A

Project Seminar Report

on

**SIGNATURE CLASSIFICATION AND ANALYSIS OF  
VARIOUS ALGORITHMS**

Submitted for partial fulfilment of the requirements for the award of the degree of

**BACHELOR OF ENGINEERING**

in

**COMPUTER SCIENCE AND ENGINEERING**

By

**K Sai Vikhyath (2451-16-733-132)**

**Y Snehal Reddy (2451-16-733-168)**

**K Sravan Kumar Reddy (2451-16-733-176)**

Under the guidance of

**Mr. K Murali Krishna**

Assistant Professor

Department of CSE



**M.V.S.R. ENGINEERING COLLEGE**

Department of Computer Science and Engineering

(Affiliated to Osmania University & Recognized by AICTE)

Nadergul, Saroor Nagar Mandal, Hyderabad – 501 510

Academic Year: 2019-20

**M.V.S.R. ENGINEERING COLLEGE**

Department of Computer Science and Engineering

(Affiliated to Osmania University & Recognized by AICTE)

Nadergul, Saroor Nagar Mandal, Hyderabad – 501 510



# CERTIFICATE

*This is to certify that the project work entitled “Signature Classification and Analysis of Various Algorithms” is a bonafide work carried out by* ***Mr. K Sai Vikhyath (2451-16-733-132), Mr. Y Snehal Reddy (2451-16-733-168), Mr. K Sravan Kumar Reddy (2451-16-733-176)*** *in partial fulfilment of the requirements for the award of degree of* ***Bachelor of Engineering*** *in* ***Computer Science And Engineering*** *from* ***M.V.S.R. Engineering College,*** *affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2019-20. under our guidance and supervision.*

*The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.*

**Internal Guide Head of the Department**

Mr. K Murali Krishna Dr. J. Prasanna Kumar

Assistant Professor Professor & Head

Department of CSE Department of CSE

MVSREC. MVSREC.

# DECLARATION

This is to certify that the work reported in the present project entitled “Signature Classification and analysis of various algorithms” is a record of bonafide work done by us in the Department of Computer Science and Engineering, M.V.S.R. Engineering College, Osmania University. The reports are based on the project work done entirely by us and not copied from any other source.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

K Sai Vikhyath Y Snehal Reddy K Sravan Kumar Reddy  
(2451-16-733-132) (2451-16-733-168) (2451-16-733-176)

# ACKNOWLEDGEMENT

We would like to express our sincere gratitude and indebtedness to our project guide **Mr. K Murali Krishna, Assistant Professor** for his valuable suggestions and interest throughout the course of this project.

We are also thankful to our principal **Dr. G. Kanaka Durga** and **Dr. J. Prasanna Kumar,** Professor and Head, Department of Computer Science and Engineering, MVSR Engineering College, Hyderabad for providing excellent infrastructure and a nice atmosphere for completing this project successfully as a part of our B.E. Degree (CSE).

We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank our family for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

K Sai Vikhyath (2451-16-733-132)

Y Snehal Reddy (2451-16-733-168)

K Sravan Kumar Reddy (2451-16-733-176)

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

# VISION

* To impart technical education of the highest standards, producing competent and confident engineers with an ability to use computer science knowledge to solve societal problems.

**MISSION**

* To make learning process exciting, stimulating and interesting.
* To impart adequate fundamental knowledge and soft skills to students.
* To expose students to advanced computer technologies in order to excel in engineering practices by bringing out the creativity in students.
* To develop economically feasible and socially acceptable software.

**PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

The Program Educational Objectives of undergraduate program in Computer Science & Engineering are to prepare graduates who will:

1. Obtain strong fundamentals concepts, technical competency and problem-solving skills to generate innovative solutions to engineering problems.
2. Continuously enhance their skills through training, independent inquiry, professional practices and pursue higher education or research by adapting to rapidly changing technology.
3. Advance in their professional careers including increased technical, multidisciplinary approach and managerial responsibility as well as attainment of leadership positions thus making them competent professionals at global level.
4. Exhibit commitment to ethical practices, societal contributions and lifelong learning.

