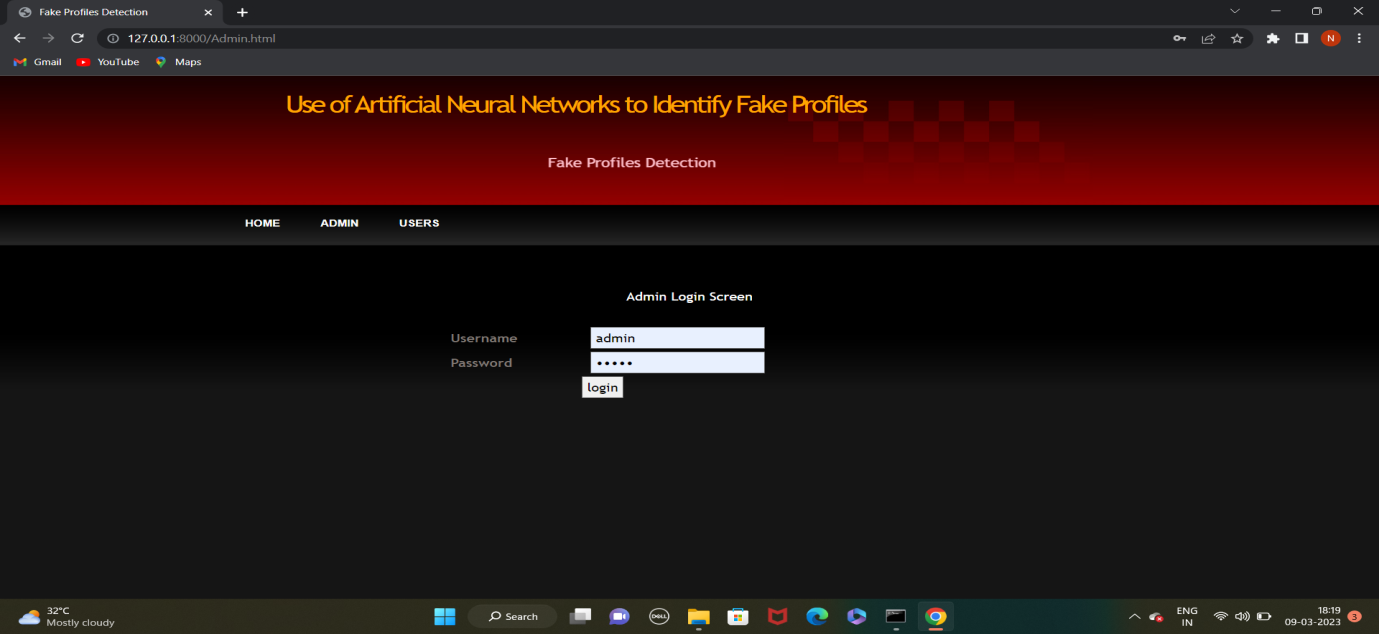
**FIG9.2:DATA PREPROCESSING**

Deploy this application on DJANGO server and then run in browser enter URL as ‘http://localhost:8000/index.html’ to get below screen

