A Project report on

PREDICTION OF SMART PHONE ADDICTION AMONG STUDENTS USING GRADIENT BOOSTING ALGORITHM

Submitted partial fulfillment of the

requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

(Artificial Intelligence and Machine Learning)

by

VASUNDHARA P **(214G1A33B6)**
SRUTHI B **(214G1A33A5)**
SUBAHAN M **(224G5A3312)**
SANTHOSH KUMAR K **(214G1A3390)**

Under the Guidance of

Mr.K. Kondanna, M.Tech.(Ph.D).
Assistant Professor

Department of Computer Science & Engineering
(Artificial Intelligence and Machine Learning)
SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY
(Autonomous)
Rotarypuram Village, B K Samudram Mandal, Ananthapuramu - 515701

2024-2025

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY
(Autonomous)

(Affiliated to JNTUA, accredited by NAAC with 'A' Grade Approved by AICTE, New Delhi, & Accredited by NBA (CSE, EEE&ECE))

Rotarypuram Village, B K Samudram Mandal, Ananthapuramu – 515701

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

Certificate

This is to certify that the project report entitled Prediction of Smart Phone Addiction among Students using Gradient Boosting Algorithm is the Bonafide work carried out by Vasundhara P bearing Roll Number 214G1A33B6, Sruthi B bearing Roll Number 214G1A33A5 and Subahan M bearing Roll Number 224G5A3312 and Santhosh Kumar K bearing Roll Number 214G1A3390 in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in the Department of Computer Science and Engineering(AI&ML) during the academic year 2024-2025.

Project Guide
Mr.K.Kondanna,Mtech(Ph.D)
Assistant Professor

Head of the Department
Dr.P. Chitralingappa ,Mtech,Ph.D
Associate Professor & HOD

Date:
Place: Anantapur

EXTERNAL EXAMINER

DECLARATION

We Ms. P. Vasundhara bearing reg no: 214GlA33B6, Ms. B. Sruthi bearing reg no: 214G1A33A5, Mr. M. Subahan bearing reg no: 224G5A3312, Mr. K. Santhosh Kumar bearing reg no: 214G1A3390 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram ,hereby declare that the dissertation entitled "PREDICTION OF SMART PHONE ADDICTION AMONG STUDENTS USING GRADIENT BOOSTING ALGORITHM" embodies the report of our project work carried out by us during IV-year Bachelor of Technology under the guidance of Mr. K. Kondanna Assistant Professor, Department of CSD and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

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

P. VASUNDHARA	Reg no: 214G1A33B6
B. SRUTHI	Reg no: 214G1A33A5
M.SUBAHAN	Reg no: 224G5A3312
K. SANTHOSH KUMAR	Reg no: 214G1A3390

VISION & MISSION of the SRIT

VISION

To evolve as a leading department by offering the best comprehensive teaching and learning practices for students to be self-competent technocrats with professional ethics and social responsibilities.

MISSION

1.Continuous enhancement of the teaching-learning practices to gain profound knowledge in theoretical & practical aspects of computer science applications.

2.Administer training on emerging technologies and motivate the students to inculcate self-learning abilities, ethical values and social consciousness to become competent professionals.

3.Perpetual elevation of Industry-Institute interactions to facilitate the students to work on real-time problems to serve the needs of the society.

VISION & MISSION of the Department of CSE (Al & ML)

VISION

To evolve as a leading department by offering the best comprehensive teaching and learning practices for students to be self-competent technocrats with professional ethics and social responsibilities.

MISSION

DM 1: Continuous enhancement of the teaching-learning practices to gam profound knowledge in theoretical & practical aspects of computer science applications.

DM 2: Administer training in emerging technologies and motivate the students to inculcate self-learning abilities, ethical values, and social consciousness to become competent professionals.

DM 3: Perpetual elevation of Industry-Institute interactions to facilitate the students working on real-time problems to serve the needs of society

ACKNOWLEDGMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express our gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our Guide **Mr. K. Kondanna, Assistant Professor, Computer Science & Engineering (Data Science),** who has guided us a lot and encouraged us in every step of the project work. We thank his for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep gratitude to **Mr. A. Kiran Kumar, Assistant Professor, CSE (Al & ML),** and **Mrs. S. Sunitha, Assistant Professor, CSE** project coordinators, for their valuable guidance and unstinting encouragement which enabled us to accomplish our project successfully in time.

We are very much thankful to **Dr. P. Chitraligappa, Associate Professor & Head of the Department, Computer Science** & **Engineering (Al&ML) for** his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr. G. Bala Krishna, Principal** of **Srinivasa Ramanujan Institute of Technology (Autonomous)** for giving the required information in doing our project work. Not forgetting, we thank all other faculty and non- teaching staff, and our friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities.

Finally, we wish to convey our gratitude to our families who have achieved all the requirements and facilities that we need.

Project Associates

214G1A33B6

214G1A33A5

224G5A3312

214G1A3390

ABSTRACT

The increasing prevalence of electronic gadgets in daily life has brought significant changes to the lifestyle and habits of students. While these gadgets offer undeniable benefits for education, communication, and entertainment, they also pose risks of addiction, negatively impacting student's academic performance, mental health, and social interactions. This project aims to utilize machine learning algorithms to predict the levels of Smart phone addiction among individuals. Smart Phone addiction is a growing concern in modern society, with adverse effects on mental health and productivity. This project focuses on using various predictive models to classify the addiction level into categories such as low, moderate, and high. This model uses Multiple Machine learning algorithms, including Gradient Boosting Algorithm, Random Forest Algorithm are employed to train models on the dataset. The proposed System used to analyze diverse factors such as screen time, sleep patterns, and academic performance. These algorithms enable accurate predictions and personalized recommendations, fostering proactive interventions. The performance of these models is evaluating using key metrics such as accuracy, precision, recall, and F-score. The results are visualized through confusion matrices and classification reports.

Keywords: Cat Boost, Random Forest, Smart Phone Addiction, Student Behavior Analysis, Classification Models

CONTENTS **PAGE NO**

ACRONYM	DESCRIPTION
CatBoost	Categorical Boosting
RAM	Random Access Memory
GPU	Graphic Processing unit
CPU	Central Processing Unit
NFR	Non-Functional Requirement
IQR	Interquartile Range
UML	Unified Modeling Language
DFD	Data Flow Diagram
MVT	Model-View-Template
CBA	Cost Benefit Analysis
GPS	Global positioning System
CSV	Comma Separated Value
ROI	Return On Investment
ROM	Read Only Memory

CHAPTER - 1

INTRODUCTION

Mobile addiction has become one of the most significant behavioural issues in today's digital era. With the proliferation of smartphones and mobile applications, people are spending an increasing amount of time on their devices. This addiction not only disrupts daily routines but also poses severe threats to mental health, such as anxiety, depression, and sleep disorders. According to recent studies, mobile addiction is now a global concern, particularly among young adults and teenagers, who use their phones excessively for social media, gaming, and entertainment. The effects of this addiction often extend to various aspects of an individual's life, including academic performance, workplace productivity, and personal relationships.