**(A) PROGRAM OUTCOMES (POs)**

At the end of the program the students (Engineering Graduates) will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

1. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principle and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
2. **Lifelong learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**(B) PROGRAM SPECIFIC OUTCOMES (PSOs)**

13**.Efficient coding:** an ability to analyse a problem, design the algorithm and optimally code its solution.

14**.Software deployment:** an ability to identify & define computing requirements to test, implement and maintain software product.

**PROJECT WORK – I**

**(PW 761 CS)**

# COURSE OBJECTIVES AND OUTCOMES

**COURSE OBJECTIVES**

* To enhance practical and professional skills.
* To familiarize tools and techniques of systematic literature survey and documentation
* To expose the students to industry practices and team work.
* To encourage students to work with innovative and entrepreneurial ideas

**COURSE OUTCOMES**

1. Demonstrate the ability to synthesize and apply the knowledge and skills acquired in the academic program to the real-world problems.
2. Evaluate different solutions based on economic and technical feasibility
3. Effectively plan a project and confidently perform all aspects of project management
4. Demonstrate effective written and oral communication skills

# ABSTRACT

In this project, we try to provide an optimal solution for verifying online signatures using machine learning, particularly Deep Learning. We have implemented about 35 features which can be fed into the neural network to identify the minute differences between genuine signatures and forged signatures of 40 users from the SVC2004 dataset.

We have developed two Deep Learning models, one to identify users and the other to classify whether the signature is genuine or forged. These models can be used in real-time to detect forged signatures efficiently and effortlessly with cutting edge technologies, like Deep Learning, with an accuracy of 95% on unseen data. These models, collectively have about 41 million parameters, can be directly used through a web app interface built with Flask, a Python web framework.

This web app is hosted on Heroku cloud platform (PAAS), to enable users to use pre-trained Deep Learning models for signature verification without concerning the end user with all the intricacies of using a state-of-the-art technology.

**Keywords:** Deep Learning, Models, Signatures, SVC2004, Neural Network, Cloud Platform.

K Sai Vikhyath (2451-16-733-132)

Y Snehal Reddy (2451-16-733-168)

K Sravan Kumar Reddy (2451-16-733-176)

# TABLE OF CONTENTS

**CONTENT** **PAGE NO.**

[Certificate i](#_Toc24358461)

[Declaration ii](#_Toc24358462)

[Acknowledgement iii](#_Toc24358463)

[Vision & Mission iv](#_Toc24358464)

[Course Objectives And Outcomes vii](#_Toc24358465)

[Abstract viii](#_Toc24358466)

[Table Of Contents ix](#_Toc24358467)

[List Of Figures x](#_Toc24358468)

[1. Introduction 1](#_Toc24358469)

[1.1 Problem Statement 1](#_Toc24358470)

[1.2 Objectives 1](#_Toc24358471)

[1.3 Motivation 2](#_Toc24358472)

[1.4 Existing System 2](#_Toc24358473)

[1.5 Proposed System 2](#_Toc24358474)

[2. Literature Survey 3](#_Toc24358475)

[2.1 Existing Systems 3](#_Toc24358476)

[2.2 Proposed System 6](#_Toc24358477)

[2.3 Applications 10](#_Toc24358478)

[3. Software Requirements Specification 10](#_Toc24358479)

[3.1 Software Requirements 14](#_Toc24358480)

[3.2 Hardware Requirements 14](#_Toc24358481)

[References 15](#_Toc24358482)

# LIST OF FIGURES

**Fig.no Figure Pg. no**

[2.1 Angle of Tangent at two end points 4](#_Toc24361405)

[2.2 Logistic Regression Example 7](#_Toc24361406)

[2.3 K-Nearest Neighbours Example 8](#_Toc24361407)

[2.4 Support Vector Machine Example 8](#_Toc24361408)

[2.5 Decision Tree Example 9](#_Toc24361409)

[2.6 Neural Network Example 9](#_Toc24361410)