****

**FIG 9.3:HOME SCREEN**

In above screen click on ‘ADMIN’ link to get below login screen

****

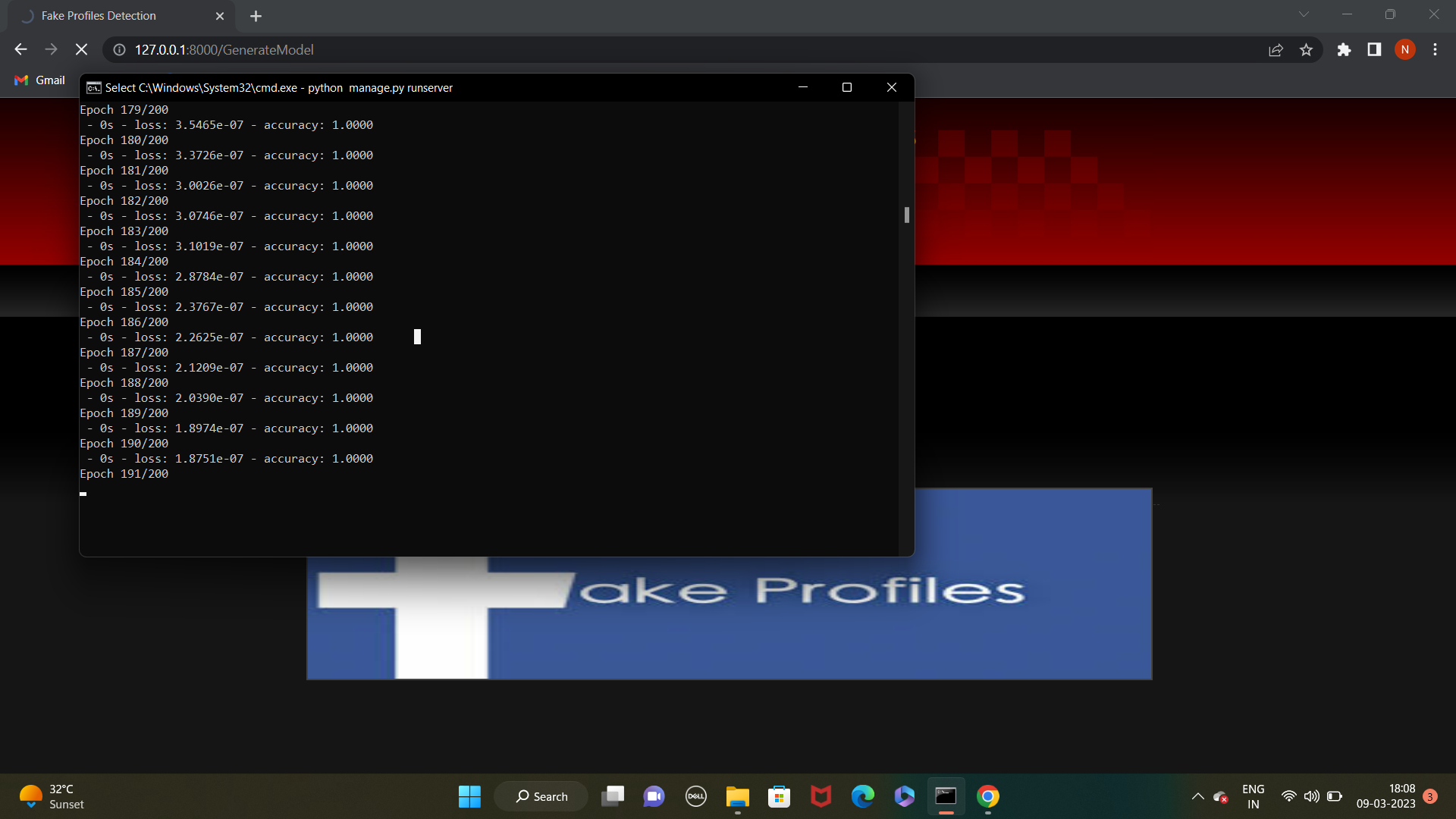
**FIG 9.4:ADMIN PAGE**

In above screen enter admin and admin as username and password to login as admin. After login will get below screen