As the problem of mobile addiction intensifies, there is a pressing need to find effective ways to understand and manage this issue. One promising approach is the use of predictive modelling techniques to assess mobile addiction levels. By leveraging machine learning (ML) algorithms, we can analyse behavioural patterns and predict addiction levels based on a variety of factors such as usage frequency, time spent on applications, and user demographics. Predictive models can provide valuable insights that help identify at-risk individuals and enable timely interventions to prevent further consequences.

The primary objective of this project is to develop an ML-based predictive model that classifies mobile addiction levels into three categories: low, medium, and high. The project leverages a dataset containing various attributes related to mobile usage, such as the duration of app usage, the number of notifications, and the type of activities performed. This dataset forms the foundation for training and testing the models, enabling us to identify patterns in the data that contribute to mobile addiction. By employing machine learning techniques, this project aims to create an automated system capable of predicting addiction levels based on the given input features. Predictive models can provide valuable insights that help identify at-risk individuals and enable timely interventions to prevent further consequences.

The methodology for this project involves several stages, beginning with data preprocessing, where the dataset is cleaned and transformed to ensure its suitability for machine learning models. In the preprocessing phase, categorical variables such as gender and addiction levels are encoded, and features are scaled to ensure they are on the same numerical scale. After preprocessing, the dataset is split into training and testing sets, with the training set being used to train the machine learning models and the testing set being used to evaluate their performance. The data-driven approach will ensure that the predictive models are robust and reliable.

The project utilizes a range of machine learning algorithms, including Gradient Boosting Algorithm, Random Forest and Decision Tree Algorithms. These models are selected for their effectiveness in classification tasks and their ability to handle complex data relationships. Each model will be trained using the training dataset, and the accuracy of the models will be evaluated using key performance metrics such as precision, recall, F-score, and accuracy. The performance of the models will be compared, and the best-performing model will be selected for further deployment.

Furthermore, the results of the models will be visualized using various techniques such as confusion matrices and classification reports. These visualizations will help to better understand the strengths and weaknesses of each model, providing insights into how the algorithms classify the data. By using visual representation techniques, the project aims to make the findings more accessible and interpretable for users, thereby making it easier for stakeholders to act on the results.

So, the development of a machine learning-based predictive model for mobile addiction presents an innovative approach to tackling a widespread issue in the modern digital age. By combining advanced data analysis techniques with machine learning algorithms, this project offers a method for predicting addiction levels that could be used in a variety of applications, from individual self-assessment to public health initiatives. With the potential to identify at-risk individuals and intervene early, this project contributes to a growing body of research aimed at mitigating the adverse effects of mobile addiction.

1.1 Problem Statement

The widespread use of smartphones among students has significantly impacted their daily lives, offering both advantages in education and communication as well as potential risks such as addiction. Smartphone addiction has raised concerns due to its negative effects on academic performance, mental health, and social interactions. This project seeks to address this issue by utilizing machine learning techniques to predict the level of smartphone addiction, categorizing it into low, moderate, or high levels. The study employs algorithms like Gradient Boosting and Random Forest to analyse factors such as screen time, sleep patterns, and academic performance. The goal is to create a predictive model that not only identifies addiction levels accurately but also provides personalized recommendations for intervention. Model performance is evaluated using metrics like accuracy, precision, recall, and F-score, with results visualized through confusion matrices to ensure the system's reliability in predicting smartphone addiction.

1.2 Objectives

To develop a machine learning model that predicts smartphone addiction levels among students, classifying them into low, moderate, or high categories. By analysing factors such as screen time, sleep patterns, and academic performance, the project seeks to provide personalized recommendations to manage addiction. The objective is to evaluate the model's effectiveness using key performance metrics like accuracy, precision, recall, and F-score, ensuring reliable predictions. Ultimately, the project aims to offer an early detection tool to mitigate the negative effects of smartphone addiction on students.

.

CHAPTER – 2

LITERATURE SURVEY

Choi, Y., & Park, H. (2019). "Mobile addiction and mental health problems among adolescents." Journal of Child and Adolescent Psychiatric Nursing,32(4),181-188.This study examines the relationship between mobile addiction and mental health problems in adolescents. Choi and Park focus on the emotional and psychological challenges posed by excessive mobile phone usage, particularly among younger populations. The research highlights a growing concern about mobile addiction's impact on adolescent well-being, leading to issues such as anxiety, depression, and social isolation. The authors propose that early interventions and mindful mobile phone usage may reduce mental health risks. They also stress the importance of parent and educator involvement in managing screen time. This study provides valuable insights into how mobile addiction affects the emotional and psychological state of adolescents, emphasizing the need for preventive measures in schools and communities.

Chen, L., & Zhao, S. (2018). "Behavioural prediction model for mobile app addiction using machine learning." Computers in Human Behaviour, 81, 183-190.Chen and Zhao explore the development of a behavioural prediction model for mobile app addiction using machine learning techniques. Their study presents a framework that uses user behaviour data, such as app usage patterns, interaction frequency, and session length, to predict potential addiction risks. By applying machine learning algorithms, they aim to identify early signs of addiction, offering a more proactive approach to managing mobile app dependency. The study demonstrates the effectiveness of combining behavioural data with machine learning to assess the risk of addiction, which could pave the way for personalized interventions. This research offers a technological approach to tackle mobile app addiction by leveraging predictive modelling and behavioural analysis.

Przybylski, A. K., & Weinstein, N. (2017). "Can you connect with me now? How the presence of mobile communication technology influences face-to-face conversation quality." Journal of Social and Personal Relationships, 34(4), 626-643**.**

Przybylski and Weinstein investigate the impact of mobile communication technology on face-to-face conversations. Their study reveals that the mere presence of mobile phones during in-person interactions can reduce the quality of conversations. Participants reported feeling less engaged and connected when a phone was visible, even if not actively used. The research suggests that mobile technology may distract individuals, leading to shallow interactions and less meaningful social connections. This study underscores the growing concern that while mobile communication fosters virtual connections, it can hinder the development of deep, in-person relationships, which are essential for mental and emotional well-being.

Kuss, D. J., & Griffiths, M. D. (2017). "Social networking sites and addiction: Ten lessons learned." International Journal of Environmental Research and Public Health,14(3),311-319.Kuss and Griffiths provide an extensive review of social networking sites (SNS) and their potential for addictive behavior. Through their analysis, they identify ten key lessons learned from research on SNS addiction, including the influence of psychological factors such as social validation, fear of missing out (FoMO), and compulsive behaviors. The authors emphasize the role of SNS in reinforcing addiction-like patterns, which can lead to negative consequences such as impaired social functioning and decreased mental health. The study highlights the need for both users and developers to be aware of these addictive elements and to promote healthier, more balanced usage patterns to mitigate the risks associated with SNS addiction.

Bian, M., & Leung, L. (2015). "The influence of mobile phone use on the mental health of young adults." Computers in Human Behaviour, 48, 126-132. Bian and Leung's research examines the relationship between mobile phone use and the mental health of young adults. The study focuses on the potential mental health issues associated with excessive phone usage, such as anxiety, stress, and sleep disturbances. Through surveys and data collection, the authors explore the psychological effects of mobile phone dependence, particularly in relation to social interactions and academic performance. The study found a significant correlation between high phone use and negative mental health outcomes, suggesting that interventions to reduce excessive phone time could improve well-being. This research

sheds light on the detrimental effects of mobile phone dependency on young adults' mental health.