1. INTRODUCTION

## 1.1 PROBLEM STATEMENT

Biometrics are being used in many fields since a long time. One of the most commonly used biometric is a signature. Signature is a natural biometric which has been used since a long time for identifying a person. Signatures are used in most of the fields like Banks. With the advent of mobiles and tablets, signatures are also signed on tablets and mobiles. Such signatures which are signed on an electronic touch pad or a tablet is called an online signature. Signature verification techniques utilize many different characteristics of an individual’s signature in order to identify that individual. The advantages of using such authentication techniques are:

1. Signatures are widely accepted by society as a form of identification and verification.
2. Information required is not sensitive.
3. Forging of one’s signature does not mean a long-life loss of that one’s identity.

Features considered for validating a signature are static features and dynamic features. Static features are those, which are extracted from signatures that are recorded as an image whereas dynamic features are extracted from signatures that are acquired in real time.

In this project we aim to verify online signatures accurately by analysing various Machine Learning techniques which can provide high efficiency in determination of the verity of the signature. Our project aims at discovering an optimal algorithm to detect frauds in online signatures at a minimal cost.

It is important to take into account external factors when investigating a signature veriﬁcation technique. Nowadays signature veriﬁcation applications are used in our daily lives and will be exposed to human emotions. The system has to give reliable accuracy in verifying an individual’s signature even if user is under different emotions.

## 1.2 OBJECTIVES

The main goal of this project is to classify the signatures accurately using various machine learning algorithms and suggest the most optimal algorithm to use.

## 1.3 MOTIVATION

The motivation behind the project is the growing need for a fool proof signature verification scheme which can guarantee maximum possible security from fake signatures. The idea behind the project is also to ensure that the proposed scheme can provide better performance than already established online signature verification schemes. The prospect of minimizing the response time and to provide the optimal algorithm for the problem has also been a driving force in the commencement of this project.

## 1.4 EXISTING SYSTEM

There are two different kinds of signatures. First being offline-signatures which are signed using a pen and paper. Second being online-signatures which are signed on a tablet or an electronic touch pad. Many systems have been developed to verify offline signatures when compared to online signatures. The algorithms used previously consists of Dynamic Time Warping technique, Fixed Point Arithmetic, Outline detection and representation, Recurrent neural networks, Neural networks with Back Propagation. Existing system only used a single approach to address the problem. The features considered were mostly static features with minimal dynamic features. Models are primarily based on similarity measures between the enrolled signature and the signature to be verified.

## 1.5 PROPOSED SYSTEM

In our proposed signature verification system, we split the SVC dataset into train, cross-validation and test sets. The dataset is pre-processed to remove inconsistency in the dataset. Feature extraction is applied on the dataset to retrieve the static features like the mean of the co-ordinates of the signature and dynamic features like velocity, pressure, number of pen lift during the signature. A dataset is created with these features in it. We then apply Machine Learning algorithms like Logistic Regression, K-Nearest Neighbours, Support Vector Machine, Decision Tree Classifier to classify the signatures and estimate the evaluation metrics for each algorithm. Based on the evaluation metrics we would analyse the optimal solution for the problem. The implementation is done using Python 3.

1. LITERATURE SURVEY

## 2.1 EXISTING SYSTEMS

In human life security takes important role. Nowadays it’s the basic fundamental of all systems developed. For this purpose, biometric authentication system got a lot of importance. Biometric authentication systems are secure, easy to use, easy to develop, uses basic techniques of signal processing and cheap to build. This improves the familiarity of biometric authentication system. Among these techniques signature verification is the most famous one because of cheap data acquisition devices. We can see the use of on-line signature verification in every kind of real-time applications such as credit card transactions, document flow applications, and identity authentication prior to access of sensitive resources. There have been several studies on on-line signature verification algorithms.

Here we describe about the previous work in the field of signature verification. The online signature verification techniques can be classified into two broad areas.

1. Using features extracted from the visible parts of the signature.

2. Using features extracted from virtual strokes or invisible parts of the signature (the parts that are not created but are imagined to be created).

