

A VERSATILE TOOL FOR PROTECTING USER FROM FRAUDSTERS AND SCAMMERS

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

**BATCHU SURENDRA (40111292)
PENDEM SAI RAM (40110934)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

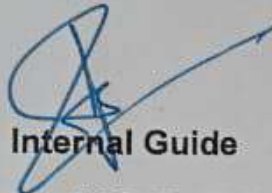
Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **PENDEM SAIRAM** (40110934) and **BATCHU SURENDRA** (40111292) who carried out the Project entitled "A VERSATILE TOOL FOR PROTECTING USER FROM FRADUSTERS AND SCAMMERS" under my supervision from November 2023 to April 2024.



Internal Guide

Dr. S. Jayanthi B.E., M.E., Ph.D.

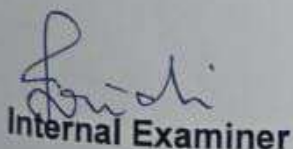


Head of the department

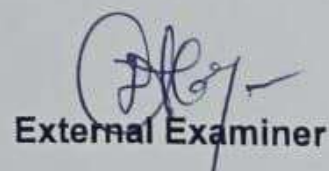
Dr. L. Lakshmanan, M.E., Ph.D.

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

Submitted for viva voce Examination held on 13/4/2024



Internal Examiner



External Examiner

DECLARATION

I, PENDEM SAIRAM(40110934), hereby declare that the Project Report entitled "A VERSATILE TOOL FOR PROTECTING USER FROM FRUADSTES AND SCAMMERS" done by me under the guidance of Dr. S. Jayanthi B.E., M.E., Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 13/4/24

PLACE: CHENNAI

P. Sairam.

SIGNATURE OF CANDIDATE

ACKNOWLEDGMENT

I am pleased to acknowledge my sincere thanks to **The Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D., Dean** , School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S.Jayanthi B.E., M.E., Ph.D.**, for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

In the present generation, the social life of everyone has become associated with the online social networks. These sites have made a drastic change in the way we pursue our social life. Making friends and keeping in contact with them and their updates has become easier. But with their rapid growth, many problems like fake profiles, online impersonation have also grown. There is no feasible solution exist to control these problems. In this project, we came up with a framework with which automatic detection of fake profiles is possible and is efficient. This framework uses classification techniques like Random Forest to classify the profiles into fake or genuine classes. As, this is an automatic detection method, it can be applied easily by online social networks which has millions of profiles whose profiles cannot be examined manual Spotting Fakes on Instagram - A Machine Learning Approach Fake accounts plague Instagram, eroding user trust and enabling scams. This project proposes a solution – a system powered by machine learning to detect these deceptive profiles. We'll go beyond just follower ratios. We'll analyze profile pictures and posts for inconsistencies that might indicate automated creation. We'll also use Natural Language Processing (NLP) to examine captions and bios, looking for unusual language or generic content often employed by bots. Additionally, factors like account age, posting habits, and sudden activity spikes will be considered. These features will be used to train a machine learning model that can accurately identify fake profiles. We'll explore different algorithms like Random Forests or Support Vector Machines. The key to success lies in the training data. We'll utilize existing datasets of labeled Instagram profiles, potentially partnering with Instagram's security teams for additional data. By equipping users with the ability to identify real profiles from fakes, this system aims to create a safer and more trustworthy environment on Instagram, ultimately enhancing the overall user experience.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
	1.1 ORIGIN OF THE PROBLEM	1
	1.2 BASIC DEFINITIONS AND BACKGROUND	1
	1.3 PROBLEM STATEMENTS	8
	1.4 APPLICATIONS	9
2	LITERATURE SURVEY	10
	2.1 DESCRIPTION OF EXISTING SYSTEMS	10
3	REQUIREMENT ANALYSIS	14
	3.1 INTRODUCTION	14
	3.2 PROCESS BASED MODEL USED WITH JUSTIFICATION	15
	3.2.1 SDLC	15
	3.3 STAGES IN SDLC	15
	3.4 REQUIREMENTS GATHERING STAGE	16
	3.5 ANALYSIS STAGE	17
	3.6 DESIGNING STAGE	18
	3.7 DEVELOPMENT STAGE	19
	3.8 INTEGRATION AND TESTING STAGE	20
	3.9 INSTALLATION AND ACCEPTANCE TEST	22

3.10	SOFTWARE REQUIREMENT SPECIFICATION	23
4	DESCRIPTION OF PROPOSED SYSTEM	26
4.1	DESIGN METHODOLOGY	27
4.2	DESIGN METHODOLOGY	28
4.3	CLASS DIAGRAM	28
4.4	SEQUENCE DIAGRAM	29
4.5	ACTIVITY DIAGRAM	30
5	IMPLEMENTATION DETAILS	32
5.1	SYSTEM ARCHITECTURE	32
5.2	DESCRIPTION OF DATA SETS AND TOOLS	42
6	RESULTS AND DISCUSSION	44
6.1	STEPWISE DESCRIPTION OF RESULTS	44
6.2	TEST CASE RESULTS	45
7	CONCLUSION AND FUTURE WORK	47
7.1	CONCLUSION	47
7.2	FUTURE STUDY	47
	REFERENCES	48
	APPENDIX	50
A.	SOURCE CODE	50
B.	SCREENSHOTS	56
C.	RESEARCH PAPER	60
D.	CERTIFICATE	70

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1	Multi class SVM	1
2	Confusion Matrix	6
3	SDLC	15
4	Requirements Gathering Stage	16
5	Analysis Stage	18
6	Design Stage	19
7	Development Stage	20
8	Integration and Test Stage	21
9	Installation and Acceptance Test	22
10	UML Representation	27
11	Use Case Diagram	28
12	Class Diagram	29
13	Sequence Diagram	30
14	Activity Diagram	31
15	System Architecture	32
16	Hyperplane	33
17	SVM Tree	37
18	Decision Tree	39
19	Confusion Matrix	45
20	Learning Curve	46
21	Screen Shots	46

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
1	CONFUSION MATRIX	6
2	DATASET	43

LIST OF ABBREVIATIONS

S NO	ABBREVIATIONS	EXPANSION
1	SVM	Support Vector Machine
2	UML	Unified Modeling Language
3	RF	Random Forest
4	API	Application Programming Interface
5	CSV	Comma Separated Values
6	ANN	Artificial Neural Network
7	CNN	Convolutional Neural Network
8	CART	A Classification And Regression Tree
9	SDLC	Software Development Life Cycle

CHAPTER 1

INTRODUCTION

1.1 ORIGIN OF THE PROBLEM:

The use of social networks such as Facebook, Twitter, Google, Instagram, and LinkedIn is on the rise. Individuals and organizations use social networks to express their views, advertise their products, and express future policies of their companies and organizations. By expanding the use of social networks, malicious users seek to violate the privacy of other users and abuse their concern for users. Hence, social networks providers are trying to detect malicious users and fake accounts in order to eliminate them from social networking environments. So, we decided to develop a driver assistance system by implementing different applications.

1.2 BASIC DEFINITIONS AND BACKGROUND:

Support vector Machine

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms classification in case of **multi class SVM**.

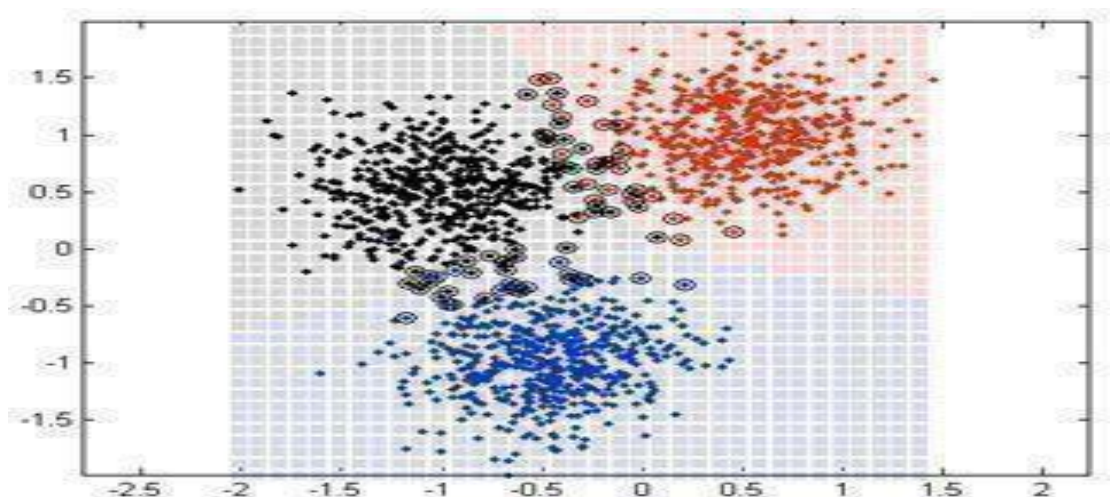


Fig-1: Multi class SVM

Whereas the original problem may be stated in a finite-dimensional space, it happens the sets to discriminate are not linearly separable in space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products of pairs of input data vectors may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant, where such a set of vectors is an orthogonal (and thus minimal) set of vectors that defines a hyperplane.

The vectors defining the hyperplanes can be chosen to be linear combinations with parameters images of feature vectors that occur in the data base. With this choice of a hyperplane, the pinpoint that it becomes small as grows further away from, each term in the sum measures the degree of closeness of the test point to the corresponding data base point. In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the fact that the set of points mapped into any hyperplane can be quite convoluted as a result, allowing much more complex discrimination between sets that are not convex at all in original space.

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below these are two different

categories that are classified using a decision boundary or hyperplane: Support Vectors – Data points that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.

RANDOM FOREST

Random Forest is one of the most used machine learning algorithms, because its simplicity and the fact that it can be used for both classification and regression tasks. It's a supervised learning algorithm. As it's seen from its name, it's bagged decision tree models that split on a subset of features on each split, it creates a forest and makes it somehow random.

The «forest» it builds, is an ensemble of Decision Trees, most of the time trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. To say it in simple words: Random Forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. For better understanding of the RF algorithm is necessary to explain what the main idea behind decision trees is. Depending on the features in each dataset, the decision tree model learns a series of questions to figure out the class labels of the instances. What makes this model successful is that it is non-parametric and it can handle heterogeneous data (ordered or categorical variables, or a mix of both).

Furthermore decision trees fundamentally implement feature selection, making them at least to some extent robust to irrelevant or noisy variables and are robust to outliers or errors in labels. To summarize, here is the steps that Random Forest algorithm follows:

- Randomly choose n samples from the training set with replacement.
- Grow a decision tree from the n sample. At each node.
- Repeat the steps 1 to 2 k -times.

- Aggregate the prediction by each tree to assign the class majority
- Random Forest, as the name implies, is a collection of trees-based models trained on random subsets of the training data. Being an ensemble model, the primary benefit of a random forest model is the reduced variance compared to training a single tree. Since each tree in the ensemble is trained on a random subset of the overall training set, the ensemble as a whole is less likely to overfit the training set. Random forest-based classifiers are some of the most accurate models in many classification challenges.

DECISION TREE

Decision-making for the product is critical, as are recommendations for consolidation or creation of a new product [14]. The decision tree is then created and reviewed to ensure that all results are weighed. The tree is evaluated in the second level of decision tree analysis. This is accomplished by weighing the solutions that are most applicable to the situation at hand. A value estimate for the result is computed. The third stage of the study is estimating the tree values. It's critical to perform estimates of the result values that are essential in decision making. The value of unknown outcomes is calculated next, followed by the value associated with the decision nodes. Following the measurement of decision incentives, the largest alternative is critical for decision-making.

1. Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome
2. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not

3. contain any further branches The decisions or the performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions a problem .
4. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches The decisions or the performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
5. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
6. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees. True Positive (TP)

The predicted value matches the actual value

The actual value was positive and the model predicted a positive value

True Negative (TN)

The predicted value matches the actual value

The actual value was negative and the model predicted a negative value

False Positive (FP) – Type 1 error

The predicted value was falsely predicted

The actual value was negative but the model predicted a positive value

Also known as the Type 1 error False

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	560	60
	NEGATIVE	50	330

Fig-2: Confusion Matrix

The different values of the Confusion matrix would be as follows:

True Positive (TP) = 560; meaning 560 positive class data points were correctly classified by the model
 True Negative (TN) = 330; meaning 330 negative class data points were correctly classified by the model

False Positive (FP) = 60; meaning 60 negative class data points were incorrectly classified as belonging to the positive class by the model

False Negative (FN) = 50; meaning 50 positive class data points were incorrectly classified as belonging to the negative class by the model

		Actual Class	
		Positive	
Class	Positive	True Positives	False Positive
	Negative	False Negative	True Negative

Table-1.2.1:Confusion Matrix

True Positives (TP) - These are the correctly predicted positive value which means that the value of the actual class is yes and the value of predicted class is also yes.

True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.

False Positives (FP) – When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN) – When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

Accuracy - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$Precision = \frac{TP}{TP + FP}$$

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

$$Recall = \frac{TP}{TP + FN}$$

F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

Bayes Theorem: Bayes Theorem is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

POST Method:

People, Objectives, Strategies and Technology are four most relevant factors defining the use of OSNs. People describe the platform over which the communication has to commence. Objectives will define the purpose of communication over OSNs. Strategy is to achieve more and more people and subscribers over the social networks. Technology demands a tool that is most appropriate to use. A combined effect of all these four factors ensures efficient OSN with high number of subscribers.

ACCURACY OF DETECTION MODELS

Most of the existing detection techniques use graph features of networks to distinguish between real and fake nodes. These techniques do not address scalability. Analysis of user metadata and machine learning techniques has an edge over the graph technique in detecting the fake profile. Active machine learning algorithms are more apt in checking the modern-day large-scale bots. Bulk content posting from a node could easily be detected on any social network.

1.3 PROBLEM STATEMENT

In advanced and persistent threats, false identities play an important role and are also involved in other malicious activities. The current issue focuses on detecting fake social media profiles. Approaches to the detection of fake social media accounts can be classified into approaches to the analysis of individual

accounts and methods to identify coordinated activities across a large group of accounts. The paper sheds light on the role of fake identities in continuing advanced threats and covers the Fake social media account detection methods. We needed a set of fake profiles to serve as a comparison data set in order to carry out this project. For this purpose, most current research in this area utilizes simulated fake profiles.

1.4 APPLICATIONS

In real life activities, approximately 1.0 billion users are using Instagram. It executes more than 10 million daily likes and shares. The growth of social networks has resulted in a very large number of false user profiles that leads to many of the profiles try to makefriends with benign users with the ultimate goal of accessing privileged data. Social engineering in any online social networks the main cause of threats.

CHAPTER 2

LITERATURE SURVEY

2.1 INFERENCES FROM LITERATURE SURVEY

1.YoungMin Kwon (2014). Fake social media profile detection.

Social media has become a vital part of our lives. From sharing attractive extravagant photographs to follow celebrities to chat with close and far away friends, everyone is active on social media. It is a great platform to share information and interact with people. But everything has a downside. As social media is footing a firm spot in our lives, there are instances where it has turned out to be a problem. There are 330 million monthly active users and 145 million daily active users on Twitter. Facebook also adds about 500,000 new users every day and six new users every second. Loads of information are shared over twitter every single day.

2.Vanshika (2019). Automatic detection of fake profiles on social networks.

These social networking sites are growing rapidly and changing the way people keep in contact with each other. The online communities bring people with same interests together which makes users easier to make new Social media has become a vital part of our lives. From sharing attractive extravagant photographs to follow celebrities to chat with close and far away friends, everyone is active on social media. It is a great platform to share information and interact with people. But everything has a downside. As social media is footing a firm spot in our lives, there are instances where it has turned out to be a problem. There are 330 million monthly active users and 145 million daily active users on Twitter.

3.Naman singh (2017).Online social networks using machine learning.

Now a day's social media is one of the easiest and a easiest way of spamming

spreading false news. Threats like spamming emails are most common in emails. A lot can be learned about how people react and the requirements of people through assessing their interactions with one another. We can assess people's usual behavior and topics of conversations to provide them with better services to the users at a vast scale.

4.Kumud patel (2018).Survey on fake profiles detection on social sites using machine learning algorithm.

In this paper Nambour Sravya, Chavana Sai Pranitha, S. Saraswathi has proposed a system for identifying malicious activity on social sites. Nowadays, social sites are a serious threat on the internet. To detect fake profile there are many models are proposed. Here, they focused on the Sybil and troll identities using Machine Learning Algorithm. Supervised Machine Learning Algorithm is recycled to overcome the problem. Sybil and troll accounts use an advanced technique, the data sets are collected by large data blogs then stored, if data is initiated malicious then data is clean and store again, after which cleaned shows the cleaned fake individualities and missing areas are fake individualities. Before clean store process, data is stored in a non-relational database for future reference and helps to remove the fake profile. To avoid the spam message, malicious and cyber bullies' activities which are mostly done by the fake profile.