Vallerand, R. J., et al. (2014). "Self-determination theory and the study of human motivation." Handbook of Motivation Science, 35-57. Vallerand and colleagues provide an in-depth overview of self-determination theory (SDT) and its application to understanding human motivation. SDT emphasizes the role of intrinsic and extrinsic motivation in human behaviour and well-being. The authors discuss how autonomy, competence, and relatedness are essential for fostering motivation, particularly in educational and organizational settings. The theory is used to explain how mobile addiction could stem from the lack of intrinsic motivation or an imbalance in psychological needs. This comprehensive review of SDT is valuable for understanding the psychological mechanisms behind mobile addiction, offering insights into how motivation can influence mobile usage patterns and potential addiction.

González, P. D., et al. (2018). "Mobile phone addiction and psychological well-being: Evidence from a large cross-sectional study in China." Computers in Human Behaviour 81,135-144González and colleagues explore the relationship between mobile phone addiction and psychological well-being through a large cross-sectional study conducted in China. Their findings reveal that excessive mobile phone use is linked to lower levels of psychological well-being, with participants reporting higher levels of anxiety, depression, and loneliness. The study suggests that mobile phone addiction negatively affects mental health, particularly among younger populations. The authors propose that interventions targeting mobile phone addiction, such as mindfulness practices or controlled usage, could enhance psychological well-being and prevent the adverse effects associated with addiction. This study contributes to the growing body of evidence supporting the mental health risks of mobile phone dependency.

CHAPTER- 3
ANALYSIS

3.1 Existing System

The existing systems that address mobile addiction primarily rely on traditional methods such as surveys, self-report questionnaires, and clinical assessments to measure and identify addictive behaviours related to mobile phone use. These methods involve manually gathering data from users, such as their screen time, app usage patterns, and psychological well-being, and then analysing this information through subjective assessments. While effective to some extent, these old methods often fail to provide real-time insights or predictive capabilities. They also rely heavily on user honesty and awareness, which can lead to bias and inaccurate data collection. Additionally, traditional approaches do not consider the dynamic and constantly changing nature of mobile usage, failing to capture subtle addiction behaviors. As a result, these older methods are less efficient in providing personalized interventions and predicting potential addiction, making them limited in their ability to proactively address the growing issue of mobile addiction.

DISADVANTAGES

1. **Subjectivity and Bias**: Traditional methods like surveys and self-report questionnaires depend on users' honesty and self-awareness, which can lead to biased or inaccurate data collection. People may underreport their usage or fail to recognize their addiction, resulting in unreliable results.
2. **Lack of Real-Time Insights**: These methods do not provide real-time data on mobile usage or addiction behaviours. Users are often assessed based on retrospective data, which can miss immediate warning signs of addiction or changes in usage patterns.
3. **Limited Personalization**: Traditional approaches are generally one-size-fits-all, making it difficult to offer personalized interventions. They don't account for the unique context of each user’s behaviour, which reduces the effectiveness of interventions tailored to individual needs.

4. **Time-Consuming and Labor-Intensive**: Manual data collection and analysis can be time-consuming and require significant resources. Clinicians and researchers must rely on human effort to analyze large datasets, slowing down the process of identifying addiction and implementing solutions.

3.2 Proposed System

The proposed system aims to address the limitations of traditional methods for detecting and predicting mobile addiction by leveraging machine learning techniques and real-time data analytics. Instead of relying on subjective self-reports, the system utilizes data collected directly from users' mobile devices, such as app usage patterns, screen time, and interaction frequency. Machine learning algorithms, such as decision trees, Random Forest, and CatBoost classifiers, will be used to analyse this data and identify patterns that indicate addictive behaviours. By training the model on historical data, the system can accurately predict the likelihood of addiction in real-time, providing personalized alerts and recommendations to users. This method not only improves the accuracy of addiction detection but also allows for continuous monitoring, offering timely interventions before the addiction escalates. The system's ability to adapt to individual behaviour patterns ensures a more tailored and proactive approach, overcoming the shortcomings of traditional methods.

ADVANTAGES

1. **Real-Time Monitoring**: The proposed system enables continuous, real-time tracking of mobile usage, providing immediate insights into potential addictive behaviours. This allows for timely interventions before the addiction worsens, unlike traditional methods that rely on retrospective data.
2. **Scalability and Efficiency**: The automated nature of the system makes it scalable to many users, without the need for extensive manual effort. It can process large datasets quickly, making it more efficient than traditional methods that require labor-intensive analysis.
3. **Objective and Data-Driven**: By using objective data from user devices, such as screen time & app usage patterns, the system eliminates the biases inherent in self-reports, leading to more accurate & reliable assessments of addiction.

3.3 Methodology

3.3.1 CatBoost Algorithm

CatBoost is a gradient boosting algorithm based on decision trees, designed to efficiently handle categorical data. It works by training an ensemble of decision trees sequentially, where each new tree corrects the errors of the previous ones. Unlike traditional boosting algorithms, CatBoost uses ordered boosting, which prevents data leakage by ensuring that each data point is learned without relying on future information. It also employs a unique categorical feature encoding method called ordered target statistics, which replaces categorical values with numerical representations based on conditional probabilities. This allows CatBoost to handle categorical data natively without requiring manual preprocessing like one-hot encoding. Additionally, it supports symmetric tree structures, which speed up training and improve model stability**.**

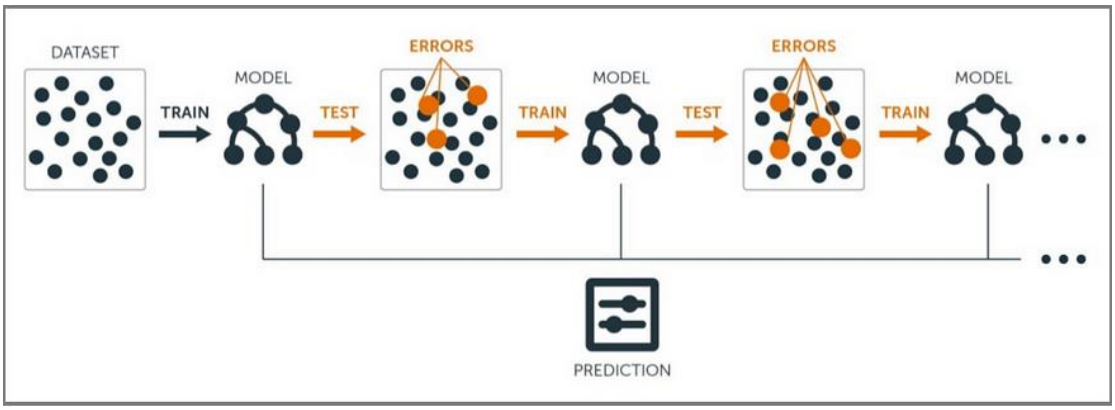


Fig 3.1: Cat Boost Algorithm

Some key features of CatBoost include its ability to handle missing values, efficient GPU and CPU training, and automatic hyperparameter tuning, making it highly scalable for large datasets. It also has built-in regularization techniques to reduce overfitting and improve generalization. CatBoost is optimized for both classification and regression tasks, performing exceptionally well on structured data with categorical features.