### 2.1.1 USING VARIABLE LENGTH SEGMENTATION AND HIDDEN MARKOV MODELS

In paper [1], Shafiei introduced a new on-line handwritten signature verification system using Hidden Markov Model (HMM). The system proposed by him is based on variable length segmentation of signatures in HMM model for on-line signature verification. To achieve this, he segmented each signature at its perceptually important points. Then after applying some pre-processing, he associated to each segment a scale and displacement invariant feature vector.

The result of segmentation is a number of variable length segments for each signature. Each segment is now characterized by location of its most significant point in the signature. Features to be extracted are pressure variance and two angles of tangent lines to curve of segment in two segment end points.

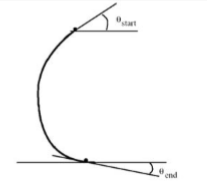


Figure 2.1 Angle of Tangent at two end points

Finally, the resultant sequence is then used for training an HMM to achieve signature verification. For each signer an HMM is trained using 5 genuine signatures. Assuming mixture of 10 Gaussians for emission probabilities for this HMM. The number of states of each HMM model equals to 0.5 times the average number of segments that is computed for each signature in the training set.

### 2.1.2 FIXED POINT ARITHMETIC METHOD

This technique is based on geometrical features which are based on two vectors which represent the envelope description and the interior stroke distribution in polar and Cartesian co-ordinates [2].

### 2.1.3 OUTLINE DETECTION AND REPRESENTATION

The outline is calculated by means of morphological operations. First, we apply a dilatation in order to reduce the signature variability and, afterward, a filling operation is applied to simplify the outline extraction process. When several objects are detected after filling, a horizontal dilatation is performed until all the objects are connected. The outline is represented as a sequence of its Cartesian coordinates (Xt, Yt) t T=1 being its length. This sequence follows the counter clockwise and starts in the point (X1, Y1) = Cx, max (Yt | Xt=Cx), (Cx, Cy) being the geometric centre of the outline [2].

### 2.1.4 NEW EXTREME POINTS WARPING TECHNIQUE

In this paper [3], Feng proposed a new warping technique for the functional base approach in signature verification. Dynamic time warping (DTW) is the commonly used warping technique. There are two common methodologies to verify signatures: the functional approach and the parametric approach so the functional based approach was originally used in application speech recognition and has been applied in the field of signature verification with some successful accuracy two decades ago. The new warping technique he proposed, named as extreme points warping (EPW). It was proved that this method is adaptive in the field of signature verification than DTW in the presence of the forgeries. In the functional approach, a straightforward way to compare two signal functions is to use a linear correlation.

It has the following two problems:

1. Due to difference of overall signal duration.

2.Due to existence of non-linear distortions within signals.

For a signal function, the signal duration is the same for different samples even from the same signer. In addition, distortions occur non-linearly within the signals for different signings. A non-linear warping process needs to be performed before comparison to correct the distortion. An established warping technique used in speech recognition is dynamic time warping (DTW). The use of DTW has also become a major technique in signature verification for the past two decades. Though DTW has been applied to the field with success, it has some drawbacks.

DTW has two main drawbacks when applied in signature verification:

1. It has heavy computational load,

2. Another is warping of forgeries.

The first drawback is a known problem in case of speech recognition, because DTW performs nonlinear warping on the whole signal. For this method, the execution time is proportional to the square of the signal size; define boundary conditions in the DTW matching matrix to reduce the computation time. The second drawback, however, is not well documented in the past, but still got good accuracy and results.

A new warping technique called EPW replaced the commonly used DTW. Instead of warping the whole signal as DTW does, EPW warps a set of selective points. We achieve the goal of warping the whole signal through matching the EPs and warping the segments linearly. Since EPW warps only EPs, the local curvatures between the EPs are saved, which prevents forged signals taking advantages from the warping process.