Identifying fake accounts on social media has become a tedious issue for various sites such as Instagram, Facebook, Twitter, etc. It is generally done using several machine learning algorithms. Few methods which were previously used are not giving results up to the mark. In , a classifier is used for detecting malicious Facebook applications. Each app URL is evaluated using this machine-learning classifier. Here, Support Vector Machine (SVM)classifier has been used for classifying malicious applications. They have made use of the D-Complete dataset and 5-fold cross validation for the

purpose of training and testing the classifier. This dataset consists of 487 malicious apps and 2255 genuine apps.

5. Kai Shu, Suhang Wang, Huan liu- understanding user profiles on social media for fake detection.

Description:Because of the negative impact on individuals and culture, identifying fake news has become very relevant and is gaining growing attention. The efficiency of detecting fake news solely based on content is generally low, so it is suggested that user social engagements be introduced as auxiliary data to enhance fake news detection.As a result, a detailed understanding of the relationship between user accounts on social media and fake news is needed.

6. Jyoti Kaubiyal, Ankit Kumar Jain- A feature based approach to detect fake profiles in twitter.

Description:Social networking websites,especially Twitter and Facebook, have exploded in popularity in the last decade,attracting users worldwide.They've grown into a common source of information,attracting the attention of bad actors,such as cheaters.Fake accounts become more prevalent as the number of people using social media grows .These false and fake identities are heavily involved in malicious activities like spreading harassment, spreading disinformation, spamming, and falsely inflating the number of users in an app to encourage and manipulate public opinion.These false and fabricated identities are often used in fraudulent activities such as abuse, misinformation and depleting the percentage of users in an app to promote and influence public sentiments.To defend legitimate users from malicious intents, it's vital to identify these false identities.

7.Yuval Elovici, Michael FIRE, Gilad Katz- Method for detecting spammers and fake profiles in social networks.

Description:A tool for protecting user privacy in an online social network by choosing negative examples of fake profiles and positive examples of legitimate profiles from the social network's database of current users.Then, for each chosen fake and legitimate profile, a predetermined collection of features is extracted by dividing the friends or followers of the chosen examples into communities and analysing the relationships of each node within and between the communities. Using supervised learning, classifiers that can detect other existing fake profiles based on their features are designed and trained.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 INTRODUCTION

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems.

Introduction

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems.

In software engineering the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system the software development process.

The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high-level estimates of effort for the out stages.

- Testing
- Maintenance

3.4 REQUIREMENTS GATHERING STAGE

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas, and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.

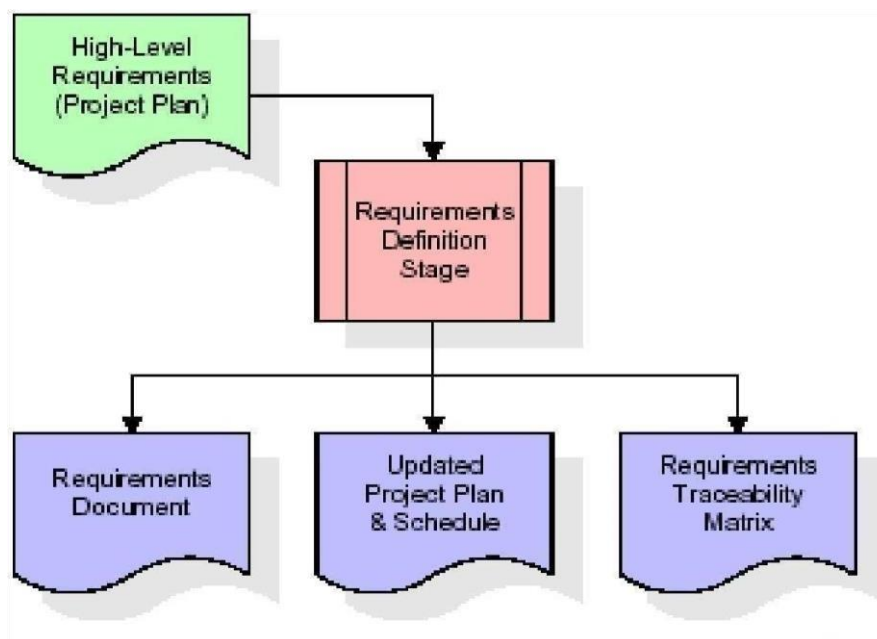


Fig-4: Requirements Gathering stage

These requirements are fully described in the primary deliverables for this stage: The Requirements Document and the Requirements Traceability

Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term *requirements traceability*.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

- Feasibility study is all about identification of problems in a project.
- No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.

3.5 ANALYSIS STAGE

The planning stage establishes a bird's eye view of the intended software and risks associated with the project, and describe appropriate management and technical approaches.

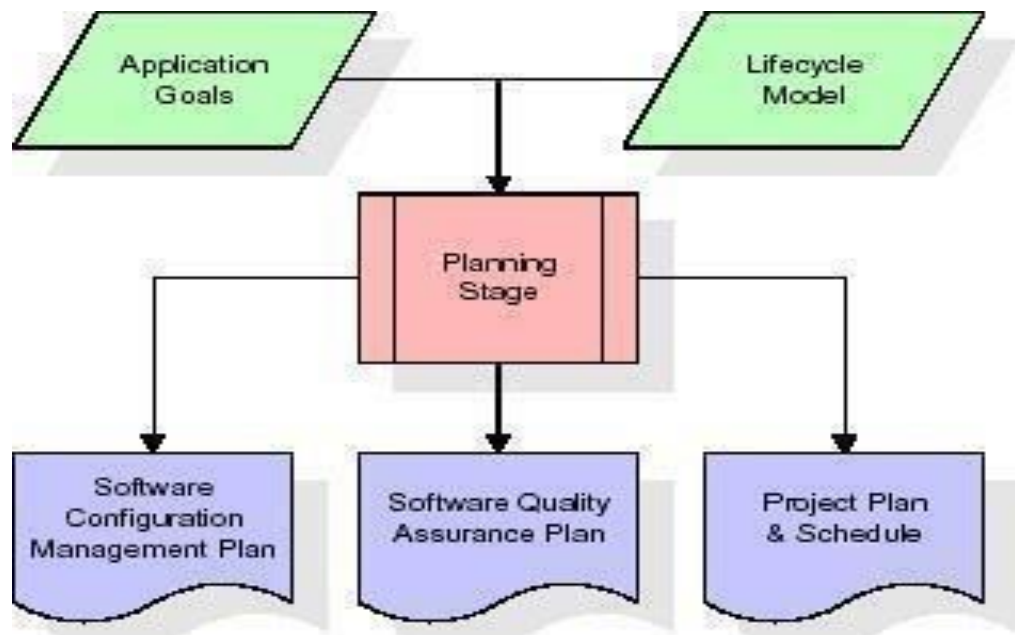


Fig-5: Analysis Stage

The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high-level estimates of effort for the out stages.

3.6 DESIGNING STAGE

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or

more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data

dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

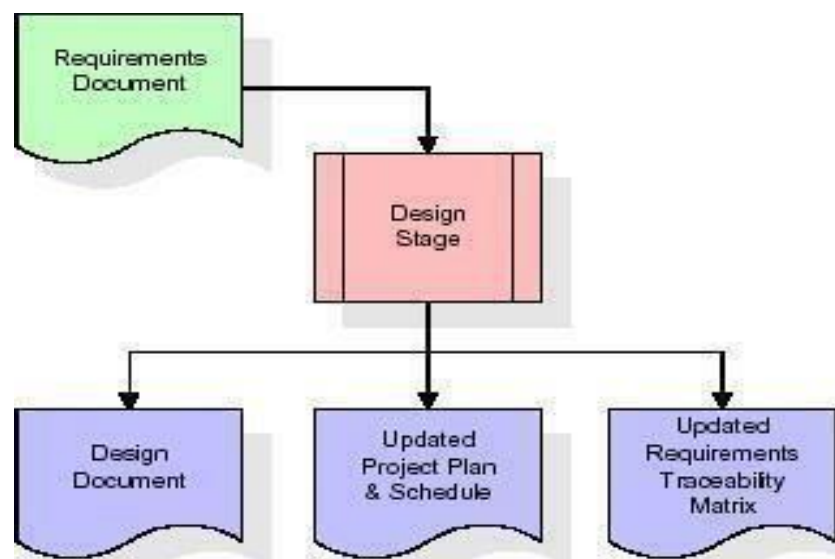


Fig-6:Designing Stage

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

3.7 DEVELOPMENT (CODING) STAGE

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set

of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

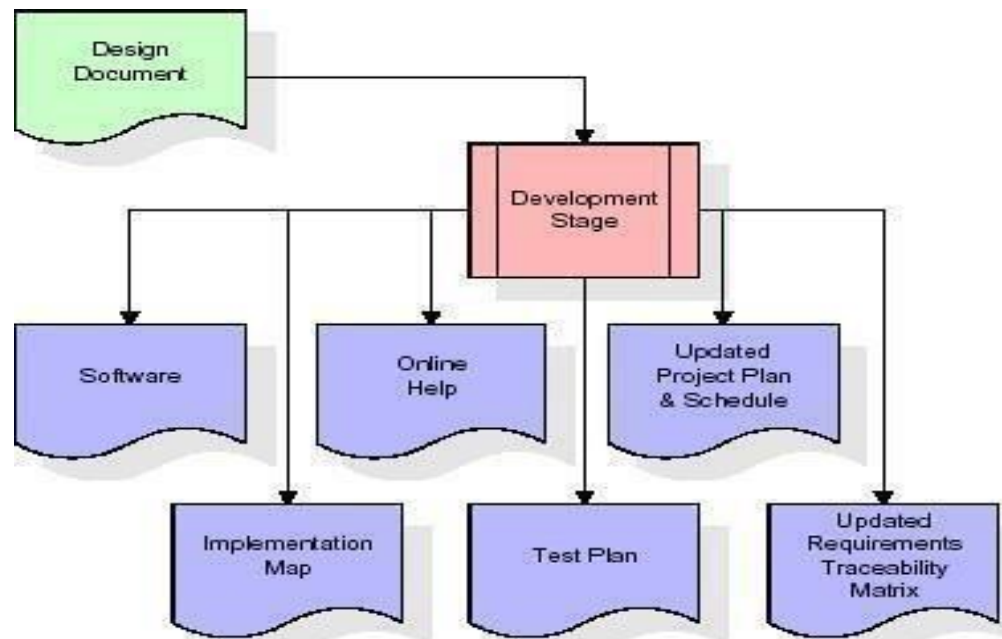


Fig-7: Development Stage

The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

3.8 INTEGRATION AND TEST STAGE

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

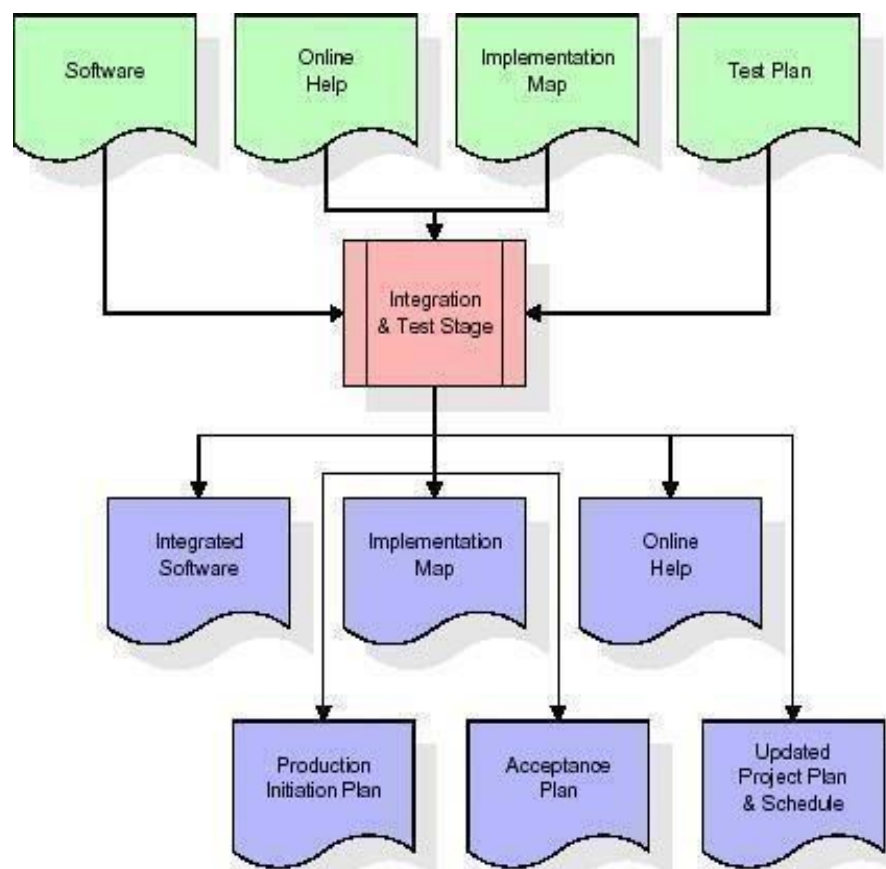


Fig-8: Integration & Test Stage

The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production

initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

3.9 INSTALLATION AND ACCEPTANCE TEST

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance. After of the software by the customer. customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software

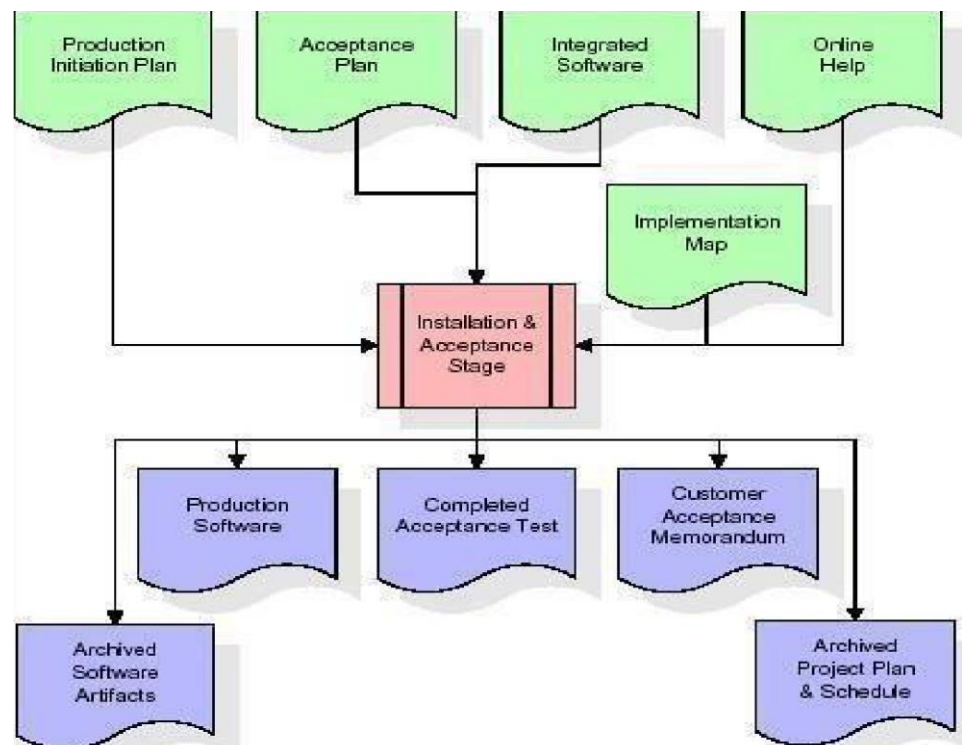


Fig-9: Installation & Acceptance Test

The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a

memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

3.9.1 MAINTENANCE

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

3.10 SOFTWARE REQUIREMENT SPECIFICATION

3.10.1 Overall Description

A Software Requirements Specification (SRS) – a requirements specification software system is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Nonfunctional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development lifecycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

- Business requirements describe in business terms *what* must be delivered or accomplished to provide value.
- Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business)

Process requirements describe activities performed by the developing organization. For instance, process requirements could specify. Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation.

ECONOMIC FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluate against the ultimate benefit derived from the new systems. Financial benefits must equal or exceeds the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, there is nominal expenditure and economic feasibility for certain.

OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above- mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

TECHNICAL FEASIBILITY

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

USER MODEL VIEW

- A. This view represents the system from the user's perspective.
- b. The analysis representation describes a usage scenario from the end- users perspective

STRUCTURAL MODEL VIEW

- i. In this model the data and functionality are arrived from inside the system.
- ii. This model view models the static structures.

BEHAVIORAL MODEL VIEW

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are to be built.