3.3.2 Workflow of System

This Model enables continuous, real-time tracking of mobile usage, providing immediate insights into potential addictive behaviors. This allows for timely interventions before the addiction worsens, unlike traditional methods that rely on retrospective data. By using objective data from users' devices, such as screen time and app usage patterns, the system eliminates the biases inherent in self-reports, leading to more accurate and reliable assessments of addiction. The automated nature of the system makes it scalable to many users, without the need for extensive manual effort. It can process large datasets quickly, making it more efficient than traditional methods that require labor-intensive analysis. By detecting addiction early, the system allows proactive interventions, helping users reduce their dependence on mobile devices before it becomes a more severe issue, ultimately promoting healthier mobile usage habits. By training the model on historical data, the system can accurately predict the likelihood of addiction in real-time, providing personalized alerts and recommendations to users. This method not only improves the accuracy of addiction detection but also allows for continuous monitoring, offering timely interventions before the addiction escalates. The system's ability to adapt to individual behavior patterns ensures a more tailored and proactive approach, overcoming the shortcomings of traditional methods

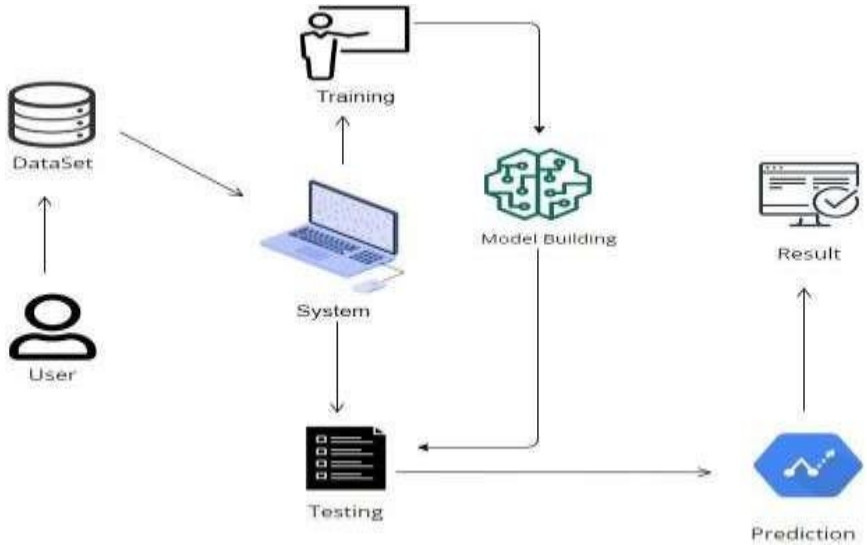


Fig 3.2: System Workflow

CHAPTER - 4

SYSTEM REQUIREMENTS SPECIFICATIONS

4.1 Hardware Requirements

The hardware requirements include the specification of the physical computer resources for a system to work efficiently. The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system.

The Hardware Requirements are listed below:

4.1.1 Processor

A processor, as depicted in fig 4.1, is an integrated electronic circuit that performs the calculations that run a computer. A processor performs arithmetical, logical, input/output (I/0) and other basic instructions that are passed from an operating system (OS). Most other processes are dependent on the operations of a processor. A minimum 1 GHz processor should be used, although we would recommend S2GHz or more. A processor includes an arithmetical logic and control unit (CU), which measures capability in terms of the following:

- Ability to process instructions at a given time
- Maximum number of bits/instructions
- Relative clock speed

Fig 4.1: Processor

4.1.2 Hard Drive

A hard drive, as depicted in fig 4.2, is an electro-mechanical data storage device that uses magnetic storage to store and retrieve digital information using one or more rigid rapidly rotating disks, commonly known as platters, coated. paired with magnetic heads, usually arranged on a moving actuator arm, which reads and writes data to the platter surfaces. Data is accessed in a random-access manner, meaning that individual blocks of data can be stored or retrieved in any order and not only sequentially. HDDs are a type of non-volatile storage, retaining stored data even when powered off. 50 GB or higher is recommended for the proposed system.

Fig 4.2: Hard Disk

4.1.3 Memory (RAM)

Random-access memory (RAM), as depicted in fig 4.3, is a form of computer data storage that stores data and machine code currently being used. A random- access memory device allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory. In today's technology, random-access memory takes the form of integrated chips. RAM is normally associated with volatile types of memory (such as DRAM modules), where stored information is lost if power is removed, although non--volatile RAM has also been developed. A minimum of RAM is recommended for the proposed system.

Fig 4.3: RAM

4.2 Software Requirements

The software requirements are a description of the features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from the client's point of view.

4.2.1 Google Colab

Google is quite aggressive m AI research. Over many years, Google developed an AI framework called TensorFlow and a development tool called Colaboratory. Today TensorFlow is open-sourced and since 2017, Google made Colaboratory free for public use. Collaboratory is now known as Google Colab or simply Colab.

Another attractive feature that Google offers to the developers is the use of GPU. Colab supports GPU and it is totally free. The reasons for making it free for the public could be to make its software standard in the academics for teaching machine learning and data science. It may also have a long-term perspective of building a customer base for Google Cloud Apis which are sold on a per-use basis.

Irrespective of the reasons, the introduction of Colab has eased the learning and development of machine learning applications.

Fig 4.4: Google Colab icon

4.2.2 Python

It is an object-oriented, high-level programming language with dynamic semantics integrated primarily for web and app development. It is extremely attractive in the field of Rapid Application Development because it offers dynamic typing and dynamic binding options. Python is relatively simple, so it's easy to learn since it requires a unique syntax that focuses on readability. Developers can read and translate Python code much easier than other languages. In turn, this reduces the cost of program maintenance and development because it allows teams to work collaboratively without significant language and experience barriers

Fig 4.5: Python icon

4.2.3 VS CODE

VS Code (Visual Studio Code) is a free and open-source source code editor developed by Microsoft. It is a popular tool used by developers for coding and debugging applications. VS Code provides features such as syntax highlighting, code completion, debugging tools, and Git integration, among others. It supports many programming languages including Java, Python, C++, and JavaScript.

One of the advantages of usmg VS Code is its lightweight and fast performance. It has many extensions available that can enhance its functionality and make it more suitable for different kinds of development tasks. Additionally, VS Code has a customizable user interface that allows developers to configure the editor to suit their needs**.**

Fig 4.6: Visual studio code icon

4.3 Functional Requirements

A Functional Requirement is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. In software engineering and systems engineering, a Functional Requirement can range from the high-level abstract statement of the sender's necessity to detailed mathematical functional requirement specifications.

- Data Collection
- Data Pre-Processing
- Training and Testing
- Modelling

4.4 Non-Functional Requirements

Non-Functional Requirement (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system.

Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non-functional Requirements allow you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users is > 10000. They specify the criteria that can be used to judge the operation of a system rather than specific behavior. They may relate to emergent system properties such as reliability, response time and store occupancy.

Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as: - Product Requirements, Organizational Requirements, User Requirements, Basic Operational Requirement, etc.

- A requirement for ease of use; a requirement for ease of maintenance.
- The need for manageability
- The Need for Security
- Requirement of availability
- Need for Scalability

4.5 Scope

The scope of this project is to develop a predictive model that classifies smartphone addiction levels among students using Advanced Machine Learning Algorithms which are Gradient Boosting Algorithm and Random Forest Algorithm. With the increasing reliance on smartphones for education, social interactions, and entertainment, excessive usage can lead to negative impacts on mental health, academic performance, and sleep patterns. The model analyzes key behavioral factors such as screen time, sleep disturbances, and study habits to predict addiction severity, categorizing it into low, moderate, and high levels. By leveraging machine learning, the system provides data-driven insights to understand the patterns of smartphone addiction among students.

4.6 Performance

The performance of the smartphone addiction prediction model is assessed using accuracy, precision, recall, and F-score, ensuring a comprehensive evaluation of classification effectiveness. Confusion matrices help visualize correct and incorrect predictions across different addiction levels (low, moderate, and high), while classification reports provide detailed insights into model reliability. To enhance predictive accuracy, hyperparameter tuning, feature selection, and ensemble methods are employed as optimization techniques. The system's ability to provide personalized recommendations based on predictions allows for targeted interventions to help students manage their smartphone usage. The results of this study can be used by educators, psychologists, and policymakers to design strategies that promote healthy digital habits while minimizing the negative impacts of smartphone overuse on mental health and academic performance.

CHAPTER -5

DESIGN

5.1 System Architecture

An architectural block diagram offers a high-level view of a system's structure, showcasing the main components and their interactions. It represents how major modules, such as data sources, processing units, and evaluation components, are organized and how they communicate with each other to accomplish the system's objectives. This diagram helps in understanding the overall design and flow of the system.

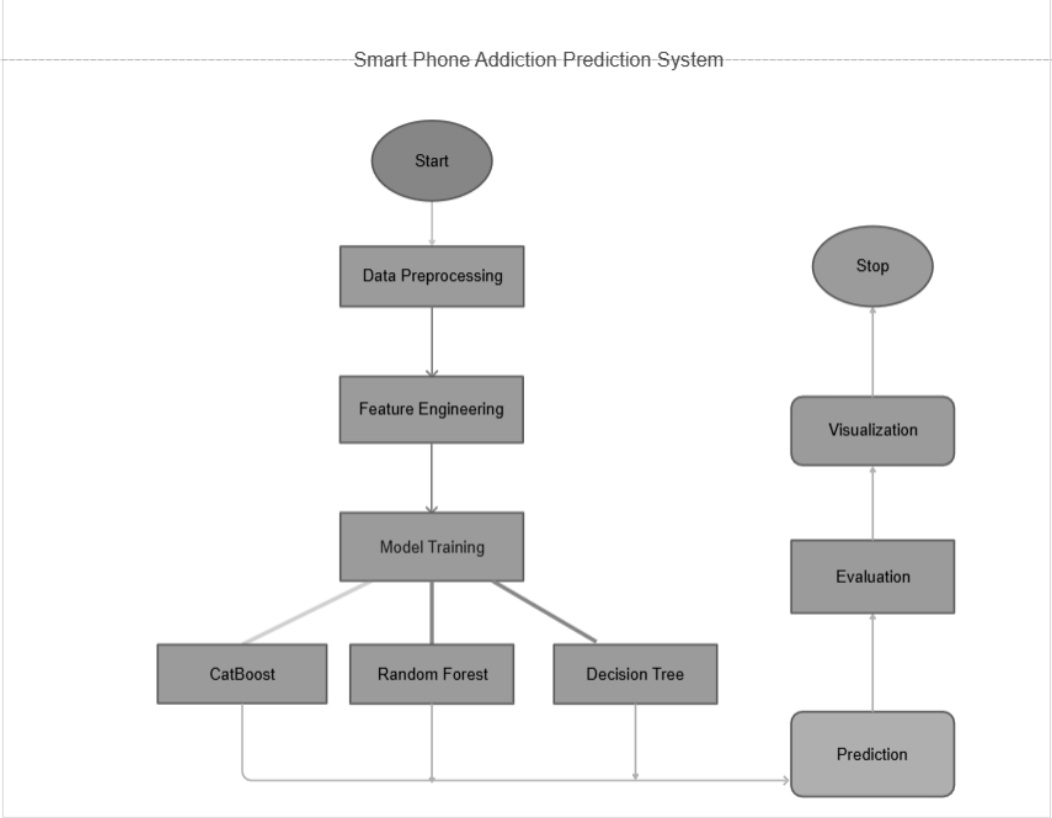


Fig 5.1 System Architecture

The Smartphone Addiction Prediction System follows a structured machine learning pipeline to classify addiction levels. The process begins with data preprocessing, where raw data is cleaned and prepared. Next, feature engineering extracts meaningful attributes such as screen time, sleep patterns, and academic performance. The model training phase utilizes multiple machines learning algorithms, including CatBoost, Random Forest, and Decision Tree, to build predictive models.

Once trained, the system generates predictions on addiction levels. The model's performance is assessed through evaluation metrics like accuracy and precision, followed by visualization of results for better interpretation. The system ultimately provides insights that help in addressing smartphone addiction effectively.

5.2 Steps Involved in Design

- Data Collection
- Data Preprocessing
- Model Training
- Model Evaluation

5.2.1 Data Collection

The data collection for this project involves gathering behavioral, psychological, and academic indicators that contribute to smartphone addiction among students. The data is collected using structured surveys, self-assessment questionnaires, and mobile usage tracking tools to measure factors such as daily screen time, frequency of phone usage, time spent on different ap plications, late-night usage, and social media engagement. Additionally, academic performance, sleep quality, stress levels, anxiety, and attention span are recorded to analyze the impact of excessive smartphone use on mental and academic well-being. The dataset includes students from various age groups, educational backgrounds, and lifestyles to ensure diversity and accuracy in prediction.

5.2.2 Data Preprocessing

Data preprocessing is a crucial step to ensure clean, structured, and high-quality data for training machine learning models. The raw dataset collected from surveys, mobile usage logs, and behavioral assessments may contain incomplete, inconsistent, or noisy data, which needs to be handled effectively. The first step involves handling missing values by using techniques like mean/mode imputation for numerical data and frequent category imputation for categorical variables. Duplicate entries and irrelevant

features are removed to improve model efficiency. Additionally, outliers in features such as screen time and social media Z-score analysis or IQR (Interquartile Range) to prevent skewed predictions. Next, feature encoding is performed to convert categorical variables (e.g., "frequent social media user" as Yes/No) into numerical values using One-Hot Encoding or Label Encoding, ensuring compatibility with machine learning algorithms. Feature scaling and normalization techniques such as Min-Max Scaling or Standardization are applied to ensure that features like screen time and academic performance have comparable ranges.