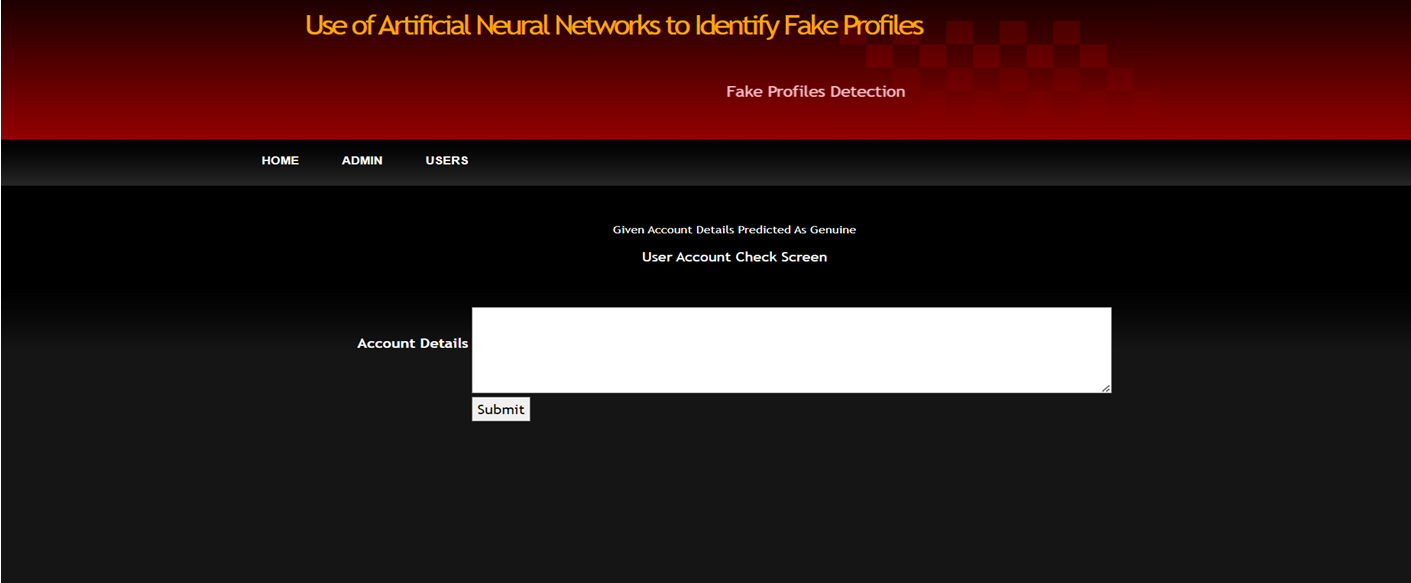
**FIG 9.5: ADMIN HOME**

In above screen click on ‘Generate ANN Train Model’ to generate training model on dataset. After clicking on that link you can see server console to check ANN processing details with accuracy



**FIG 9.6 TRAIN ANN MODEL**

In above black console we can see all ANN details.



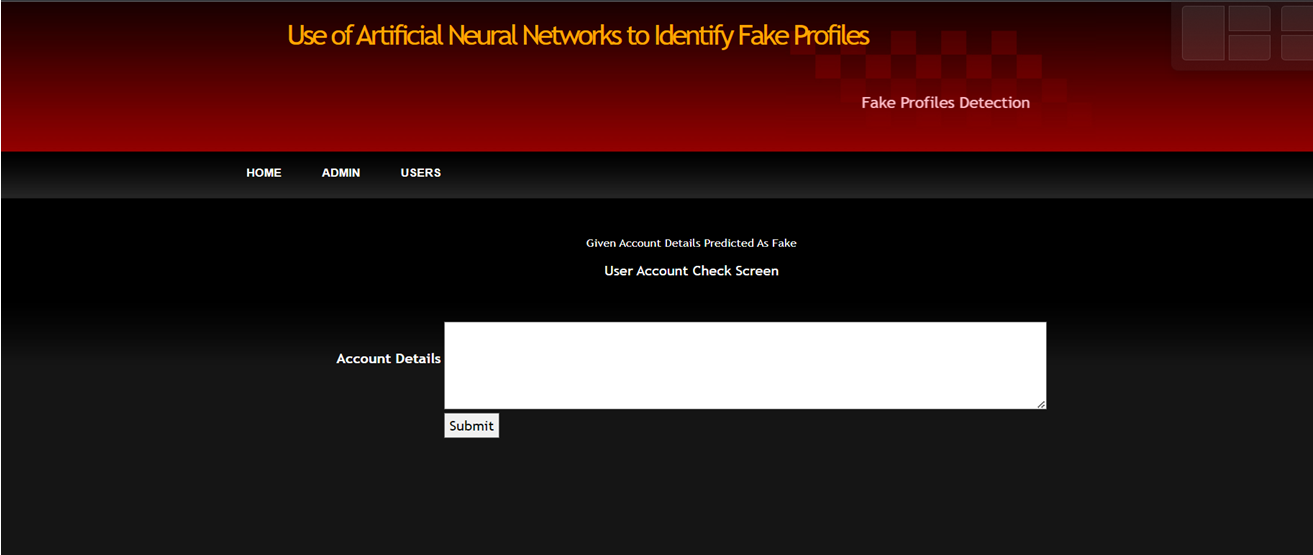
**FIG 9.7: USER SCREEN**

In above screen enter some test account details to get prediction/identification from ANN. You can use below records to check 10, 1, 44, 0, 280, 1273, 0, 0