Environmental Model View

In these the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

4.1 DESIGN METHODOLOGY(UML representation):

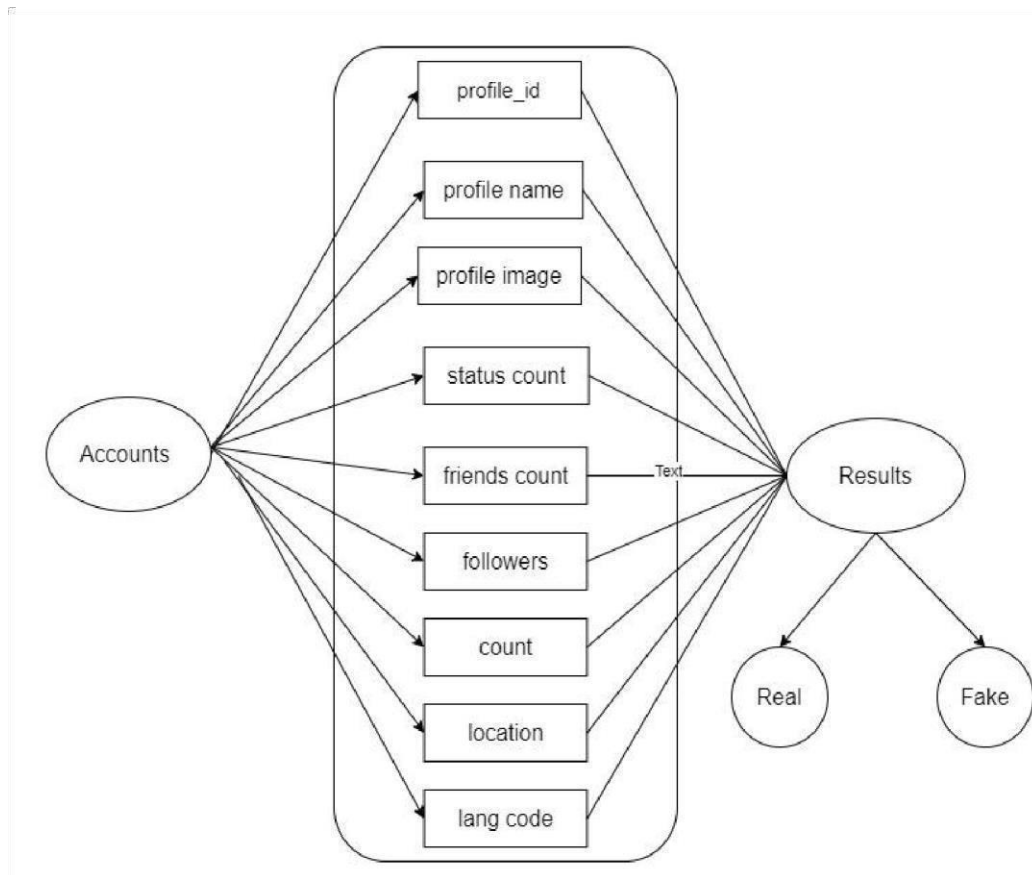


Fig-10: UML Representation

4.2 USE CASE DIAGRAM

use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well. Use case diagram is used to show which operations are performed by the user and which operation are performed by the system.

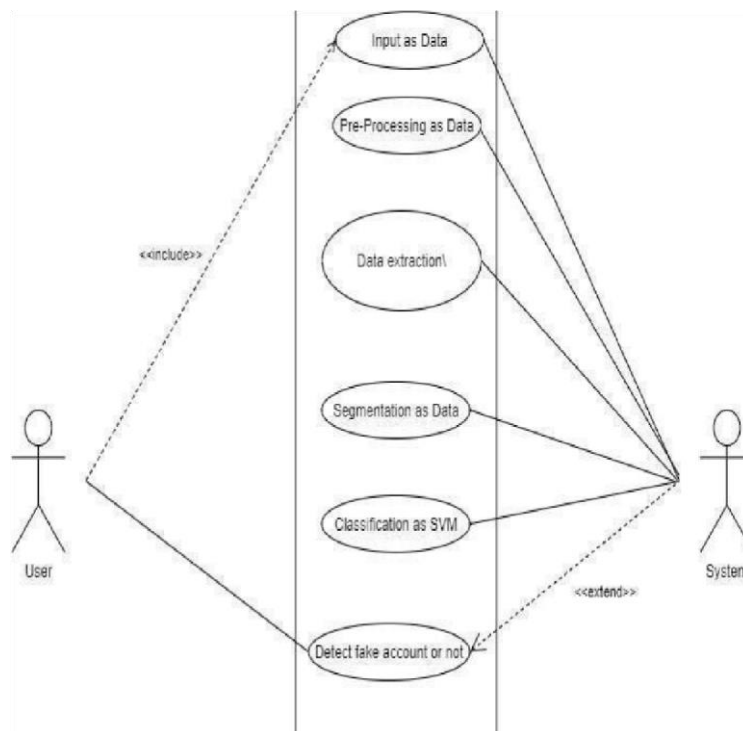


Fig-11: Use case Diagram

4.3 CLASS DIAGRAM

The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to

be programmed. A class with three sections, in the diagram, classes is represented with boxes which contain three parts:

- The upper part holds the name of the class
- The middle part contains the attributes of the class
- The bottom part gives the methods or operations the class can take or undertake

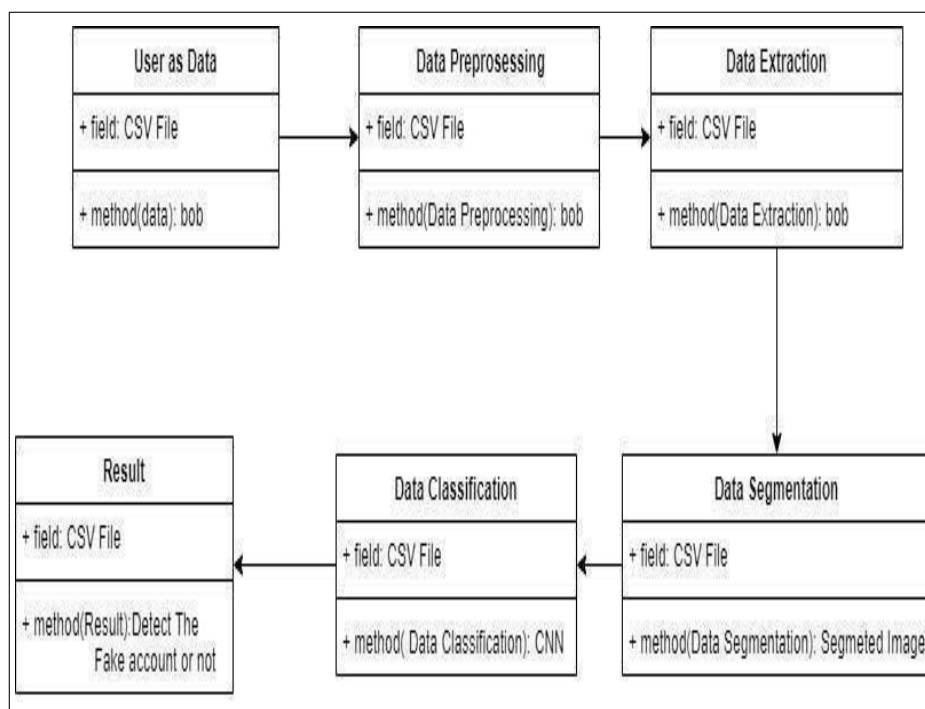


Fig-12:Class Diagram

4.4 SEQUENCE DIAGRAM

In sequence diagram step by step sequence of steps is shown. In above diagram first preprocess all train data and test data. Then by applying the train data Train the machine and build the module and at the last apply machine learning algorithm on it. For testing purpose apply the test data on module and see the classification either fake or real.

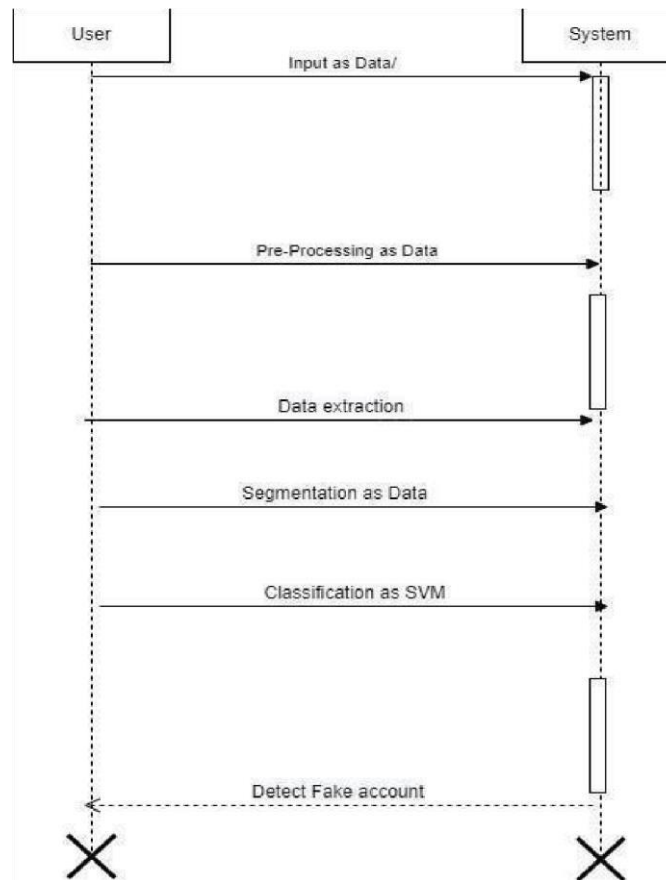


Fig-13: Sequence Diagram

4.5 ACTIVITY DIAGRAM

Activity Diagram shows the active flow of the system. In above diagram the flow of our project is shown how the data flow. Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

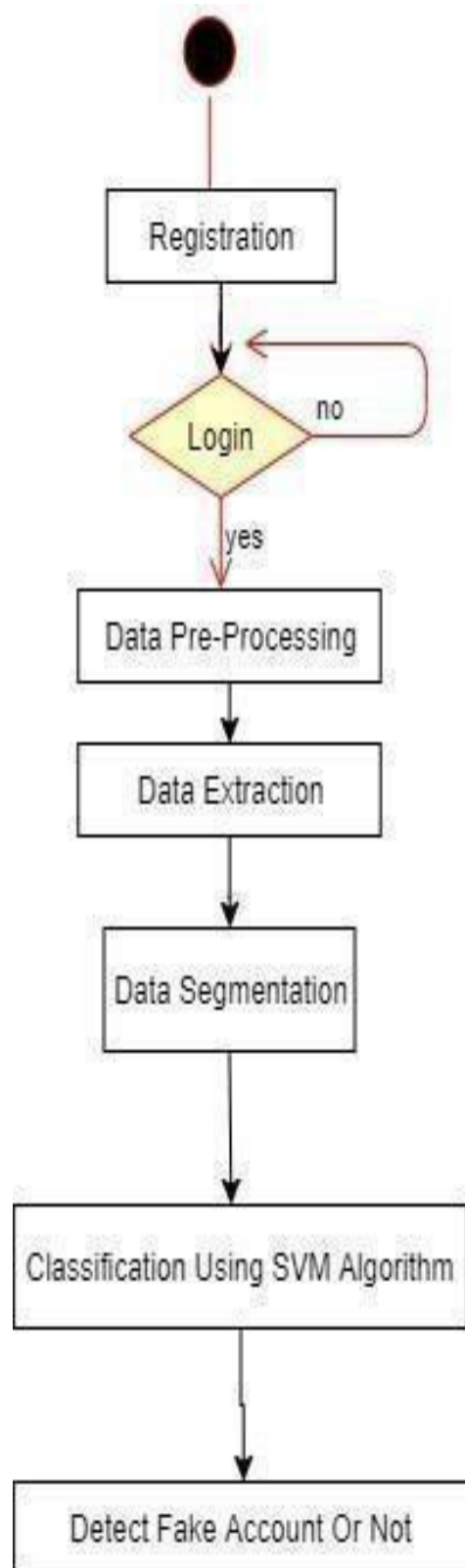


Fig-14: Activity Diagram

CHAPTER 5

IMPLEMENTATION DEATILS

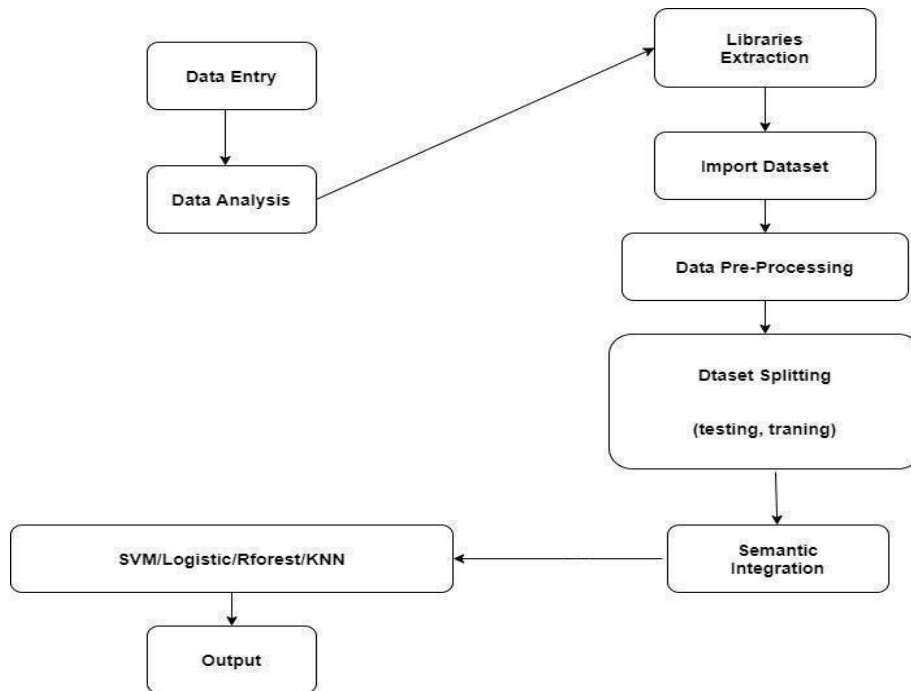


Fig -15:SYSTEM ARCHITECTURE

5.1 DESCRIPTION OF ALGORITHM:

Support Vector Machine Classifier:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems.

However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision

boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane

These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

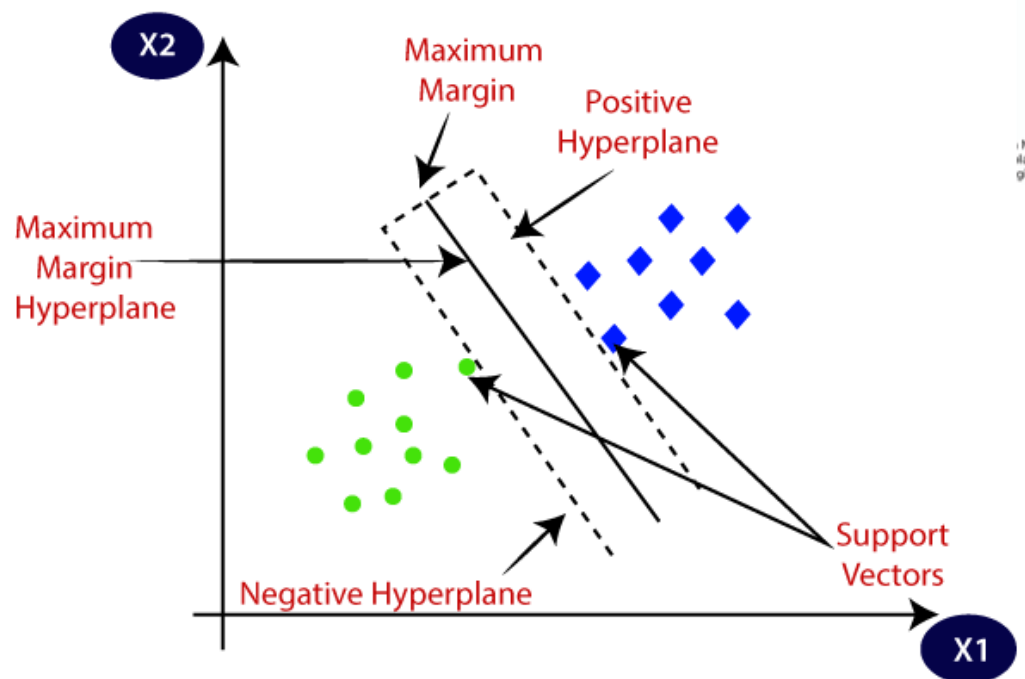


Fig-16 :HYPERPLANE

Types of SVM: SVM can be of two types:

Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Nonlinear SVM classifier

HYPERPLANE :

There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM. The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane. We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

SUPPORT VECTORS

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

INPUT: Raw text files consist of RNA

- 1 Extract the RNA sequence from the raw text files
- 2 Building the data from count vectorization
- 3 Standardization of the data using sklearn package.
- 4 Split the data into training and testing
- 5 Plot the data set data in N dimensional space
- 6 determine the hyperplane with the help of weight vector(w) and bias(b).
- 7 If > 0 then positive
- 8 If < 0 then negative

- 9 Repeat the step 6,7,8 until the find the hyperplane with maximum margin.
- 10 Find the accuracy using confusion matrix.
- 11 Classify new RNA sequence using above model.