5.2.3 Model Training

After data preprocessing, the next step is to train the machine learning models—Decision Tree, Random Forest, and CatBoost—to predict smartphone addiction levels. The processed dataset is split into training (80%) and testing (20%) subsets to evaluate model performance effectively. Each model is trained on labelled data where the addiction level (low, moderate, high) is the target variable, while features such as screen time, social media engagement, academic performance, sleep quality, and stress levels act as predictors.

1. Decision Tree Model: A Decision Tree classifier is trained using Gini impurity or entropy as a splitting criterion. It learns patterns from training data and creates a hierarchical structure for classification.

2. Random Forest Model: This ensemble model trains multiple Decision Trees on random subsets of data, combining their outputs to improve accuracy and reduce overfitting.

3. CatBoost Model: A gradient boosting algorithm that efficiently handles categorical data and minimizes overfitting by assigning different weights to misclassified samples.

Each model is trained using cross-validation to fine-tune hyperparameters like tree depth, learning rate, and number of estimators for optimal performance.

5.2.4 Model Evaluation

After training the Decision Tree, Random Forest, and CatBoost models, their performance is evaluated using key classification metrics to determine their accuracy in predicting smartphone addiction levels. The testing dataset (20% of the total data) is used to assess how well each model generalizes to unseen data.

1. **Accuracy:** Measures the overall correctness of predictions by calculating the ratio of correctly predicted instances to the total instances.
2. **Precision & Recall:** Precision indicates how many predicted positive cases (high addiction) are correct, while recall measures how many actual positive cases were identified by the model.
3. **F1-Score:** A balance between precision and recall, ensuring the model performs well in cases where false positives and false negatives need to be minimized.
4. **Confusion Matrix:** Provides a detailed breakdown of correctly and incorrectly classified instances across low, moderate, and high addiction levels.

5.3 UML Introduction

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of the best engineering practices that have proven successful in the modeling of large and complex systems. UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

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

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.

5.3.1 Class Diagram

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes.

The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram could provide certain functionalities. These functionalities provided by the class are termed "methods" of the class.

Apart rom this, each class may have certain "attributes" that uniquely identify the class.

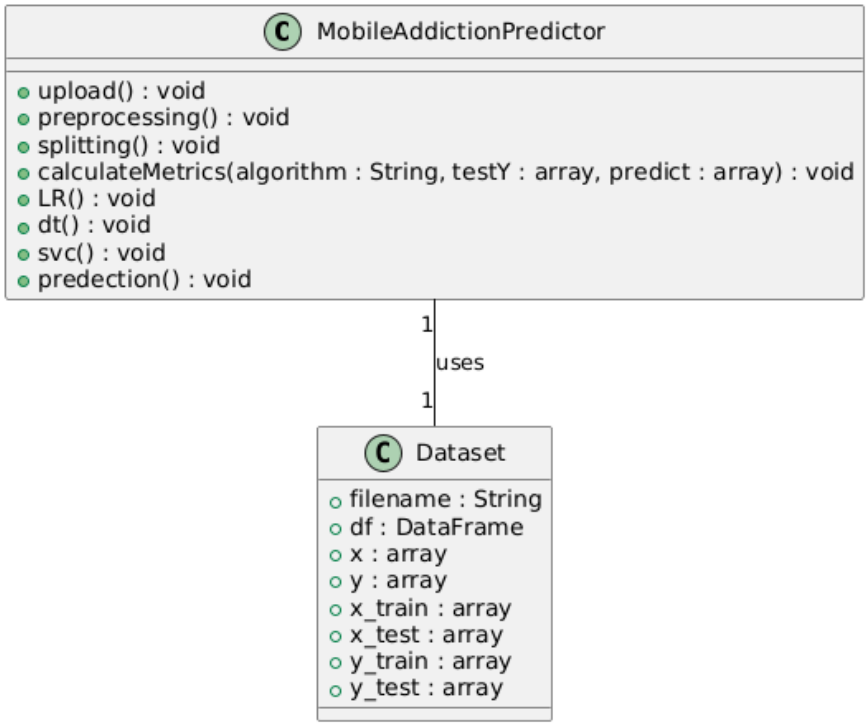


Fig 5.2 Class Diagram

5.3.2 Activity diagram

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

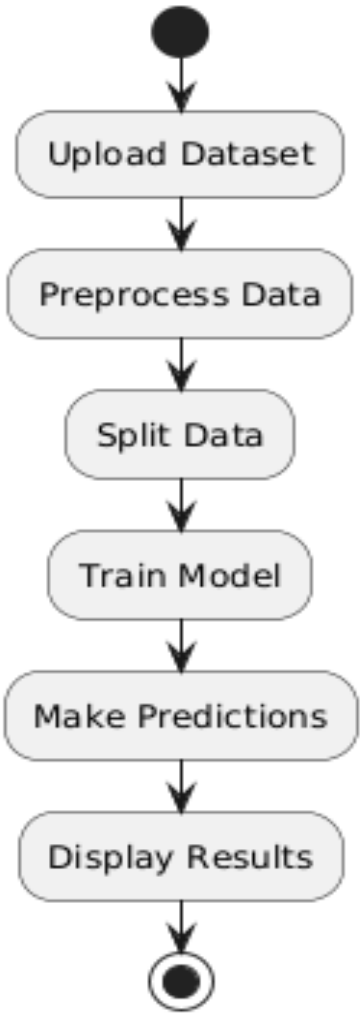


Fig 5.3 Activity Diagram

5.3.3 Data flow diagram

A data flow diagram (DFD) is a graphical representation of how data moves within an information system.

It is a modeling technique used in system analysis and design to illustrate the flow of data between various processes, data stores, data sources, and data destinations within a system or between systems.

Data flow diagrams are often used to depict the structure and behavior of a system, emphasizing the flow of data and the transformations it undergoes as it moves through the system.

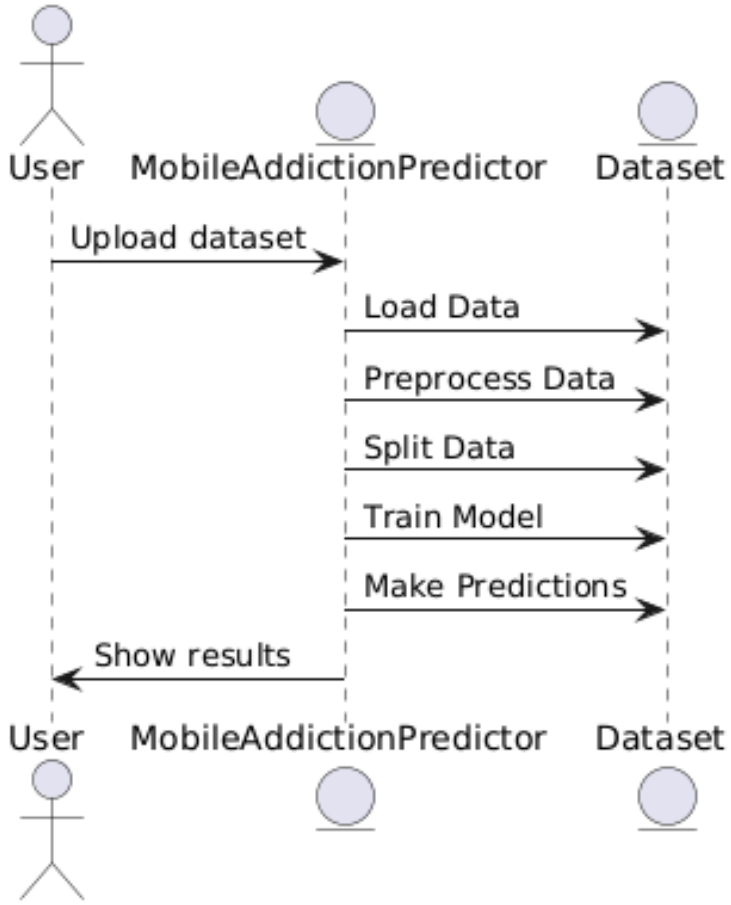


Fig 5.4 Data Flow Diagram

5.3.4 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is the construction of a Message Sequence Chart.A sequence diagram shows, as parallel vertical lines ("lifelines"), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