## 2.2 PROPOSED SYSTEM

Proposed system consists of pre-processing the dataset to remove duplicate tuples and make the dataset consistent. Pre-processed dataset is transformed into a data-frame consisting the data of all the signatures and outputs of the respective signature. Dynamic features are extracted and composed into a feature vector along with a few relevant static features. Algorithms mentioned in the below sub-section are used to train and test the dataset. Ultimately these algorithms are analysed based on their accuracies and other evaluation metrics.

### 2.2.1 ALGORITHMS

**1.LOGISTIC REGRESSION (LINEAR CLASSIFIER)**

* Logistic Regression is a machine learning algorithm which is used to classify data.
* Logistic regression uses Sigmoid function to build the model.
* A hypothesis function is built for the input data which is given as an input to the sigmoid function.
* Based on the value obtained from the sigmoid function, the class of the signature is predicted as in Figure 2.2.

Where,

J(θ) is the cost function

h(θ) is the hypothesis of the given problem

x, y is the feature vector and output variable respectively

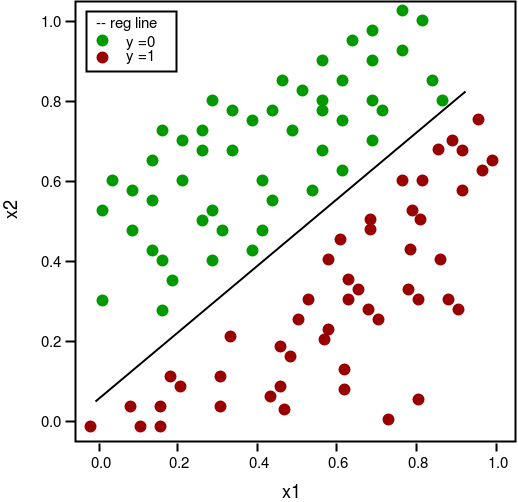
****

Figure 2.2 Logistic Regression Example

**­­­­­­­­­­**

* A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification task.
* The crux of the classifier is based on the Bayes theorem.
* Bayes theorem finds the probability of an event occurring given the probability of another event that has already occurred.

Where,

P(c│x) is Posterior Probability,

P(x│c) is Likelihood,

P(c) is class Prior Probability, and

P(x) is Predictor Prior Probability

**3.K-NEAREST NEIGHBOUR (KNN)**

* KNN is simple, supervised machine learning algorithm that can be used to solve both classification and regression problem
* It works based on minimum distance from the query instance to the training samples to determine the K-Nearest Neighbours
* After we gather the K-Nearest Neighbours, we take simple majority of

these K-Nearest Neighbours to be the prediction of the query instance.

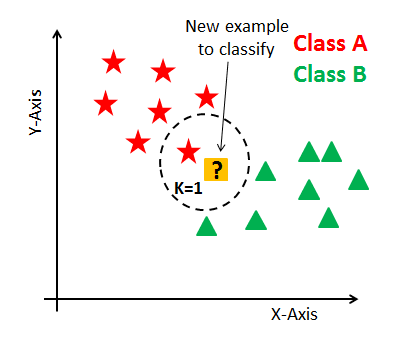


Figure 2.3 K-Nearest Neighbours Example

**4.SUPPORT VECTOR MACHINE (SVM)**

* SVM is a supervised machine learning algorithm which can be used for classification and regression.
* It builds a model by constructing a hyperplane to separate the dataset based on their class.
* The hyperplane is built in such a way that it is at maximum distance from the margins of the dataset.

A close up of a map

Description automatically generated

Figure 2.4 Support Vector Machine Example

**5.DECISION TREES**

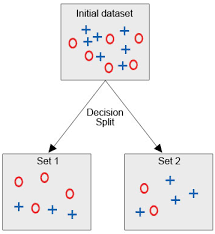
* A Decision Tree is a simple representation for classifying examples.
* It is a Supervised Machine Learning where the data is continuously split according to a certain parameter.
* ****Decision trees are built using a heuristic called **recursive partitioning**.