10, 0, 54, 0, 5237, 241, 0, 0

7, 0, 42, 1, 57, 631, 1, 1

7, 1, 56, 1, 66, 623, 1, 1

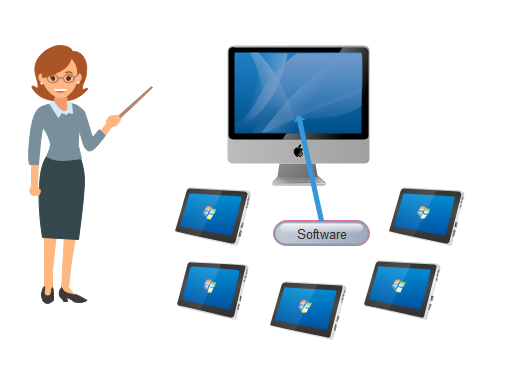


**FIG 9.8: DETECTING FAKE ACOUNT**

In above screen we got result as fake for given account data

**10.SYSTEM TESTING**

System Testing includes testing of a fully integrated software system. Generally, a computer system is made with the integration of software (any software is only a single element of a computer system). The software is developed in units and then interfaced with other software and hardware to create a complete computer system. In other words, a computer system consists of a group of software to perform the various tasks, but only software cannot perform the task; for that software must be interfaced with compatible hardware. System testing is a series of different type of tests with the purpose to exercise and examine the full working of an integrated software computer system against requirements.



**FIG10.1 SYSTEM TESTING**

To check the end-to-end flow of an application or the software as a user is known as System testing. In this, we navigate (go through) all the necessary modules of an application and check if the end features or the end business works fine, and test the product as a whole system.

It is end-to-end testing where the testing environment is similar to the production environment.

**10.1 SOFTWARE TESTING STRATEGIES**

Software testing strategy is the planning done before testing commences and exercised systematically to test the software. The testing strategy could be developed by the project manager, or by the software engineers or it could even be a testing specialist.

Developing a testing strategy for software is important because if testing is not conducted properly it would lead to wastage of time and effort and it would even be the case that some error or bugs remain undetected. Some general **characteristics** that should be considered while developing the testing strategy

**10.1.1 STATIC TESTING**

Static testing is a software testing method that examines a program -- along with any associated documents -- but does not require the program to be executed. Dynamic testing, the other main [category of software testing](https://www.techtarget.com/searchsoftwarequality/tip/Static-and-dynamic-code-analysis-Complementary-techniques), requires testers to interact with the program while it runs. The two methods are frequently used together to ensure the basic functionalities of a program.

Instead of executing the code, static testing is a process of checking the code and designing documents and requirements before it's run to find errors. The main goal is to find flaws in the early stages of development because it is normally easier to find the sources of possible failures this way.

### **What is subject to static testing?**

It's common for code, design documents and requirements to be static tested before the software is run to find errors. Anything that relates to functional requirements can also be checked. More specifically, the process will involve reviewing written materials that provide a wider view of the tested software application as a whole. Some examples of what's tested include the following:

* [requirement specifications](https://www.techtarget.com/searchsoftwarequality/answer/What-are-requirements-types)
* design documents
* user documents
* webpage content
* [source code](https://www.techtarget.com/searchapparchitecture/definition/source-code)
* test cases, test data and test [scripts](https://www.techtarget.com/whatis/definition/script)
* specification and [matrix](https://www.techtarget.com/whatis/definition/matrix) documents

### **Benefits of static testing**

Some benefits of static testing include the following:

* Early detection and correction of any coding errors.
* Reduced cost in the early stages of development based on the amount of rework needed to fix any errors.
* Reduced time scales for development.
* Feedback received at this stage will help improve the overall software function. Once other testing types such as dynamic testing start, there won't be as many errors found. This means the code has become more maintainable.
* This process will also help give developers a better idea of the quality issues found in the software.
* Automated tools can expedite the code and document review process.
* Static testing can also boost the amount of communication between teams.

**10.1.2 STRUCTURAL TESTING**

Another type of [**software testing**](https://www.javatpoint.com/software-testing-tutorial) technique is **Structural testing**, which is used to test the internal design of the software or structure of the coding for the particular software.

In this testing, the development team members are included in the testing team to execute the software's internal design. The working of structural testing is opposite to **Behavioral testing.**

In other words, we can say that structural testing tests the different features of an application based on its types.

Structural testing is also known as white-box testing, **glass box testing**, and **clear-box testing**. Developers mostly implement it to identify the issue and fix them quickly.

The structural testing process requires an in-depth knowledge of the programming language and is opposite to Functional Testing.

The knowledge of the code's internal executions and how the software is implemented is a necessity for the test engineer to implement the structural testing.

Throughout the structural testing, the test engineer intends on how the software performs, and it can be used at all levels of testing.

**For example**, the coverage of menu options or significant business transactions could be the system's structural element or acceptance testing.