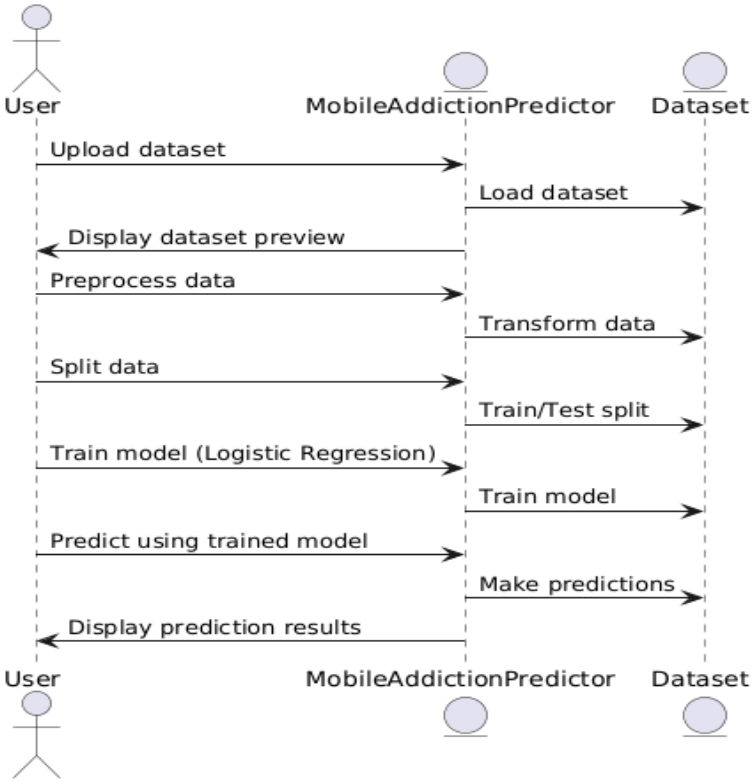


Fig 5.5 Sequence Diagram

These diagrams are commonly used in software engineering to model real-time processes, such as API calls, user authentication, or transaction processing. They help developers and analysts understand the sequence of events and the dependencies between different system elements**.**

5.3.5 Use Case diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.

Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use

cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

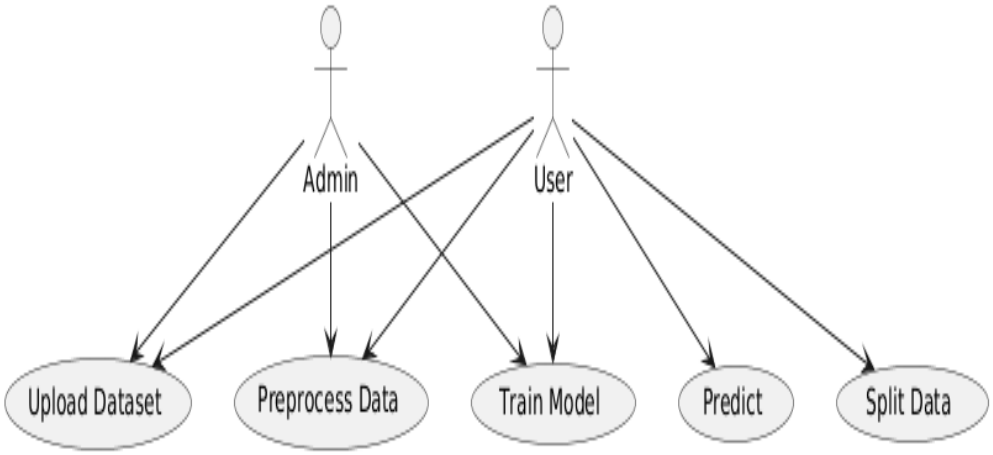


Fig 5.4 Use Case Diagram

Use case diagrams are crucial in defining system scope and user expectations, making them valuable for stakeholders during the initial stages of system development. They are widely used in requirement gathering, providing a clear understanding of system behavior before actual implementation begins**.**

5.3.6 Deployment Diagram:

A deployment diagram in UML illustrates the physical arrangement of hardware and software components in the system. It visualizes how different software artifacts, such as data processing scripts and model training components, are deployed across hardware nodes and interact with each other, providing insight into the system's infrastructure and deployment strategy.

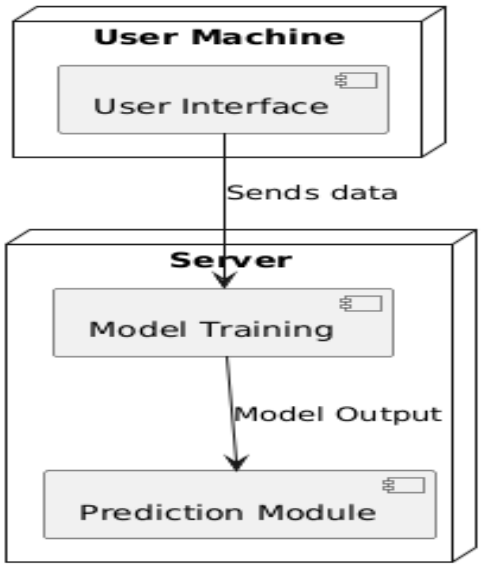


Fig 5.5 Deployment Diagram

CHAPTER -6

IMPLEMENTATION

The implementation of this project involves multiple stages, starting with data collection, preprocessing, model training, and evaluation to predict smartphone addiction levels among students. First, relevant data is gathered from surveys, mobile usage logs, and behavioral assessments, capturing key features such as screen time, social media engagement, sleep patterns, academic performance, and mental health indicators. The collected data undergoes preprocessing, including handling missing values, encoding categorical variables, normalizing numerical data, and removing outliers to improve model efficiency.

6.0 Libraries

Pip: The pip command is a tool for installing and managing Python packages, such as those found in the Python Package Index. It's a replacement for easy installation. The easiest way to install the NFL* python modules and keep them up to date is with a Python-based package manager called pip.