RANDOM FOREST:

Random Forest is one of the most used machine learning algorithms, because its simplicity and the fact that it can be used for both classification and regression tasks. It's a supervised learning algorithm. As it's seen from its name, it's bagged decision tree models that split on a subset of features on each split, it creates a forest and makes it somehow random. The forest it builds, is an ensemble of Decision Trees, most of the time trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result. To say it in simple words: Random Forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

The better understanding of the RF algorithm is necessary to explain what the main idea behind decision trees is. Depending on the features in each dataset, the decision tree model learns a series of questions to figure out the class labels of the instances. What makes this model successful is that it is nonparametric and it can handle heterogeneous data (ordered or categorical variables, or a mix of both). Furthermore, decision trees fundamentally implement feature selection, making them at least to some extent robust to irrelevant or noisy variables and are robust to outliers or errors in labels. To summarize, here are the steps that Random Forest algorithm follows: Randomly choose n samples from the training set with replacement. Grow a decision tree from the n sample. At each node. • Repeat the steps 1 to 2 k -times. Aggregate the prediction by each tree to assign the class label by majority vote. Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But

together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.
- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique.
- Since the random forest combines multiple trees to predict the class of the dataset
- It takes less training time as compared to other algorithms. It predicts output with high accuracy, even for the large dataset it runs efficiently.
- As its seen from its name, its bagged of decisiontree models, that creates forest and makes it somehow random. It can also maintain accuracy when a large proportion of data is missing. The below diagram explains the working of the Random Forest algorithm:
- These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine
- The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector
- The final feature importance, at the Random Forest level, is its average over all the trees.

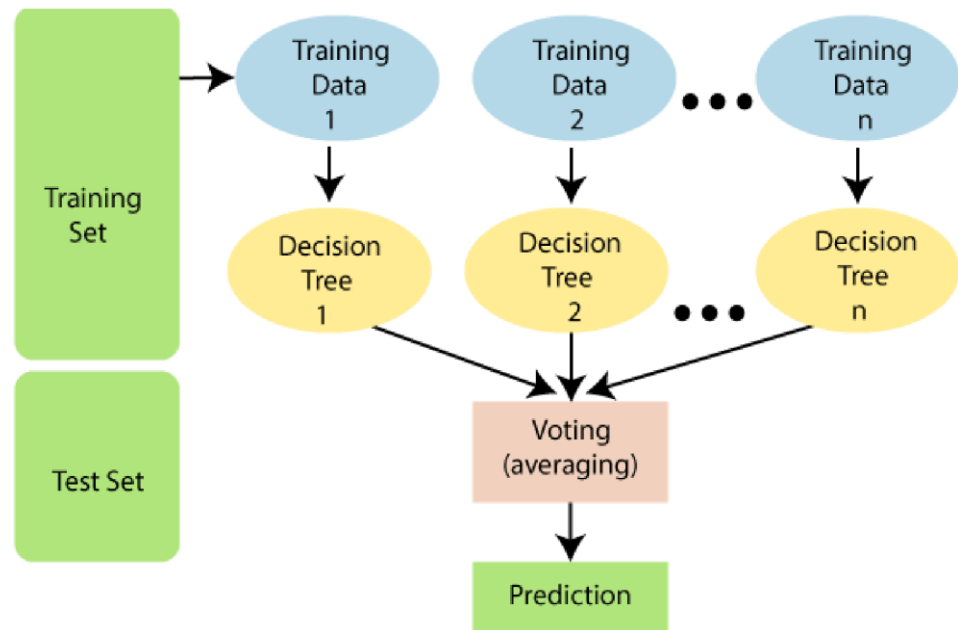


Fig-17: SVM Tree

The importance for each feature on a decision tree is then calculated as:

$$f_i = \frac{\sum_{j: \text{node } j \text{ splits on feature } i} n_{ij}}{\sum_{k \in \text{all nodes}} n_{ik}}$$

- f_i = the importance of feature i

- n_{ij} = the importance of node j

These can then be normalized to a value between 0 and 1 by dividing by the sum of all feature importance values:

The final feature importance, at the Random Forest level, is its average over all the trees. The sum of the feature's importance value on each tree is calculated and divided by the total number of trees:

RFf_i = the importance of feature i calculated from all trees in the Random Forest model

- $normf_i$ = the normalized feature importance for i in tree j

- T = total number of trees

DECISION TREE

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node.

Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs tree-like structure.

How does the Decision Tree algorithm Work?

for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs tree-like structure. The complete process can be better understood using the below algorithm:

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.

Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).

Step-3: Divide the S into subsets that contains possible values for the best attributes.

Step-4: Generate the decision tree node, which contains the best attribute.

Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you

cannot further classify the nodes and called the final node as a leaf node.

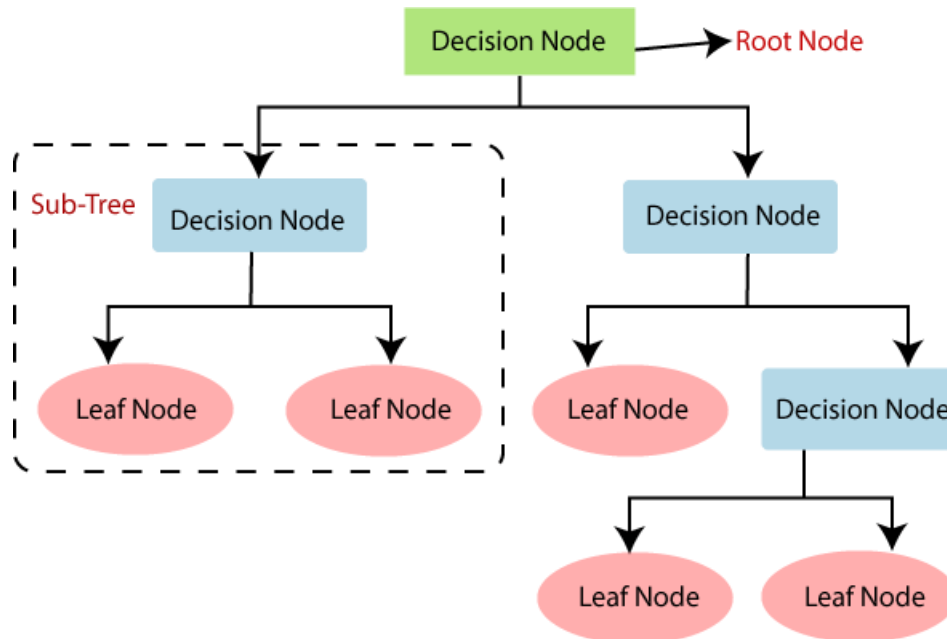


Fig-18: Decision Tree

Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

Splitting: Splitting is the process of dividing the decision node/root node into sub- nodes according to the given conditions. **Branch/Sub Tree:** A tree formed by splitting the tree.

Pruning: Pruning is the process of removing the unwanted branches tree.

Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes. While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as Attribute selection measure or ASM. By this measurement, we can

easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

Information Gain: Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute. It calculates how much information a feature provides us about a class.

Entropy: Entropy is a metric to measure the impurity in a [□] given attribute. It specifies randomness in data. To compute Gini impurity for a set of items with J classes, suppose $i \in \{1, 2, \dots, J\}$ and let p_i be the fraction of items labeled with class i in the set.

$$I_G(p) = \sum_{i=1}^J p_i(1 - p_i)$$

Evaluation Parameters

Efficiency/Accuracy = Number of predictions/Total

Number of Predictions Percent Error = $(1 - \text{Accuracy}) * 100$

Confusion Matrix - Confusion Matrix is a technique for summarizing the performance of a classification algorithm. TPR- True Positive Rate

$\text{TPR} = \text{TP} / (\text{TP} + \text{FN})$

FPR- False Positive Rate $\text{FPR} = \text{FP} / (\text{FP} + \text{TN})$ TNR- True

Negative Rate $\text{TNR} = \text{TN} / (\text{FP} + \text{TN})$

FNR- False Negative Rate $\text{FNR} = 1 -$

TPR

Precision- Precision is how many of the returned hits were true positive i.e. how many of the found were correct hits.

$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$

PROPOSED METHODOLOGY

Each profile (or account) in a social network contains lots of information such as gender, no. of friends, no. of comments, education, work etc. Some of this information are private and some are public. Since private information is not accessible so, we have used only the information that are public to determine the fake profiles in social network. However, if our proposed scheme is used by the social networking companies itself then they can use the private information of the profiles for detection without violating any privacy issues. We have considered this information as features of a profile for classification of fake and real profiles. The steps that we have followed for detection of fake profiles are as follows.

After proper selection of attributes, the dataset of previously identified fake and real profiles are needed for the training purpose of the classification algorithm. We have made the real profile dataset whereas the fake profile dataset is provided by the Barracuda Labs, a privately held company providing security, appliances and cloud services. The collection of datasets is discussed. The attributes selected in step 1 are needed to be extracted from the profiles (fake and real). The process of extraction of the features that we have used is discussed. For the social networking companies which want to implement our scheme don't need to follow the scrapping process, they can easily extract the features from their database. We applied scrapping of the profiles since no social network dataset is available publicly for the research purpose of detecting the fake profiles.

After this the dataset of fake and real profiles are prepared. From this dataset, 80% of both profiles (real and fake) are used to prepare a training dataset and 20% of both profiles are used to prepare a testing dataset. We find the efficiency of the classification algorithm using the training dataset containing 922 profiles and testing dataset containing 240 profiles.

After preparation of the training and the testing dataset, the training dataset is feed to the classification algorithm. It learns from the training algorithm and is expected to give correct class levels for the testing dataset.

The levels from the testing dataset are removed and are left for determination by Proposed framework Proposed Work the trained classifier. The efficiency of the classifier is calculated by calculating the no. of correct prediction divided by total no. of predictions. The result of classification is shown in the section. We have used three classification algorithms and have compared the efficiency of classification of these algorithms

Firstly, the data of different users have been collected from the Instagram with some useful columns.

Then the dataset is preprocessed by using the data cleaning techniques that fill the missing values and also the outlier detection.

- Then the preprocessed data of fake users and genuine users is splitter into 80% training and 20% testing data by Scipy and Scikit learn modules in Python.
- Then feature column are extracted from the data according to the weight they play on finding an account credibility.
- Then the classifier Random Forest is trained with the training data.
- Finally the model is tested with the test data and the accuracy is built as 90%.
- Then the Confusion matrix and also Learning curve is given for result analysis.

5.2 DESCRIPTION OF DATASETS AND TOOLS:

Dataset:

We considered a dataset of fake and genuine users from Instagram. Various attributes included in the dataset are a number of friends, followers, status count. Dataset is divided into training and testing data. Classification

algorithms are trained using a training dataset and the testing dataset is used to determine the efficiency of the algorithm. From the dataset used, 80% of both profiles (genuine and fake) are used to prepare a training dataset and 20% of both profiles are used to prepare a testing dataset.

Name: Social network fake account dataset

Source: <https://www.kaggle.com/bitandatom/social-network-faccountdataset>

S. No	Attribute	Description
1.	Profile ID	The Profile ID of account holder
2.	Profile Name	The name of the account holder
3.	Status Count	The number of tweets made by the account
4.	Followers Count	The number of followers for the account
5.	Friends Count	The number of friends for the account
6.	Location	The location of the account holder
7.	Created Date	The date the account was created
8.	Share count	The number of shares done by account holder
9.	Gender	The Gender of the account holder
10.	Language Code	The language of account holder

TABLE 1.2: DATASET

Above table shows the Attributes considered for fake profile identification and the description for each of the attributes is provided.

Software Requirements:

Jupyter Notebook (Python

3.6 to 3.6) Scipy, Pandas,

Scikitlearn, Imputer

Windows 10, I3 4GB RAM

CHAPTER-6

RESULTS AND DISCUSSION

6.1 STEPWISE DESCRIPTION OF RESULTS

Each profile (or account) in a social network contain lots of information such as gender, no. of friends, no. of comments, education, work etc. Some of this information are private and some are public. Since private information is not accessible so, we have used only the information that are public to determine the fake profiles in social network. However, if our proposed scheme is used by the social networking companies itself then they can use the private information of the profiles for detection without violating any privacy issues. We have considered these information as features of a profile for classification of fake and real profiles.

The steps that we have followed for detection of fake profiles are as follows.

The detection process starts with the selection of the profile that needs to be tested.

- After selection of the profile, the suitable attributes (i.e. features) are selected on which the classification algorithm is implemented.
- The attributes extracted is passed to the trained classifier. The classifier gets trained regularly as new training data is feed into the classifier.
- The classifier determines the whether the profile is fake or real.
- The classifier may not be 100% accurate in classifying the profile so; the feedback of the result is given back to the classifier. For example, if the profile is identified.

- This process repeats and as the time proceeds, the no. of training data increase and the classifier becomes more and more accurate in predicting the fake profiles

6.2 TEST CASE RESULTS

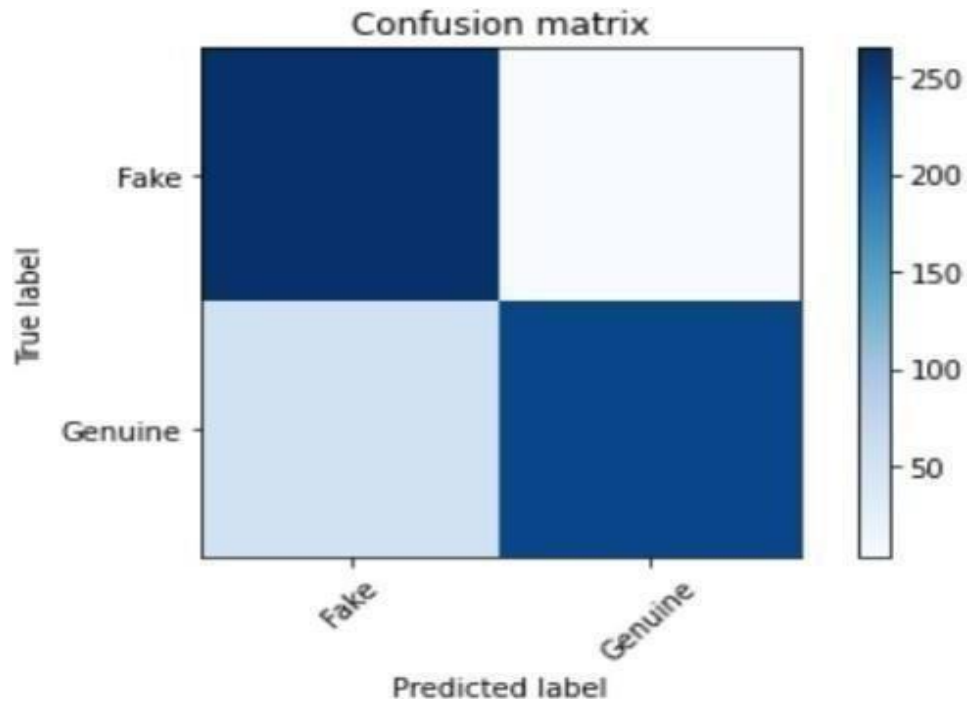
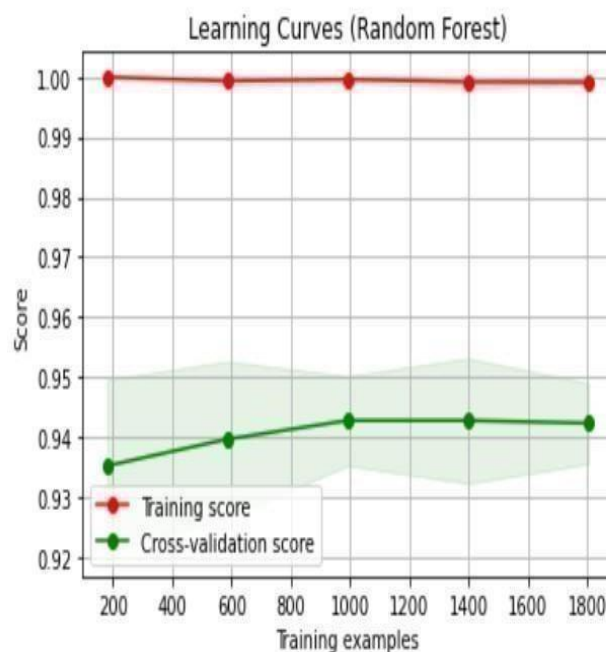


Fig -19:CONFUSTION MATRIX

CONFUSTION MATRIX

	precision	recall	f1-score	support
Fake	0.83	0.99	0.90	268
Genuine	0.99	0.82	0.90	296
accuracy			0.90	564
macro avg	0.91	0.90	0.90	564
weighted avg	0.91	0.90	0.90	564

The best classifier is: RandomForestClassifier(n_estimators=40, oob_score=True)
 [0.92904656 0.93791574 0.94456763 0.94678492 0.94666667]
 Estimated score: 0.94100 (+/- 0.00340)



Classification Accuracy on Test dataset: 0.9450354609929078
 Confusion matrix, without normalization
 [[265 3]
 [28 268]]

Fig- 20 Learning curve

CHAPTER -7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

Fake profile over the online social networks is a reality. Fake profiles are sharply on the rise as people accept it as a common medium of communication. More and more personal details are now included on social network profiles, causing enormous information content to be available for manipulation. Social engineering techniques and Fake profiles have really threatened stealing of information and careful manipulation towards furtherance of ulterior motives. In this paper we reviewed Random Forest methods of detecting the fake profiles in existing social networks. using machine learning methods.

7-2 FUTURE STUDY

We have given a framework using which we can identify fake profiles in any online social network by using Random Forest method Classifier with a very high efficiency as high as around 90%. Fake profile Identification can be improved by applying NLP techniques and Neural Networks to process the posts and the profiles.