## Types of Structural Testing

Structural testing is divided into four different categories, which are as follows:

* **Mutation testing**
* **Data flow testing**
* **Control flow testing**
* **Slice-based testing**

**Mutation testing**

* It is used to check the quality of the test case that should fail the mutant code.
* Mutation testing involves the development of new tests to be implemented on the software for its testing process.
* When we identify various errors, it implies that either the program is correct or the test case is inefficient in locating the fault.
* In the mutation testing, the developers make small modifications to the previously accessible software tests and generate a mutant of the old software test.
* It used to cause an error in the program, which implies that the mutation testing is performed to evaluate the test case's productivity.

### **Data flow testing**

* It is a group of testing approaches used to observe the control flow of programs to discover the sequence of variables as per the series of events.
* It implements a control flow graph and analysis the points where the codes can change the data.
* If we execute the data flow testing technique, the information is kept safe and unchanged during the code's implementation.

### **Control flow testing**

* The **control flow testing** is the basic model of **Structural testing**.
* It is to check the implementation order of commands or statements of the code over a control structure.
* In the control flow testing, a specific part of an extensive program is selected by the test engineer to set the testing path.
* Generally, the control flow testing technique is used in unit testing.
* In this testing, the entire test is based on how the control is executed during the code.
* The complete information of all the software's features and logic is necessary to execute the control flow testing.

### **Slice-based testing**

* It was initially created and established to keep the software.
* The basic idea is to sort the complete code into small chunks and then evaluate each portion carefully.
* The slice-based testing is very beneficial for the maintenance of the software along with fixing the application too.

**10.1.3.BEHAVIORAL TESTING**

**Behavior testing** in software testing is an example of a technique we can use to validate certain behaviors of a software program instead of the technical perspective, under various circumstances.Behavior testing helps determine how the system must behave externally.We perform behavior testing without knowing the internal contents of the application. It effectively knows how the system must respond when specific inputs are provided to the software under testing.

**10.1.3.1 BLACK BOX TESTING**

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer.

In this method, tester selects a function and gives input value to examine its functionality, and checks whether the function is giving expected output or not. If the function produces correct output, then it is passed in testing, otherwise failed. The test team reports the result to the development team and then tests the next function. After completing testing of all functions if there are severe problems, then it is given back to the development team for correction.

Black box testing

**FIG10.1.3.1 BLACK BOX TESTING**

## Generic steps of black box testing

* The black box test is based on the specification of requirements, so it is examined in the beginning.
* In the second step, the tester creates a positive test scenario and an adverse test scenario by selecting valid and invalid input values to check that the software is processing them correctly or incorrectly.
* In the third step, the tester develops various test cases such as decision table, all pairs test, equivalent division, error estimation, cause-effect graph, etc.
* The fourth phase includes the execution of all test cases.
* In the fifth step, the tester compares the expected output against the actual output.
* In the sixth and final step, if there is any flaw in the software, then it is cured and tested again.

## 10.1.3.2 Grey Box Testing

White box testing involves complete knowledge of the inner workings of a system under test and black box involves no knowledge. [Grey box testing](https://www.imperva.com/learn/application-security/gray-box-testing/), however, is a compromise – testing a system with partial knowledge of its internals. It is most commonly used in integration testing, end-to-end system testing, and [penetration testing](https://www.imperva.com/learn/application-security/penetration-testing/).Grey box testing combines inputs from developers and testers and can result in more effective testing strategies. It reduces the overhead required to perform functional testing of a large number of user paths, focusing testers on the paths most likely to affect users or result in a defect.Grey box testing combines the benefits of black box and white box testing:

* Ensuring that tests are performed from the user’s perspective, like in black box testing.
* Leveraging inside knowledge to focus on the problems that matter most, and to identify and resolve internal weaknesses of the system, like in white box testing.

In the world of [Application Security Testing](https://www.imperva.com/learn/application-security/application-security/), the grey box testing approach is called Interactive Application Security Testing (IAST). IAST combines:

* **SAST** — which performs white box testing by evaluating static application code.
* [**Dynamic Application Security Testing**](https://www.imperva.com/learn/application-security/sast-iast-dast/)**(DAST)** — which performs black box testing, by interacting with running applications and discovering faults and vulnerabilities like a user or external attacker would.

**10.1.3.3 White Box Testing**

White box testing is a technique that uses a program’s internal or source code to design different test cases to check the quality of the program. In this technique, the internal structure and implementation of how an application works are known to the tester.