Figure 2.5 Decision Tree Example

**6.NEURAL NETWORKS**

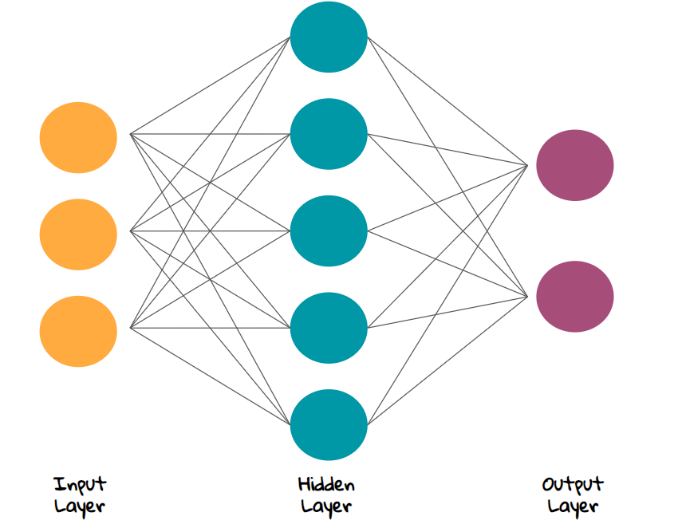
* Artificial neural networks are computing systems that are used for binary and multi-class classification.
* A hypothesis function is built with the help of weights assigned to the edges and the input.
* ****Activation function is used to predict the class of the signature.

Figure 2.6 Neural Network Example

## 2.3 APPLICATIONS

* Detect genuine and forged signatures.
* Applicable in various fields like banking, online-shopping etc
* Comparison and suggestion of an optimal algorithm which can be used to enhance the efficiency of existing approaches.

1. SOFTWARE REQUIREMENTS SPECIFICATION

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) – is a complete description of the behaviour of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Non-functional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints)

**MACHINE LEARNING**

Machine learning is the [scientific study](https://en.wikipedia.org/wiki/Branches_of_science) of [algorithms](https://en.wikipedia.org/wiki/Algorithm) and [statistical models](https://en.wikipedia.org/wiki/Statistical_model) that [computer systems](https://en.wikipedia.org/wiki/Computer_systems) use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence). Machine learning algorithms build a mathematical model of sample data, known as "[training data](https://en.wikipedia.org/wiki/Training_data)", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as [email filtering](https://en.wikipedia.org/wiki/Email_filtering), and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), where it is infeasible to develop an algorithm of specific instructions for performing the task. Machine learning is closely related to [computational statistics](https://en.wikipedia.org/wiki/Computational_statistics), which focuses on making predictions using computers. The study of [mathematical optimization](https://en.wikipedia.org/wiki/Mathematical_optimization) delivers methods, theory and application domains to the field of machine learning.

[Classification](https://en.wikipedia.org/wiki/Statistical_classification) algorithms and [regression](https://en.wikipedia.org/wiki/Regression_analysis) algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a [limited set](https://en.wikipedia.org/wiki/Discrete_number) of values. In [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning), the algorithm builds a mathematical model from a set of data which contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in [feature learning](https://en.wikipedia.org/wiki/Feature_learning). [Dimensionality reduction](https://en.wikipedia.org/wiki/Dimensionality_reduction) is the process of reducing the number of "features", or inputs, in a set of data. Machine learning algorithms can be used to find the unobservable [probability density function](https://en.wikipedia.org/wiki/Probability_density_function) in [density estimation](https://en.wikipedia.org/wiki/Density_estimation) problems. [Meta learning](https://en.wikipedia.org/wiki/Meta_learning_(computer_science)) algorithms learn their own [inductive bias](https://en.wikipedia.org/wiki/Inductive_bias) based on previous experience. In [developmental robotics](https://en.wikipedia.org/wiki/Developmental_robotics), [robot learning](https://en.wikipedia.org/wiki/Robot_learning) algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