REFERENCES

- [1] A. S. M. Salih and A. Abraham, "Novel ensemble decision support and health care monitoring system," *Journal of Network and Innovative Computing*, vol. 2, no.2014, pp.041-051,2014.
- [2] B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. P. Gummadi, Krishnamurthy, and A. Mislove, "Towards detecting anomalous user behavior inonline social networks." in *USENIX Security Symposium*, 2014, pp. 223–238.
- [3] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao, "You arehow you click: Clickstream analysis for sybil detection." in *USENIX Security Symposium*, vol. 9, 2013, pp. 1–008.
- [4] Integro: Leveraging victim prediction for robust fake account detection in large scaleosns," *Computers & Security*, vol. 61, pp.142–168, 2016. Q. Cao, M. Sirivianos, X.
- [5] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi, "Sok: The evolution of sybil defense via social networks," in *Security and Privacy (SP)*, 2013 IEEE Symposium on. IEEE, 2013, pp. 382–396.
- [6] M. Tsikerdekis and S. Zeadally, "Multiple account identity deception detection social media using nonverbal behavior," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 8, pp. 1311–1321, 2014.
- [7] Matt. Social Media Comparison Info graphic.2014.atnetworks. Knowledge-Based Intelligent Information and Engineering Systems Springer (2018) Human or 'bot'? doubts over italian comic beppe grillo's twitter followers.
- [8] P. Patel, K. Kannoorpatti, B. Shanmugam, S. Azam, and K. C. Yeo, "A theoretical V review of social media usage by cyber-criminals," in *Computer Communication and Informatics (ICCCI)*, 2017 International Conference on. IEEE, 2017, pp. 1–6.

- [9] Q last check 2018) Fake followers check: A new free tool from socialbakers.Internetdraft.[Online].
- [10] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "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.
- [11] S. Adikari and K. Dutta, "Identifying fake profiles in linkedin." in PACIS, 2014, p. 278 S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "Fame for sale: efficient detection of fake twitter followers," Decision Support Systems, vol. 80, pp. 56– 71, 2015.
- [12] S. Fong, Y. Zhuang, and J. He, "Not every friend on a social network can be trusted: Classifying imposters using decision trees," in Future Generation Communication Technology (FGCT), 2012 International Conference on. IEEE, 2012, pp. 58–63.
- [13] Yang, and T. Pregueiro, "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. USENIX Association, 2012, pp. 15–15

A.APPENDIX

SOURCE CODE

```
# libraries imported import

pandas as pd import

matplotlib.pyplot as plt

import numpy as np import

seaborn as sns import

tensorflow as tf from

tensorflow import keras

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

tensorflow.keras.optimizers import Adam from

tensorflow.keras.metrics import Accuracy from sklearn import metrics

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import

classification_report,accuracy_score,roc_curve,confusion_matrix

# Load the training dataset

instagram_df_train=pd.read_csv('/content/insta_train.csv')

instagram_df_train # Load the testing data

instagram_df_test=pd.read_csv('/content/insta_test.csv')

instagram_df_test instagram_df_train.head()

instagram_df_train.tail() instagram_df_test.head()

instagram_df_test.tail()
```

#Performing Exploratory Data Analysis EDA

Getting dataframe info instagram_df_train.info()

Get the statistical summary of the dataframe

instagram_df_train.describe() # Checking if null

values exist instagram_df_train.isnull().sum()

Get the number of unique values in the "profile pic" feature instagram_df_train['profile pic'].value_counts()

Get the number of unique values in "fake" (Target column)

instagram_df_train['fake'].value_counts()

instagram_df_test.info() instagram_df_test.describe()

instagram_df_test.isnull().sum()

instagram_df_test['fake'].value_counts() # Perform Data

Visualizations # Visualize the data

sns.countplot(instagram_df_train['fake']) plt.show()

Visualize the private column data

sns.countplot(instagram_df_train['private'])

plt.show()

Visualize the "profile pic" column data

sns.countplot(instagram_df_train['profile pic']) plt.show() #

Visualize the data plt.figure(figsize = (20, 10))

sns.distplot(instagram_df_train['nums/length username'])

plt.show()

Correlation plot plt.figure(figsize=(20,

20)) cm = instagram_df_train.corr() ax

```

= plt.subplot()

# heatmap for correlation matrix
sns.heatmap(cm, annot = True, ax = ax)

plt.show()

sns.countplot(instagram_df_test['fake'])

sns.countplot(instagram_df_test['private'])

sns.countplot(instagram_df_test['profile pic'])

# Preparing Data to Train the Model

# Training and testing dataset (inputs)

X_train = instagram_df_train.drop(columns = ['fake'])

X_test = instagram_df_test.drop(columns = ['fake'])

X_train

X_test

# Training and testing dataset (Outputs)

y_train = instagram_df_train['fake']
y_test = instagram_df_test['fake']

# Scale the data before training the model

from sklearn.preprocessing import StandardScaler, MinMaxScaler
scaler_x = StandardScaler()

```

```
X_train = scaler_x.fit_transform(X_train)
```

```
X_test = scaler_x.transform(X_test)
```

```
y_train = tf.keras.utils.to_categorical(y_train, num_classes = 2)
```

```
y_test = tf.keras.utils.to_categorical(y_test, num_classes = 2) y_train
```

```
y_test
```

```
# print the shapes of training and testing datasets
```

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
Training_data = len(X_train)/( len(X_test) + len(X_train) ) * 100
```

```
Training_data
```

```
Testing_data = len(X_test)/( len(X_test) + len(X_train) ) * 100
```

```
Testing_data
```

```
# Building and Training Deep Training Model
```

```
import tensorflow.keras from
```

```
tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.layers import Dense, Dropout  
model = Sequential()
```

```
model.add(Dense(50, input_dim=11, activation='relu'))
```

```

model.add(Dense(150, activation='relu')) model.add(Dropout(0.3)) model.add(Dense(150,
activation='relu')) model.add(Dropout(0.3))

model.add(Dense(25, activation='relu'))

model.add(Dropout(0.3))

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

model.summary()      model.compile(optimizer = 'adam', loss =
'categorical_crossentropy', metrics =
['accuracy']) epochs_hist = model.fit(X_train, y_train, epochs = 50, verbose = 1, validation_split
= 0.1) import tensorflow.keras

from tensorflow.keras.models

import Sequential from tensorflow.keras.layers import
Dense, Dropout model = Sequential() model.add(Dense(50,
input_dim=11, activation='relu')) model.add(Dense(150,
activation='relu')) model.add(Dropout(0.3))
model.add(Dense(25, activation='relu'))
model.add(Dropout(0.3)) model.add(Dense(25,
activation='relu')) model.add(Dropout(0.3))
model.add(Dense(2, activation='softmax'))
model.summary()

# Access the Performance of the model

print(epochs_hist.history.keys())

```

```

plt.plot(epochs_hist.history['loss'])

plt.plot(epochs_hist.history['val_loss']) plt.title('Model Loss
Progression During Training/Validation') plt.ylabel('Training
and Validation Losses') plt.xlabel('Epoch Number')

plt.legend(['Training Loss', 'Validation Loss']) plt.show()

predicted = model.predict(X_test) predicted_value = []

test = [] for i in predicted:

predicted_value.append(np.argmax(i))

for i in y_test:

test.append(np.argmax(i)) print(classification_report(test, predicted_value))

plt.figure(figsize=(10, 10))

cm=confusion_matrix(test, predicted_value) sns.heatmap(cm, annot=True)