You may understand this by a simple example. Let’s assume that there is a car which is not working and therefore you take it to a mechanic to get it fixed. Now the mechanic will examine why the car is not working. Similarly, a tester studies the code of an application and determines all the inputs, and verifies the outputs against desired outcomes.

**What do you verify in White Box Testing?**

Following are the steps that are taken into consideration while performing white box testing:

* Verification of security holes in source code
* Testing of any broken or incomplete path
* To verify the flow of structure as mentioned in the software requirement document

**10.2 TEST CASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO.** | **TSETCASE** | **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** | **RESULTS** |
| **1** | Admin sign up | Admin enters valid credentials for signup | Admin account is successfully created | Admin account is successfully created | Pass |
| **2** | Admin login | Admin enters valid credentials for login | Admin is successfully logged in | Admin is successfully logged in | Pass |
| **3** | Generate ANN | Admin clicks on the "Generate ANN" button to create a neural network | The neural network is successfully generated | The neural network is successfully generated | Pass |
| **4** | Logout | Admin clicks on the "Logout" button to exit the system | Admin is successfully logged out | Admin is successfully logged out | Pass |
| **5** | User Input | A genuine user profile with real personal information is entered into the system by the user | The neural network correctly identifies the profile as real | The neural network correctly identifies the profile as real | Pass |
| **6** | User give mixed input | A mix of genuine and fake user profiles is entered into the system by the user PROFILE” | The neural network correctly identifies most of the fake profiles | The neural network correctly identifies most of the fake profiles | Pass |

**Fig10.2 TEST CASES TO IDENTIFY FAKE PROFILE**

**11. CONCLUSION**

We use machine learning, namely an artificial neural network to determine what are the chances that a friend request is authentic are or not. Each equation at each neuron (node) is put through a Sigmoid function. We use a training data set by Facebook or other social networks. This would allow the presented deep learning algorithm to learn the patterns of bot behavior by back propagation, minimizing the final cost function and adjusting each neuron's weight and bias.

**12.FUTURE ENHANCEMENT**

There seems to be a newsworthy issue involving social media networks getting hacked every day. Recently, Facebook had a data breach which affected about 50 million users . Facebook provides a set of clearly defined provisions that explain what they do with the user's data [4]. The policy does very little to prevent the constant exploitation of security and privacy. Fake profiles seem to slip through Facebook's built-in security features

**13.REFERENCES**

[1] https://www.statista.com/topics/1164/socialnetworks/

[2] https://www.cnbc.com/2018/01/31/facebookearnings-q4-2017- arpu.html

[3] https://www.cnet.com/news/facebookbreach-affected-50-millionpeople/

[4] <https://www.facebook.com/policy.php>

[5] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. 2012. Aiding the detection of fake accounts in large scale social online services. In Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation (NSDI'12). USENIX Association, Berkeley, CA, USA, 15- 15.

[6] Akshay J. Sarode and Arun Mishra. 2015. Audit and Analysis of Impostors: An experimental approach to detect fake profile in an online social network. In Proceedings of the Sixth International Conference on Computer and Communication Technology 2015 (ICCCT '15). ACM, New York, NY, USA, 1-8. DOI: <https://doi.org/10.1145/2818567.2818568>

[7] Devakunchari Ramalingam, Valliyammai Chinnaiah. Fake profile detection techniques in large-scale online social networks: A comprehensive review. Computers & Electrical Engineering, Volume 65, 2018, Pages 165-177, ISSN 0045-7906, [https://doi.org/10.1016/j.compeleceng.2017.05.0 20](https://doi.org/10.1016/j.compeleceng.2017.05.0%2020).

[8] https://www.enigmasoftware.com/top-20- countries-the-most-cybercrime

[9] pages.cs.wisc.edu/~bolo/shipyard/neural/local.ht ml

[10] https://stackoverflow.com/questions/40758562/c an-anyone-explain-mestandardscaler

[11] <https://pandas.pydata.org>

[12] [https://www.tutorialspoint.com/python\_pandas/i ndex.html](https://www.tutorialspoint.com/python_pandas/i%20ndex.html)

[13] 1. For our project we have collected the data from kaggle with the following <https://www.kaggle.com/code/vikramb/fusers/data.com>

[14]. https://www.statista.com/topics/1164/socialnetworks/

[15]. [https://www.cnbc.com/2018/01/31/facebookearnings-q4-2017- arpu.html](https://www.cnbc.com/2018/01/31/facebookearnings-q4-2017-%20arpu.html)