**TYPES OF LEARNING ALGORITHMS**

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

**SUPERVISED AND SEMI-SUPERVISED**

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as [training data](https://en.wikipedia.org/wiki/Training_data), and consists of a set of training examples. Each training example has one or more inputs and a desired output, also known as a supervisory signal. In the case of semi-supervised learning algorithms, some of the training examples are missing the desired output. In the mathematical model, each training example is represented by an [array](https://en.wikipedia.org/wiki/Array_data_structure) or vector, and the training data by a [matrix](https://en.wikipedia.org/wiki/Matrix_(mathematics)). Through iterative optimization of an [objective function](https://en.wikipedia.org/wiki/Loss_function), supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

Supervised learning algorithms include [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression](https://en.wikipedia.org/wiki/Regression_analysis). Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. [Similarity learning](https://en.wikipedia.org/wiki/Similarity_learning) is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in [ranking](https://en.wikipedia.org/wiki/Ranking), [recommendation systems](https://en.wikipedia.org/wiki/Recommendation_systems), visual identity tracking, face verification, and speaker verification.

**UNSUPERVISED LEARNING**

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms therefore learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of [density estimation](https://en.wikipedia.org/wiki/Density_estimation) in [statistics](https://en.wikipedia.org/wiki/Statistics), though unsupervised learning encompasses other domains involving summarizing and explaining data features.

**JUPYTER NOTEBOOK**

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical, simulation, statistical modelling, data visualization, machine learning, and much more. Jupyter notebooks next generation notebook interface is called as JupyterLab. JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones.

**PYTHON**

Python was conceived in the late 1980s, and its implementation began in December 1989by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing with the Amoeba operating system. Van Rossum remains Python's principal author. His continuing central role in Python's development is reflected in the title given to him by the Python community: Benevolent Dictator for Life(BDFL).

**Python Features**

Python's features include −

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly definedsyntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allowsinteractive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has thesame interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. Thesemodules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created andported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs thanshell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

## 3.1 SOFTWARE REQUIREMENTS

Operating System : Windows 10

Technology : Python

IDE : Jupyter

Libraries:

Scikit-Learn : Library for Machine Learning algorithms

Pandas : Load and storing of file and manage data

NumPy : Library to handle arrays

Deployment : AWS / Kubeflow / GCP

## 3.2 HARDWARE REQUIREMENTS

Hardware : Pentium

Speed : 1.1 GHz

RAM : 2GB

Hard Disk : 40 GB

# REFERENCES

1. Shafiei, M.M. and Rabiee, H.R., 2003, August. A new online signature verification algorithm using variable length segmentation and hidden Markov models. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* (pp. 443-446). IEEE.
2. Ferrer, M.A., Alonso, J.B. and Travieso, C.M., 2005. Offline geometric parameters for automatic signature verification using fixed-point arithmetic. *IEEE transactions on pattern analysis and machine intelligence*, *27*(6), pp.993-997.
3. Feng, H. and Wah, C.C., 2003. Online signature verification using a new extreme point warping technique. *Pattern Recognition Letters*, *24*(16), pp.2943-2951.
4. Manjunatha, K.S., Manjunath, S., Guru, D.S. and Somashekara, M.T., 2016. Online signature verification based on writer dependent features and classifiers. *Pattern Recognition Letters*, *80*, pp.129-136.
5. Taherzadeh, G., Karimi, R., Ghobadi, A. and Beh, H.M., 2011, February. Evaluation of online signature verification features. In *13th International Conference on Advanced Communication Technology (ICACT2011)* (pp. 772-777). IEEE.
6. Bhatia, S., Bhatia, P., Nagpal, D. and Nayak, S., 2013. Online Signature Forgery Prevention. *International Journal of Computer Applications*, *75*(13).
7. Saffar, M.H., Fayyaz, M., Sabokrou, M. and Fathy, M., 2018. Online signature verification using deep representation: a new descriptor. *arXiv preprint arXiv:1806.09986*.
8. Feng, H. and Wah, C.C., 2003. Online signature verification using a new extreme point warping technique. *Pattern Recognition Letters*, *24*(16), pp.2943-2951.