plt.show()

```

B.SCREENSHOTS

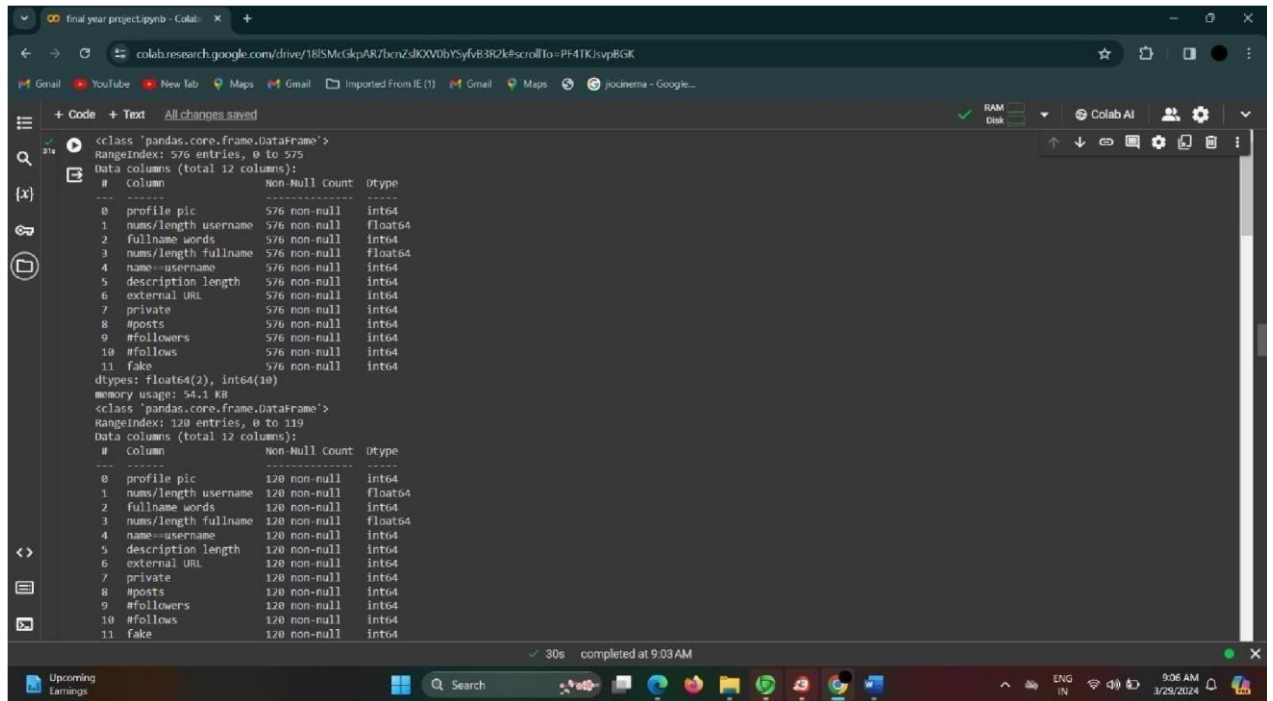


Fig B.1:DATA COLUMNS

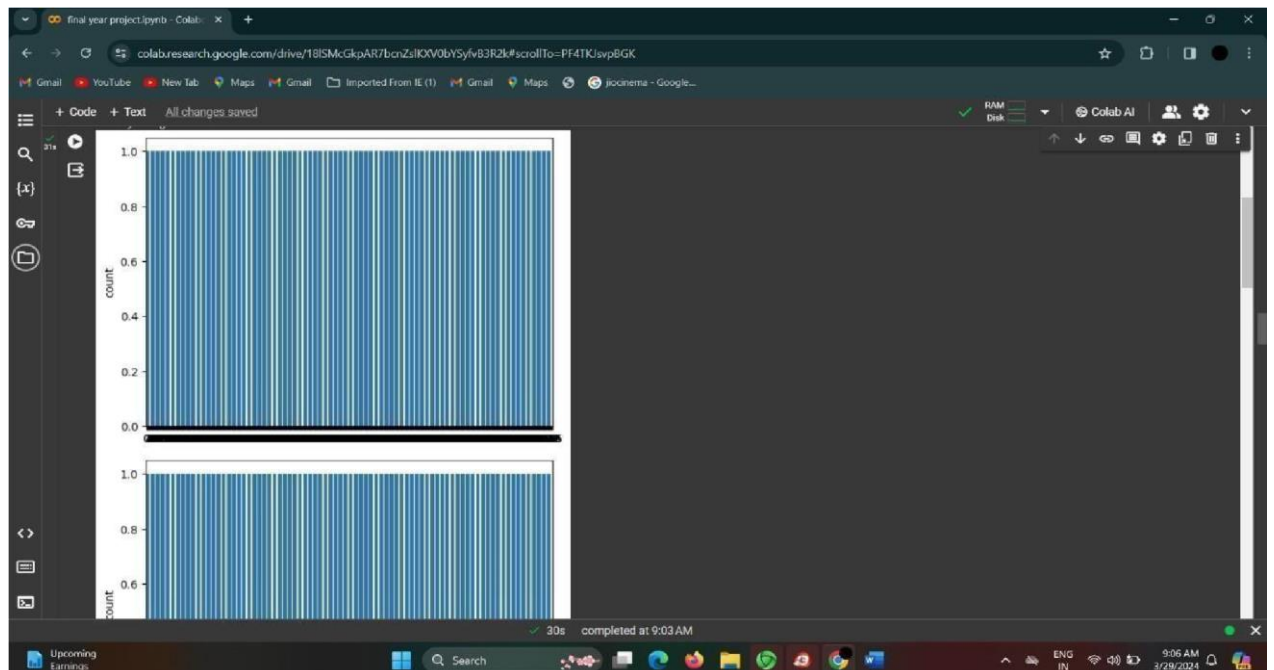


Fig B.2 :GRAPH OF DATA COLUMNS

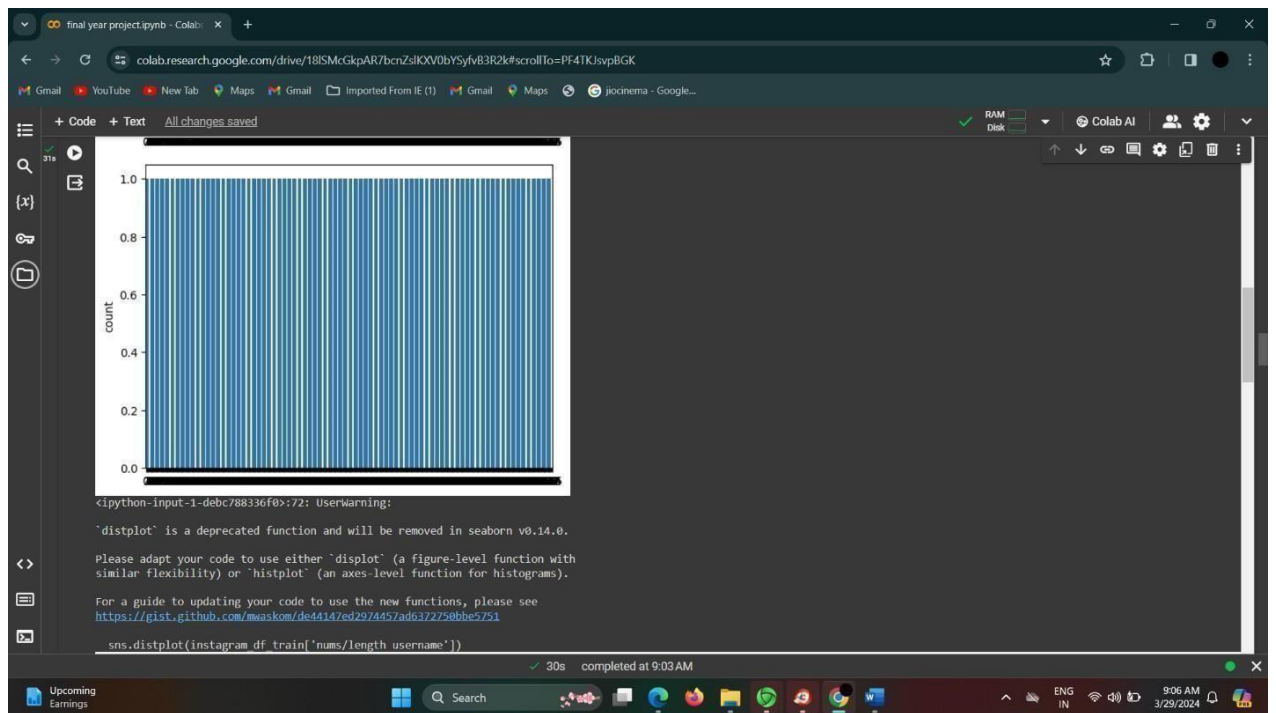


Fig B.3 :GRAPH OF DATA COLUMNS

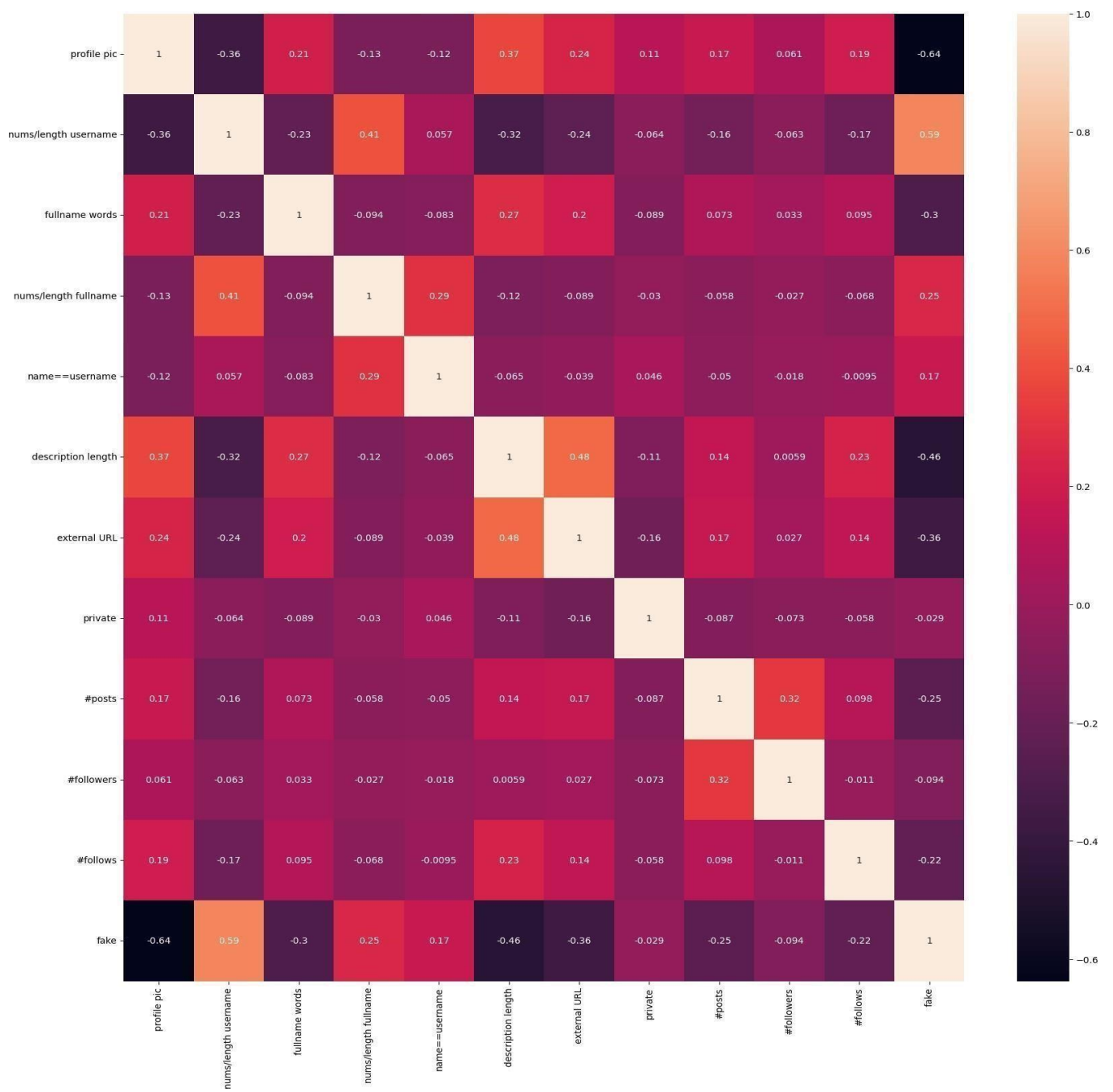


FIG B.4 CORRELATION MATRIX

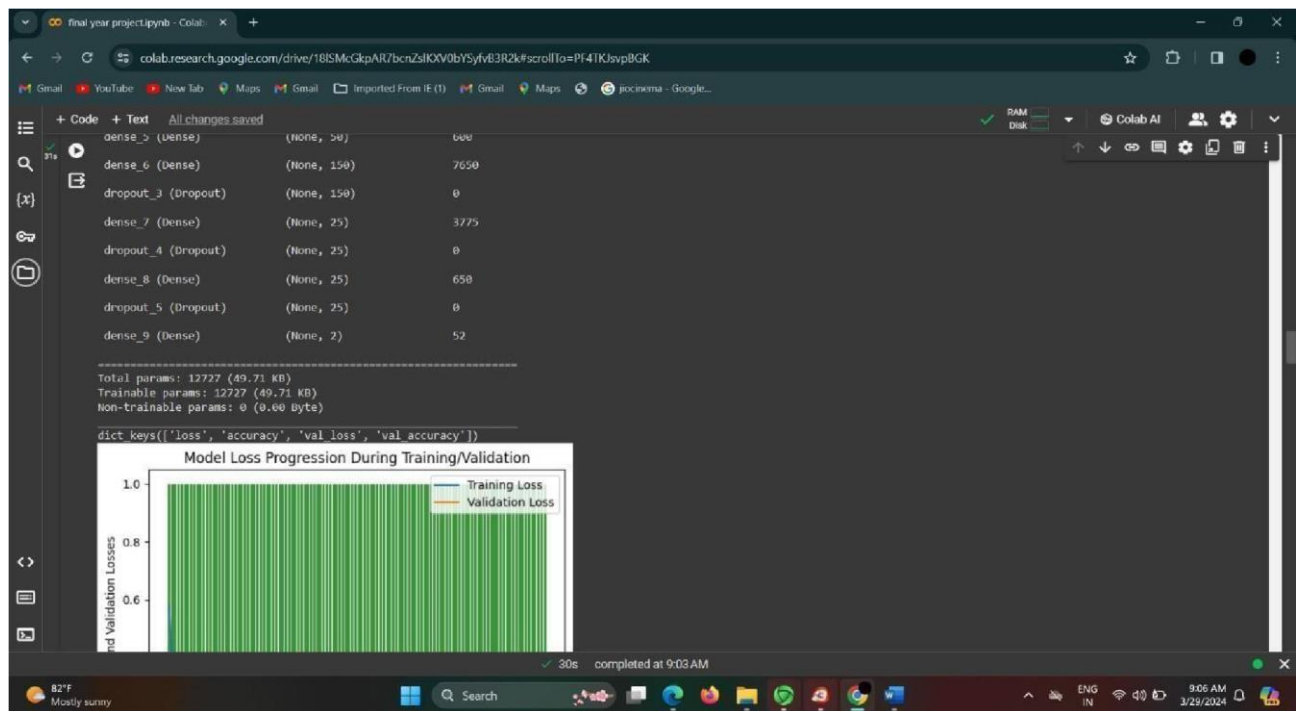


Fig B.5: RESULT OF CORRELATION MATRIX

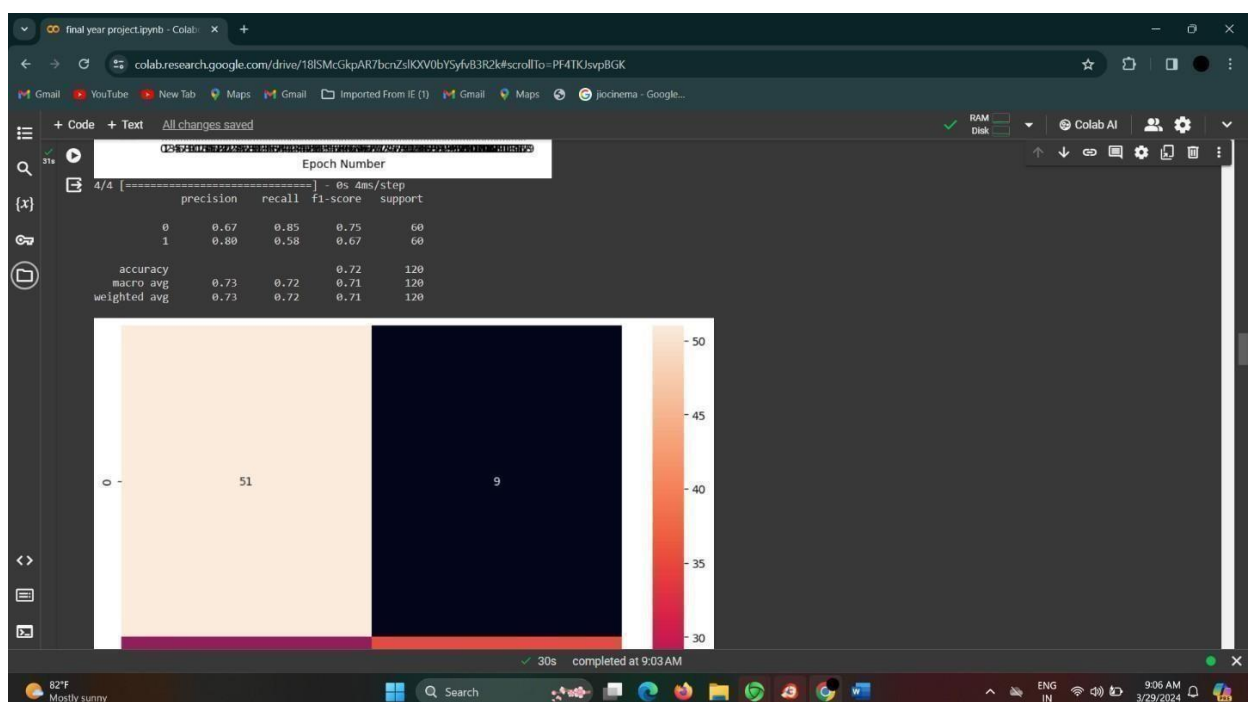


FIG B.7 : HEATMAP OF DATA

C.RESEARCH PAPER

A Versatile Tool for Protecting User from Fraudsters and Scammers

Sai Ram p

Department of Computer Science and Engineering
Sathyabama Institute of Science and Technology,
sairampendem547@gmail.com

Surendra B

Department of Computer Science and Engineering
Sathyabama Institute of Science and Technology,
suri9494031640@gmail.com

Dr.s.Jayanthi.M.E.,Ph.D

Department of Computer Science and Engineering
Sathyabama Institute of Science and Technology,
jayanthi.cse@sathyabama.com

Abstract--- Nowadays, no one's social life would be complete without their participation in any online social network. The way we pursue our social lives has drastically changed as a result of these websites. It's now simpler to make friends, stay in touch with them, and read their updates. However, several issues including false profiles and online impersonation have also increased along with their quick rise. There isn't a workable answer to manage these issues. In this research, we developed a framework that makes efficient automatic identification of bogus profiles possible. This system classifies the profiles into fictitious and real classes using classification techniques such as Random Forest. Online social networks may simply use this automated identification method because there are millions of profiles to go through.**Keywords---** Support Vector Machine, Random Forest,

I.INTRODUCTION

a) Origin of the Problem: -

The utilization of social media platforms including Instagram, LinkedIn, Google+, Facebook, and Twitter is increasing. Social networks are used by people and businesses to share their opinions, promote their goods, and communicate their future corporate and organizational strategies. individuals are becoming concerned as a result of malicious individuals' attempts to abuse other people's names and credentials by creating fictitious profiles on social networks, which aims to violate their privacy. Therefore, in an effort to remove harmful users and phony identities from social networking environments, social network providers are attempting to detect them. Thus, we made the decision to install various apps in order to construct a driver support system.

b) Basic Definitions and Background: -

i) Support Vector Machine:

Support-vector machines, alternatively referred to as support-vector networks in the field of machine learning, are supervised learning models. They analyze data to perform tasks like regression and classification, employing specific learning techniques.

The illustration below showcases multi-class SVM classification. Often, the groups or sets that need to be distinguished are not separable by a straight line in the finite-dimensional space where the initial problem was introduced. Hyperplanes are sets of points in three-dimensional space where the dot product of each point with a given vector is zero. This group of vectors defines the hyperplane as an orthogonal or minimum group of vectors. To define the hyperplanes, you can use the database's feature vectors, which can be a linear combination of other vectors. When choosing this hyperplane, keep in mind that the sum of the terms represents the proximity of the test point to its database counterpart. The phrases get smaller as the distance gets bigger. Consequently, by adding up these kernels, we can determine how close each test point is to the data points in the two sets that need to be discriminated. This method produces a complicated point arrangement that may be projected onto any hyperplane, allowing for a more nuanced separation of sets that might not have been convex in the initial space.

While Support Vector Machine (SVM) is most commonly employed for classification problems in machine learning, it is also used for regression tasks and is a popular supervised learning technique overall. Finding the best decision boundary in n-dimensional space—also known as a hyperplane—to effectively divide classes is its goal. The key data points necessary to construct the hyperplane are located via support vector machines (SVMs). These support vectors—the data points that are nearest to the hyperplane—are essential for defining the separation line. Below is a schematic showing the use of a decision boundary or hyperplane to classify two separate groups:

Utilizing the support vectors—the data points that are geographically nearest to the hyperplane—is essential for locating the dividing line. This led to the theory that the initial finite-dimensional region may be better separated by projecting it onto a higher-dimensional space. Support vector machine algorithms guarantee efficient computation of dot products between pairs of input data vectors by defining these mappings in the context of a problem-specific kernel function. The computational strain is reduced as a result.

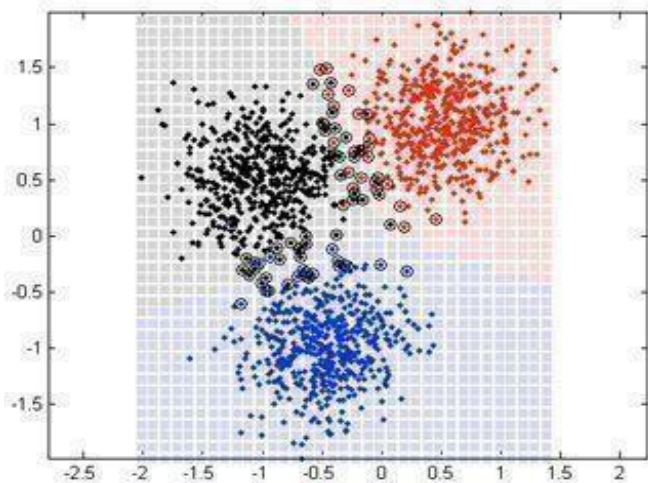


Fig.1 classification in case of multi class SVM.

ii) Random Forest:

Random Forest is a well-liked ML method that excels at both classification and regression due to its flexibility and ease of use. The supervised learning category includes this algorithm. It makes use of a network of decision tree models, as the name implies, with each branch taking into account a different set of features. A somewhat haphazard "forest" is born from this procedure.

A common method for training the "forest" of Decision Trees is the "bagging" approach. The basic idea behind bagging is that mixing different learning models improves overall performance. Basically, Random Forest builds a bunch of decision trees and then uses all of their forecasts to make one better prediction.

It is essential to understand decision trees in order to comprehend the Random Forest algorithm. To learn the class labels of instances using the dataset's features, Answering a set of questions is the job of a decision tree model. The decision-making process and the final tagging of data points with class labels are guided by these questions. This model works well because it is non-parametric and can deal with data that is heterogeneous (variables that are ordered, categorized, or a combination of both).

Moreover, decision trees are robust to outliers or label errors and are robust to irrelevant or noisy variables to some extent because they essentially execute feature selection. In brief, the Random Forest algorithm takes the following steps:

Select n samples at random, with replacement, from the training set. From the n sample, create a decision tree. every node. One to two k times, repeat the steps.

Combine the forecasts from every tree to determine the class label using a majority vote.

The fact that Random Forest is a series of tree-based models trained on haphazard portions of the training dataset is the inspiration for its name. The ensemble aspect of a Random Forest model reduces variance, which is one of the main advantages versus employing a single decision tree.

Overfitting the training data is less likely to occur in the Since each tree in the Random Forest ensemble is trained using a unique, randomly selected subset of the total training data, it is a kind of ensemble. Because of the great accuracy that can be achieved using this ensemble approach, Random Forest-based classifiers are often considered to be among the best models in various contexts.

c) Decision Tree:

Both product decision-making and suggestions for mergers or the development of new products are crucial [14]. After that, the decision tree is made and examined to make sure all outcomes are taken into account. In the second step of decision tree analysis, the tree is assessed. Weighing the options that are most appropriate for the current circumstance allows us to do this. For the outcome, a value estimate is calculated. Estimating the tree values is the third phase of the research. Estimating the outcome values that are crucial for making decisions is crucial. The value linked to the decision nodes is computed after the value of uncertain outcomes. The largest option is crucial for decision-making after the decision incentives have been measured.

Certainly, let me provide explanations for the points you mentioned:

i) Classification issues are often addressed by decision trees, a supervised learning technique. On the other hand, regression problems can also benefit from their use. The structure of this classifier is like a tree: the nodes inside the tree stand for dataset attributes, the branches for decision rules, and the leaf nodes for the outcome or class.

ii) The 2 most basic kinds of nodes in a decision tree are decision nodes and leaf nodes. There are several branches on decision nodes, which stand for many choices or conditions, and they are used to create decisions. In contrast, these decisions terminate at leaf nodes, which do not have any offshoots. Decision trees base their tests and choices on the features and attributes of the dataset that is provided. Visualizing all the potential outcomes of a decision or problem under certain parameters is its primary use.

iii) Its structure is like a tree, which is where the name "decision tree" comes from. The process begins at the root node, which is also the first point of decision-making. The decision tree gets its name from the fact that, as you go down the tree, it branches out into more nodes, forming a structure similar to a tree.

iv) When building trees, we employ the CART method, which stands for Classification and Regression Tree.

v) By asking a yes/no question, a decision tree automatically creates subtrees based on the answer.

"True Positive" (TP) means that the actual value is close to the projected value.

The model's prediction was also positive, and the actual result was positive. The anticipated value is the same as the actual value in a true negative (TN) situation.

Both the model's prediction and the actual value turned out to

be negative. Unwanted Reaction—A Type 1 Error The

estimated value turned out to be inaccurate.

A positive number was predicted by the model, while the actual result was negative. The Type 1 mistake is another name for it.

A type 2 error known as a false negative (FN).

It was a faulty prediction of the value.

Despite the model's negative prediction, the actual value turned out to be positive.

Alternatively called the Type 2 error.

Here is an example of a Type 2 error. To help you understand, I'll provide an example. So, supposing we have a dataset with 1000 categorization points. We got the confusion matrix below after we fitted a classifier to it.

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	560	60
	NEGATIVE	50	330

Fig.2 Confusion Matrix

Here are the possible values for the Confusion matrix:

“True Positive (TP) = 560; meaning 560 positive class data points were correctly classified by the model

True Negative (TN) = 330; meaning 330 negative class data points were correctly classified by the model

False Positive (FP) = 60; meaning 60 negative class data points were incorrectly classified as belonging to the positive class by the model

False Negative (FN) = 50; meaning 50 positive class data points were incorrectly classified as belonging to the negative class by the model”

		ss	
		positives	negative
		active	active

Table.1 Confusion Matrix

Correctly predicted positive values, also known as true positives (TP), occur when both the actual and anticipated values of a class are yes.

When both the actual and expected values of a class are zero, we say that the prediction was accurate; this is known as a true negative (TN).

In the case of false positives (FP), the predicted class is true even while the actual class is false. It's possible, for instance, that this passenger will actually live, even though the predicted class suggests otherwise.

When a true class is present but an expected class is absent, this is known as a false negative (FN). For instance, suppose you know the passenger will die based on the anticipated class but the actual class result shows that they survived.

The most basic and easily understood performance metric is accuracy, which is defined as the percentage of observations that were correctly predicted relative to the total number of observations.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

Precision - Precision is defined as the proportion of correctly anticipated positive observations to the total number of predicted positive observations.

$$Precision = \frac{TP}{TP + FP}$$

Recall, also known as sensitivity, is the percentage of correctly predicted positive observations relative to the total number of observations in the real class.

$$Recall = \frac{TP}{TP + FN}$$

F1 score—The F1 Score is the weighted average of REcall and Precision. Because of this, the score takes both the positive and negative outcomes into consideration.

$$F1\ Score = \frac{2 * Recall * Precision}{Recall + Precision}$$

Bayes Theorem: To determine the likelihood of a hypothesis, one uses previous information and the Bayes Theorem. This is decided by the conditional probability.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

POST Method:

Individuals, Goals, Plans, and Technology are the four key components that define how OSNs are used. People give an explanation of the communication platform that must be used. The goals will specify the reason for communicating across OSNs. Boosting social media engagement by drawing in more followers and users is the plan. Technology requires the most suitable tool to be used. All four of these elements work together to guarantee an effective OSN with a large subscriber base.

Accuracy of detection models:

To distinguish between real and fake nodes, most current detection algorithms exploit features of network graphs.

These methods don't take scalability into account. Graph methods are not as effective as machine learning algorithms and user metadata analysis when it comes to identifying bogus profiles. Nowadays, active machine learning algorithms do a better job at checking massive bots. All social networks could easily recognize posts made in bulk by a single node.

II.RELATED WORKS

Problem Statement:

False identities are a significant component of advanced and persistent threats and are also linked to other malevolent actions. The detection of phony social media profiles is the current problem. Techniques for identifying coordinated activity across a large number of accounts and techniques for analyzing individual accounts are two categories into which approaches to the identification of fraudulent social media accounts can be divided. The paper addresses the detection techniques for phony social media accounts and clarifies the role that fake identities play in ongoing advanced threats. To complete this project, we needed a collection of fictitious profiles to act as a comparative data set. The majority of recent studies in this field use fictitious, synthetic profiles for this reason.

Applications:

In real life activities approximately 1.0 billion users are using Instagram. It executes more than 10 million daily likes and shares. The growth of social networks has resulted in a very large number of false user profiles that leads to many of the profiles try to make friends with benign users with the ultimate goal of accessing privileged data. Social engineering in any online social networks the main cause of threats.

III.LITERATURE SURVEY

a) Description of Existing System:

i) Title: FAKE SOCIAL MEDIA PROFILE DETECTION

Our lives now revolve around social media. Everyone uses social media, whether it be for following celebrities, sharing eye-catching photos, or conversing with friends nearby or across the globe. It's a fantastic platform for communication and information sharing. However, everything has a drawback. Social media is becoming a permanent fixture in our lives, but occasionally it can cause issues. There are 330

million unique users on Twitter every month, and 145 million unique users per day. Facebook also adds 5,000 new users daily, or 6 new people every second. There is a deluge of data exchanged on Twitter daily. News, trending hashtags, and individuals' most recent adventures are all available on Twitter. During the limited 280 characters, people respond, like, comment, express themselves, and raise their ideas. While there are discussions on real topics, there are also occasionally rumors. These rumors cause tensions amongst various societal groups. In recent times, concerns about privacy, misuse, cyberbullying, and false information have surfaced. These are all bogus profiles doing these activities. In APT incidents, social media aliases are frequently used to propagate malware or links to it, build confidence, and obtain intelligence before the attack. These false personas are also employed in other nefarious endeavors.

There has been a lot of study into ways to identify when someone is posing as someone else on social media in an effort to stop these kinds of things. The taxonomy developed by Song et al. (2015) classifies methods for detecting fake social media accounts into two broad groups: those that examine each account independently (profile-based and graph-based techniques) and those that record the coordination of crowdsourcing, also known as Crowd Target. The authors concentrate on identifying the target items of crowdsourcing projects (such as posts, pages, and URLs) rather than trying to identify workers. The suggested approach is more resistant to detection evasion strategies since it can reliably discriminate between innocuous tweets and crowdsourcing with a true positive rate of up to 98%, even when they originate from the same account. It has been demonstrated that the following characteristics are discriminative: (i) the distribution of tweet times; (ii) the ratio of the most popular application; (iii) the number of unreachable retweeters; and (iv) the quantity of clicks obtained. Unfortunately, just like the methods mentioned above for identifying spamming operations, crowd-turfing detection methods also presume large-scale activity, making it difficult to identify small-scale activities conducted in conjunction with a targeted attack.

ii) Title: AUTOMATIC DETECTION OF FAKE PROFILES ON SOCIAL NETWORKS

Our lives now revolve around social media. Everyone uses social media, whether it be for following celebrities, sharing eye-catching photos, or conversing with friends nearby or across the globe. It's a fantastic platform for communication and information sharing. However, everything has a drawback. Social media is becoming a permanent fixture in our lives, but occasionally it can cause issues. There are 330 million unique users on Twitter every month, and 145 million unique users per day. Facebook also adds 5,000 new users daily, or 6 new people every second. There is a deluge of data exchanged on Twitter daily. News, trending hashtags, and individuals' most recent adventures are all available on Twitter. During the limited 280 characters, people respond, like, comment, express themselves, and raise their ideas. There are debates about actual subjects, but there are also rumors every once in a while. Various social groupings experience tensions as a result of these rumors. In recent times, concerns about privacy, misuse, cyberbullying, and false information have surfaced. These are all bogus profiles doing these activities. In APT incidents, social media aliases are frequently used to propagate malware or links to it, build confidence, and obtain intelligence before the attack.

These false personas are also employed in other nefarious endeavors.

There has been a lot of study into ways to identify when someone is posing as someone else on social media in an effort to stop these kinds of things. The taxonomy developed by Song et al. (2015) classifies methods for detecting fake social media accounts into two broad groups: those that examine each account independently (profile-based and graph-based techniques) and those that record the coordination of crowdsourcing, also known as Crowd Target. Instead of attempting to find workers, the writers focus on identifying the target elements of crowdsourcing projects, which include posts, pages, and URLs. The suggested approach is more resistant to detection evasion strategies since it can reliably discriminate between innocuous tweets and crowdsourcing with a true positive rate of up to 98%, even when they originate from the same account. It has been demonstrated that the following characteristics are discriminative: (i) the distribution of tweet times; (ii) the ratio of the most popular application; (iii) the number of unreachable retweeters; and (iv) the quantity of clicks obtained. Unfortunately, just like the methods mentioned above for identifying spamming operations, crowd-turfing detection methods also presume large-scale activity, making it difficult to identify small-scale activities conducted in conjunction with a targeted attack.

iii) Title: Using Machine Learning to Detect Fake Profiles in Online Social Networks

Social media is currently one of the simplest platforms for spamming or disseminating fake news. The most frequent threats in emails are those related to spamming. By observing how individuals interact with one another, a lot about human nature and its needs can be discovered. We can evaluate people's typical behavior and conversational topics to offer them improved services to users on a large scale. The same thing can be applied in a way that makes it possible to trick people with them. As an illustration, consider the following: a single message has the power to sway a significant number of people's opinions on a given subject, even when many of these persons are fictitious. One of the most popular places for people to meet new people, talk about what they're thinking, and post photos and videos online is on social media sites like Facebook, Twitter, and Instagram., discuss current events, and treasure old memories in the previous few years. Launched in 2006, Twitter is a microblogging platform that has gained immense popularity among users of online social networks. Users can publish messages, known as "tweets," that are limited to 140 characters. With 330 million active users, Twitter posts around 500 million times a day.

Because of its enormous popularity, spammers take notice of Twitter and exploit it for malicious purposes, such as disseminating abusive URLs within tweets, starting rumors, and sending unsolicited messages to other users. Researchers have also become interested in Online Social Networks (OSNs) because of their vast amounts of data, which can be mined and analyzed for insights into user behavior and the identification of unusual activity. Researchers have conducted a study to determine the most effective cognitive traits that predict customers' attitudes in order to forecast, assess, and explain customer loyalty towards an online brand community based on social media. These gaps are often exploited since it is exceedingly challenging to identify

phony human accounts. We think that these identity forgeries could be employed for further reasons:

Studies have been done to identify phony accounts that are the result of bot creation. On social media sites, machine learning (ML) is used to detect bots, but it is also used to determine the bot's objective. We can identify fake using a variety of techniques.

iv) Title: A MACHINE LEARNING ALGORITHM-BASED STUDY ON THE IDENTITY OF FAKE PROFILES ON SOCIAL MEDIA WEBSITES

A method for spotting harmful activity on social media platforms has been proposed by Nambour Sravya, Chavana Sai Pranitha, and S. Saraswathi in this work. Social media platforms pose a major risk to online safety these days. Numerous models are offered to detect bogus profiles. To zero in on Sybil and the troll's identities, they used a machine learning technique. With the help of the supervised machine learning algorithm, the problem can be solved. Advanced techniques are used by Sybil and troll accounts. First, data sets are gathered from huge data blogs and saved. If malicious activity is detected in the data, it is cleaned and stored once again. The cleansed data then displays the cleaned false individualities, with the missing portions being fake individualities. Data is kept in a non-relational database prior to the clean store procedure for future use and to aid in the removal of the false profile. should stay away from malicious activity, spam messages, and online bullying—activities that are primarily carried out by phony profiles. The social network communities' privacy regulations are being questioned by these actions. False information is disseminated on social media platforms by these phony personas. There has been a lot of study done in this field to identify phony profiles, duplicate accounts, spam, and bot accounts. A machine learning technique was used to detect most of the bogus accounts, and it was successful. In this research, we take a look at how social media platforms are using machine learning to identify false profiles.

III.METHODOLOGY

a) SYSTEM ARCHITECTURE DIAGRAM:

DESCRIPTION OF ALGORITHM:

One of the most popular supervised learning methods in machine learning is Support Vector Machine (SVM). You can use this classifier to solve problems with classification and regression together. Classification issues, however, are its specialty. Support vector machines (SVMs) primarily aim to find the optimal decision boundary, sometimes known as a hyperplane, to divide the n-dimensional space into classes. To do this, SVM uses a set of predetermined data points called "support vectors" and other important points that define the hyperplane extensively. The algorithm is called a "Support Vector Machine" because it is designed to handle these extreme cases, which are called support vectors. When it comes to classification jobs, support vector machines (SVMs) are a strong tool. They use support vectors to determine the optimum hyperplane to segregate classes in high-dimensional spaces.

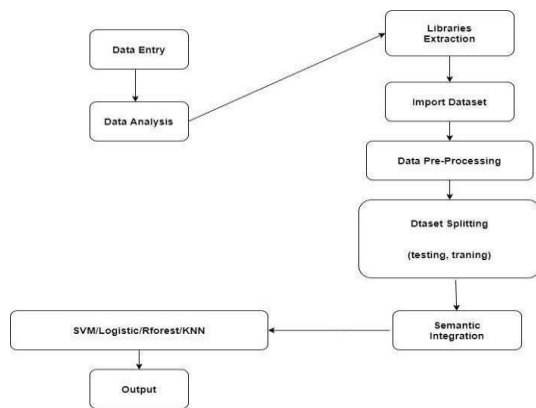


Fig.3 System Architecture

Look at the following figure, which uses a decision boundary (sometimes called a hyperplane) to divide the world into two groups:

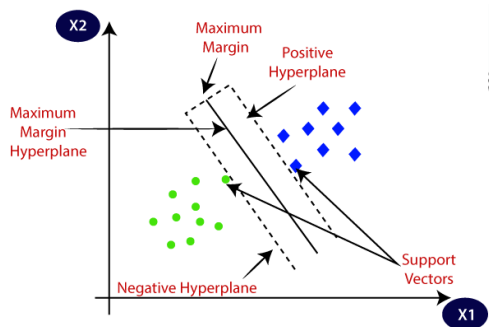


Fig.4 Hyperplane

Types of SVM: SVM can be of two types:

Linear support vector machine (SVM) classifiers apply linearly separable data, which can be separated into two groups by a single straight line, to create classification models. Non-Linear SVM is employed for non-linearly separated data when a dataset cannot be categorized using a straight line. This instance makes use of a Nonlinear Support Vector Machine (SVM) classifier.

Vertical line: Finding the optimum decision boundary or line for data point classification is essential in n-dimensional space, because there may be multiple such lines. This perfect boundary is known as the SVM hyperplane. The hyperplane's dimensions are defined by the features in the dataset; for instance, a straight line will represent the hyperplane's dimensions if the image exhibits two qualities. Furthermore, in the presence of three properties, a hyperplane is defined as a plane with two dimensions.

We consistently incorporate the maximum margin, or distance between data points, when we create hyperplanes.

A support vector is a data point or vector that influences the position of the hyperplane and is nearest to it. These vectors aid in maintaining the hyperplane's position, which is why they are known as support vectors.

INPUT: Raw text files consist of RNA sequence

OUTPUT: Classifying the new RNA into covid or not.

1. Extract the RNA sequence from the raw text files
2. Building the data from count vectorization

3. Standardization of the data using sklearn package.
4. Divide the data set into two sets: training and testing.
5. Plot the data set data in N dimensional space
6. find the distance by using the formula say
7. determine the hyperplane with the help of weight vector(w) and bias(b).
8. If > 0 then positive
9. If < 0 then negative
10. Repeat the step 6,7,8 until the find the hyperplane with maximum margin.
11. Find the accuracy using confusion matrix.
12. Classify new RNA sequence using above model.

RANDOM FOREST:

One of the most popular machine learning algorithms, Random Forest is great for both classification and regression because of its flexibility and user-friendliness. Supervised learning is the purpose of this algorithm. Its moniker alludes to the fact that it employs bagged decision-tree models, which randomly generate a forest by splitting on a subset of features at each split. Decision Trees, typically trained using the "bagging" technique, make up the forest it generates. The core principle of the bagging method is that combining many learning models increases the total outcome. If you want a more precise and dependable prediction, Random Forest builds multiple decision trees and then combines them.

$$f_i = \frac{\sum_{j: \text{node } j \text{ splits on feature } i} n_{ij}}{\sum_{k \in \text{all nodes}} n_{ik}}$$

The fundamental idea behind decision trees can be better understood with a more in-depth knowledge of the RF algorithm. Based on the features in each dataset, the decision tree model learns a set of questions to decide the class labels of the instances. Because it is nonparametric and can handle heterogeneous data (variables that are ordered, categorized, or a mix of both), this model performs admirably. Decision trees essentially carry out feature selection, making them powerful against outliers or label errors and somewhat resilient against irrelevant or noisy factors. In a nutshell, the Random Forest algorithm does the following: Replace n random samples taken from the training set. Using each node to construct a decision tree. With the n-sample, repeat steps 1–2 k times. A majority vote is used to determine the class label after adding up all of the predictions from each tree. Some decision trees might be right about the output prediction and others could be wrong because the random forest uses multiple trees to estimate the dataset's class. If you combine them all, though, you'll see that each tree delivers an accurate forecast.

As a result, a better Random forest classifier can be achieved by assuming the two following: • The classifier can only make accurate predictions (rather than guesses) if the dataset's feature variable has actual values. • The correlation between the predictions made by different trees must be negligible.

For example, Random Forest is a popular machine learning technique that falls within the category of supervised learning. We use it for regression and classification because it's easy to use and versatile.

- Its training time is lower than that of competing methods. It works well with big datasets and makes accurate predictions.
- Its name suggests it is made up of decision tree models, which create a forest and introduce some randomness.
- It can keep accuracy even when a lot of data is missing.

Below is a graphic that explains the Random Forest algorithm:

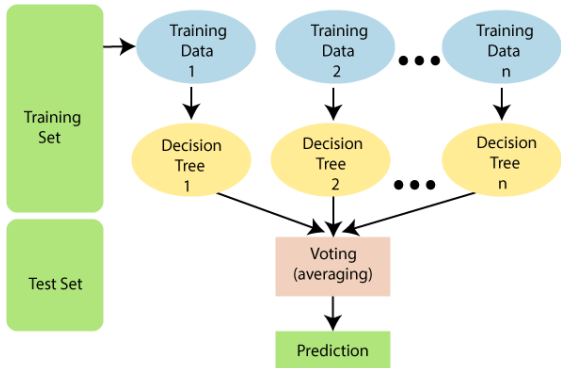


Fig.5 SVM Tree

Then, in a decision tree, the weight of each feature is determined by: f_i sub(i) = the importance of feature i

- n_i sub(j) = the importance of node j

These can then be normalized to a value between 0 and 1 by dividing by the sum of all feature importance values:

- The final feature importance, at the Random Forest level, is its average over all the trees. The sum of the feature's importance value on each trees is calculated and divided by the total number of trees: RFf_{ii} = the importance of feature i calculated from all trees in the Random Forest model
- $normf_{iij}$ = the normalized feature importance for i in tree j
- T = total number of trees

Decision Tree:

Even though they are a kind of supervised learning, decision trees are most often used to address categorization issues. On the other hand, regression problems can also be solved with their help. Within this tree-structured classifier, each outcome is represented by a leaf node. The decision rules are represented by branches, and the dataset attributes are represented by internal nodes. There are two distinct kinds of nodes in a decision tree: decision nodes and leaf nodes.

In comparison to leaf nodes, which merely display decision outcomes and do not possess any extra branches, decision nodes are capable of making a wide variety of decisions and possess numerous branches. The characteristics of the provided dataset form the basis of the decisions and testing. It is a visual aid that, given a set of criteria, displays every possible solution or course of action. A decision tree gets its

name from the fact that, similar to a tree, it has a central node from which branches extend outward.

What is the inner workings of the Decision Tree algorithm?

The process of a decision tree begins at the root node in order to forecast the class of the dataset. The method continues to the next node by comparing the values of the root attribute with those of the record (actual dataset) attribute.

As the procedure moves on to the next node, it compares the value of its attributes to those of the previous sub-nodes once again. This process will be repeated until it reaches the node that attaches the tree's leaves.

To gain a better grasp of the whole procedure, consider using the following algorithm:

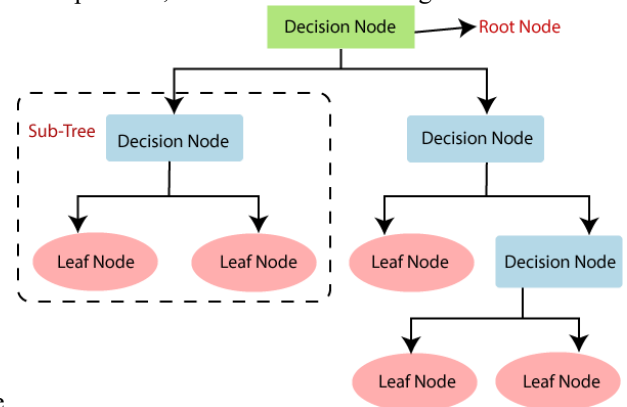
As per S's instructions, one must start the tree at the root node, which houses the whole dataset.

Two, use the Attribute Selection Measure (ASM) to find the dataset's best attribute.

The third step is to partition the set S into subsets that include all the potential values for the paramount qualities.

Fourth, you'll want to make sure that the decision tree node centered on the best attribute.

Fifth, make additional decision trees from the dataset by recursively creating them. subsets established in step 3. The process should be repeated until further classification of the nodes is impossible; the last node at this stage is called a leaf



node.

Fig.6 Decision Tree

The decision tree starts at the node known as the root. It shows the full dataset before it is partitioned into two or more sets of comparable data.

The tree can only be further segmented up to the leaf node level; these nodes serve as the final output.

Splitting is the process of dividing the root node or decision node into sub-nodes according to the given conditions. A tree that has been cut from another tree into smaller trees.

When you prune a tree, you take off any branches that aren't needed or wanted.

The two types of nodes that make up a tree are called "parent" and "child" nodes, accordingly. Root nodes are parent nodes. When building a decision tree, the most common issue is deciding which attribute is best for the parent node and all of its offspring. An approach called attribute selection

measure (ASM) was devised to tackle these problems. This metric allows us to pick the best characteristic for the nodes of the tree in a flash. There are two main approaches to ASM, and they are:

After a dataset is partitioned based on a feature, the information gain can be determined by tracking the changes in entropy. The quantity of information a feature provides about a class is determined by it.

The entropy of a system is a measure of how unclear a certain property is. It explains how unpredictable the data is. Assume $i \in \{1, 2, \dots, J\}$ and let p_i be the fraction of items in the set that are labeled with class i in order to calculate the Gini impurity for a set of objects with J classes.

$$I_G(p) = \sum_{i=1}^J p_i(1 - p_i)$$

Evaluation Parameters

Number of forecasts/Total number of predictions equals efficiency/accuracy. Accuracy - 1-Percent Error = $100 * 100$

Confusion Matrix: This method is used to summarize how well a classification algorithm performs. True Positive Rate, or TPR TPR is equal to TP divided by (TP+FN). False Positive Rate, or FPR FPR is equal to FP/(FP+TN). True Negative Rate, or TNR TNR is equal to TN/(FP+TN). False Negative Rate, or FNR FNR minus 1-TPR Precision: The precise number of actual positive hits, or the number of correct hits discovered, among the returned hits.

$TP / (TP + FP)$ equals precision.

PROPOSED METHODOLOGY:

There is a wealth of personal information stored in each social media profile, including gender, number of friends, comment count, educational background, employment status, and more. There is both public and private information here. We have relied solely on publicly available data to identify phony profiles on social media platforms since private information is inaccessible. On the other hand, social media platforms can use users' private profile data for detection purposes in our suggested system without compromising users' privacy. In order to distinguish between authentic and fraudulent profiles, we have taken these data into account as profile attributes. Following are the procedures that have been implemented to identify false profiles.

A. The classification method is applied after all features have been selected. In this part, we'll go over the classification algorithm. It is important to pick features with care, making sure they don't rely on other features and picking ones that can improve categorization efficiency. Our chosen characteristics are detailed in the Overview of Proposed Work.

In order to train the classification algorithm, it is necessary to have a dataset that includes both real and fake profiles that have been previously identified, after the attributes have been appropriately selected. Barracuda Labs is a privately owned business that offers storage, networking, and security solutions through cloud services and network appliances.

We created the actual profile dataset, while they supply the phony one. We talk about the dataset collecting.

- Using the ones selected in step 1, it is essential to extract the true and fake qualities from the profiles. We detail the process by which we retrieved the attributes we used. Rather than going through the scraping process, social media organizations that are interested in utilizing our program can easily extract the features from their database. In order to conduct research on the topic of fraudulent accounts, we turned to profile scraping since no publicly available social network dataset exists specifically for this purpose.
- After that, a dataset is generated that has profiles that are both legitimate and fraudulent. Using 80% of the true profiles and 20% of the false profiles, respectively, this dataset is used to create a training dataset and a testing dataset. We use two datasets to assess the classification method's performance: one including 922 profiles and the other containing 240 profiles.

After the testing and training datasets have been compiled, the training dataset is input into the classification algorithm. With the knowledge it gained during training, it should be able to accurately assign classes to the testing dataset. Step 6: The levels from the testing dataset are removed and the proposed framework is left to determine them. What the trained classifier is supposed to do is proposed activity. The classifier's effectiveness is determined by dividing the total number of predictions by the number of right predictions. This part displays the categorization outcome. We have tested three different algorithms for categorization and compared their performance. Firstly, the data of different users have been collected from the Instagram with some useful columns.

- Then the dataset is preprocessed by using the data cleaning techniques that fill the missing values and also the outlier detection.
- Then the preprocessed data of fake users and genuine users is splitter into 80% training and 20% testing data by Scipy and Scikit learn modules in Python.
- Then feature column are extracted from the data according to the weight they play on finding an account credibility.
- Then the classifier Random Forest is trained with the training data.
- By using the StratifiedKfold and the GridsearchCV in the Random forest, the best classifier function is traced out.
- Finally the model is tested with the test data and the accuracy is built as 90%.
- Then the Confusion matrix and also Learning curve is given for result analysis.

DESCRIPTION OF DATASETS AND TOOLS:

Dataset:

We looked at an Instagram dataset that included both real and phony users. Some of the attributes that are part of the dataset are the amount of friends, followers, and statuses. There are two sections in the dataset: training and testing. A training dataset is utilized to train classification algorithms, while the testing dataset is employed to evaluate the algorithm's

performance. A training dataset is prepared using 80% of the dataset's profiles (real and fake), while a testing dataset is prepared using 20% of the profiles. Name: Social network fake account dataset Source: <https://www.kaggle.com/bitandatom/social-network-fake-accountdataset>

S.No	Attribute	Description
1.	Profile ID	The Profile ID of account holder
2.	Profile Name	The name of the account holder
3.	Status Count	The number of tweets made by the account
4.	Followers Count	The number of followers for the account
5.	Friends Count	The number of friends for the account
6.	Location	The location of the account holder
7.	Created Date	The date the account was created
8.	Share count	The number of shares done by account holder
9.	Gender	The Gender of the account holder
10.	Language Code	The language of account holder

TABLE.2 DATASET

The attributes that are taken into account for identifying fraudulent profiles are included in the table above, along with descriptions of each attribute.

V.RESULTS AND DISCUSSIONS

a) Stepwise Description of Results:

The vast majority of social media profiles (or accounts) contain extensive personal information, such as occupation, degree, number of friends, gender, and comments. Some of this data is available to the public, while some of it is not. We have solely relied on publicly available data to detect fake social media accounts because private information is not accessible. However, social media businesses can utilize users' private information for detection without violating privacy restrictions if they adopt our recommended system. These details have been taken into account as profile criteria for identifying authentic profiles. The following are the procedures that we used to identify phony profiles. The first step in the detection procedure is choosing the profile that has to be examined.

1. Following the profile selection, the appropriate attributes, or features, are chosen, and the classification method is then applied to them.
2. The trained classifier receives the extracted properties.
3. As new training data is fed into the classifier, it is regularly trained. The classifier establishes if the profile is authentic or fraudulent.
4. Since the classifier might not have classified the profile with 100% accuracy, it receives feedback on its performance. As in the event that the profile is recognized.
5. As time goes on, more training data are added, and the classifier's accuracy in identifying phony profiles improves. This procedure is repeated.

b) Test e results:

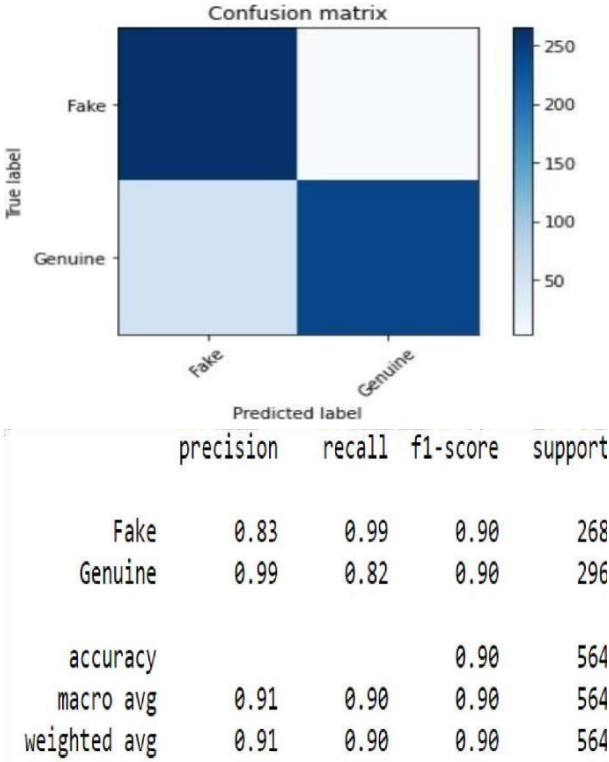


Fig.7 Confusion Matrix Confusion matrix

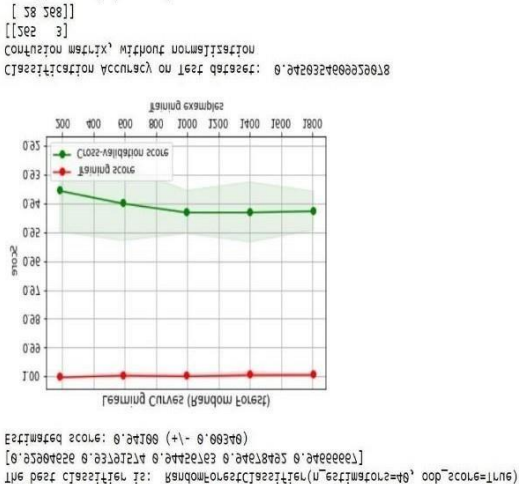


Fig.8 Learning curve

VI. CONCLUSION

It is a reality that anyone can create fake profiles on social media websites. People are starting to embrace it as a standard way to communicate, which has led to a dramatic increase in fake profiles. With the proliferation of detailed user profiles on social media platforms, there is a deluge of potentially manipulable data at anyone's fingertips. The use of social engineering and fake profiles has increased the risk of data theft and the delicate manipulation of individuals for malicious ends. We examined Random Forest techniques for identifying bogus profiles in preexisting social networks in this article. employed machine learning techniques.

VII. FUTURE WORK

With the help of our provided framework, we can detect false profiles in any social media platform using the Random Forest technique Classifier with an efficiency of up to 90%. When

applied to posts and profiles, Natural Language Processing (NLP) methods and Neural Networks can enhance Fake Profile Identification.

REFERENCES

- [1].Matt. Social Media Comparison Info graphic.2014.at
 - [2].Social Media comparison info graphic <https://leveragenewagemedia.com/blog/social-media-info-graphic/>.
 - [3].Tumblr. URL:www.tumblr.com/.
 - [4]Foursquare.URL:<https://foursquare.co/>.
 - [5].R.Nithin Reddy &Nitesh Kumar,“Automatic Detection of Fake Profiles in Online Social Networks”, Department of Computer Science and Engineering, National Institute of Technology, Rourkela.
 - [6].J. H. Parmelee and S. L. Bichard, Politics and the twitter revolution: How tweets influence the relationship between political leaders and the public, Lexington books, 2011.
 - [7].L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, “All your contacts are belong to us: automated identity theft attacks on social networks,” in Proceedings of the 18th international conference on World wide web, pp. 551– 560, Madrid, Spain, 2009.
- View at: Publisher Site | Google Scholar
- [8]. L.Jin, H. Takabi, and J. B. Joshi, “Towards active detection of identity clone attacks on online social networks,” in Proceedings of the the first ACM conference, p. 27, San Antonio, TX, USA, February 2011. View at: Publisher Site | Google Scholar
 - [9].Conti, R. Poovendran, and M. Secchiero, “FakeBook: Detecting fake profiles in on- line social networks,” in Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012, pp. 1071–1078, Turkey, August 2012.View at: Google Scholar
 - [10].Z. Shan, H. Cao, J. Lv, C. Yan, and A. Liu, “Enhancing and identifying cloning attacks in online social networks,” in Proceedings of the the 7th International Conference, pp. 1–6, Kota Kinabalu, Malaysia, January 2013. [1] Michael Fire et al. (2012). "Strangers intrusion detection-detecting spammers and fake profiles in social networks based on topology anomalies." Human Journal 1(1): 26-39.Günther, F. and S. Fritsch (2010). International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 6 (2018) pp. 4133-4136 © Research India Publications. <http://www.ripublication.com>
 - "neuralnet: Training of neural networks." The R Journal 2(1): 30-38
 - [11].Dr. S. Kannan, Vairaprakash Gurusamy,“Preprocessing Techniques for Text Mining”, 05 March 2015 .
 - [12].Shalinda Adikari and Kaushik Dutta, Identifying Fake Profiles in LinkedIn, PACIS 2014 Proceedings, AISel
 - [13].Z. Halim, M. Gul, N. ul Hassan, R. Baig, S. Rehman, and F. Naz,“Malicious users’ circle detection in social network based on spatiotemporal co-occurrence,” in ComputerNetworks and Information Technology(ICCNI),2011 International Conference on, July, pp. 35–390.
 - [14].Liu Y, Gummadi K, Krishnamurthy B, Mislove A,” Analyzing Facebook privacy settings: User expectations vs. reality”, in: Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference,ACM,pp.61–70.
 - [15].Mahmood S, Desmedt Y,” Poster: preliminary analysis of google?’s privacy. In: Proceedings of the 18th ACM conference on computer and communications security”, ACM 2011,PP 809-81.

D.CERTIFICATE