Pip install Module Name

CatBoost Module: CatBoost (Categorical Boosting) is a high-performance gradient boosting library developed by Yandex, specifically designed to handle categorical data efficiently. Unlike traditional boosting algorithms, CatBoost automatically processes categorical features without requiring extensive preprocessing, such as one-hot encoding or label encoding, making it highly efficient for real-world datasets. It is particularly useful for classification, regression, and ranking tasks, offering superior performance with minimal hyperparameter tuning. The library is optimized for speed and accuracy, reducing overfitting by implementing ordered boosting and handling missing values effectively. CatBoost supports GPU acceleration

Pandas: Pandas is a powerful data manipulation and analysis library in Python, widely used for handling structured data. It provides data structures like Data Frame and Series that allow efficient storage, processing, and transformation of datasets. With built-in functions for data cleaning, filtering, merging, and grouping, Pandas are essential for preprocessing data before feeding it into machine learning models.

NumPy: NumPy (Numerical Python) is a fundamental library for numerical computing**,** offering support for multi-dimensional arrays and matrices**,** along with mathematical functions for operations like linear algebra, statistics, and random number generation**.** NumPy's objective provides efficient storage and manipulation of numerical data, making it a core component for scientific computing and machine learning.

Matplotlib: Matplotlib is a comprehensive data visualization library that allows users to create static, animated, and interactive plots. It provides various chart types, including line plots, histograms, bar charts, and scatter plots**,** helping in exploration data analysis (EDA) and result interpretation. It works seamlessly with Pandas and NumPy, allowing easy plotting of datasets.

Scikit-Learn (Sklearn): Scikit-Learn is a widely used machine learning library that provides tools for classification, regression, clustering, and dimensionality reduction**.** It includes preprocessing functions, model selection techniques, and evaluation metrics for building and optimizing machine learning models. With built-in implementations of algorithms like Decision Trees, Random Forest, Support Vector Machines, and Gradient Boosting, it simplifies the development of predictive models.

Django: Django is a high-level Python web framework that enables developers to build secure, scalable, and maintainable web applications efficiently. It follows the Model-View-Template (MVT) architecture, which separates data handling (Model), business logic (View), and presentation (Template), making development structured and organized. With built-in authentication, database management, and security features, Django reduces development time and enforces best practices.

6.1 Implementation

6.1.1 Importing modules and Libraries

Modules and libraries provide pre-written functionalities, making it easier to develop applications efficiently. Importing them allows developers to use existing functions instead of writing code from scratch. The import statement is used to bring external modules into a script.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression

from sklearn.preprocessing import StandardScaler

import warnings
warnings.filterwarnings('ignore')
import joblib
import os
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
from catboost import CatBoostClassifier
import pandas as pd
import joblib
```

Fig 6.1 importing modules and libraries

6.1.2 Upload data and Training the Model

The Upload data function processes a file upload in a Django web application. When a user uploads a CSV file via a POST request, the function saves the file temporarily and loads its content into a Pandas Data Frame. It then applies labels encoding to categorical columns like "Gender" and "Phone Dependency Level" to convert them into numerical format. After preprocessing, a count plot is generated using Seaborn to visualize the distribution of "Phone Dependency Level," with bar annotations for better readability. The dataset is then split into training and testing sets using an 80-20 ratio, ensuring a structured approach for machine learning model training.

Once processing is complete, the temporary file is deleted, and the first 100 rows of the dataset are displayed on the frontend. The dataloaded flag is set to True to indicate that the data has been successfully processed. If no file is uploaded, the function simply renders the upload page.

```
def Upload_data(request):
    load=True
    global df,dataloaded
    global X_train,X_test,y_train,y_test
    if request.method == 'POST' and request.FILES.get('file'):
        uploaded_file = request.FILES['file']
        file_path = default_storage.save(uploaded_file.name, uploaded_file)
        df=pd.read_csv(default_storage.path(file_path))
        df["Gender"]=le.fit_transform(df["Gender"])
        df["Phone Dependency Level"]=le.fit_transform(df["Phone Dependency Level"])
        sns.set(style="darkgrid")  # Set the style of the plot
        plt.figure(figsize=(8, 6))  # Set the figure size
        ax = sns.countplot(x='Phone Dependency Level', data=df)
        plt.title("Count Plot")  # Add a title to the plot
        plt.xlabel("Categories")  # Add label to x-axis
        plt.ylabel("Count")  # Add label to y-axis
        # Annotate each bar with its count value
        for p in ax.patches:
            ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()),
                        ha='center', va='center', fontsize=10, color='black', xytext=(0, 5),
                        textcoords='offset points')
        plt.xticks(rotation=90)
        plt.show()
        x=df.iloc[:,2:19]
        y=df.iloc[:,-1]
        X_train,X_test,y_train,y_test= train_test_split(x,y,test_size=0.20,random_state=42)
        default_storage.delete(file_path)
        outdata=df.head(100)
        dataloaded=True
        return render(request,'train.html',{'temp':outdata.to_html()})
    return render(request,'train.html',{'upload':load})
```

Fig 6.2 upload the data

6.1.3 Model Architecture

The model architecture defines the structure of the machine learning or deep learning model, including the number of layers, types of neurons, activation functions, and connections between layers. In this project, Decision Tree, Random Forest, and CatBoost algorithms are used for predicting smartphone addiction among students. After defining the architecture, the model is compiled to specify the optimizer, loss function, and evaluation metrics. For machine learning models, compilation involves setting up hyperparameters such as the number of estimators, depth of trees, and learning rate. Decision Tree Classifier (DTC): A tree-based model that splits data based on feature importance using Gini impurity.

Random Forest Classifier (RFC): An ensemble model that creates multiple decision trees and aggregates their results for better accuracy. CatBoost Classifier: A gradient boosting model optimized for categorical data, which automatically encodes categorical features.

```
def RFC(request):
    if dataloaded == False:
        return redirect('upload')
    if os.path.exists('model/RF_model.pkl'):
        predict = clf.predict(X_test)
        calculateMetrics("RF_model", predict, y_test)
    else:
        clf = RandomForestClassifier(n_estimators=100, criterion='gini', max_depth=5, random_state=42)
        clf.fit(X_train, y_train)
        joblib.dump(clf, 'model/RF_model.pkl')
        print("Model saved successfully.")
        predict = clf.predict(X_test)
        calculateMetrics("RF_model", predict, y_test)
    return render(request,'train.html',
                  {'algorithm':'Random Forest',
                   'accuracy':accuracy[-1],
                   'precision':precision[-1],
                   'recall':recall[-1],
                   'fscore':fscore[-1]})
def catboost_view(request):
    if dataloaded == False:
        return redirect('upload')
    if os.path.exists('model/CatBoost_model.pkl'):
        clf = joblib.load('model/CatBoost_model.pkl')
        predict = clf.predict(X_test)
        calculateMetrics("CatBoost_model", predict, y_test)
    else:
        clf = CatBoostClassifier(iterations=500, depth=6, learning_rate=0.1,
        loss_function='MultiClass', verbose=0)  # Use MultiClass for multiple categories
        clf.fit(X_train, y_train)
        joblib.dump(clf, 'model/CatBoost_model.pkl')
        predict = clf.predict(X_test)
        calculateMetrics("CatBoost_model", predict, y_test)
    return render(request,'train.html',
                  {'algorithm':'CatBoostClassifier',
                   'accuracy':accuracy[-1],
```

Fig 6.3 Model Architecture

6.2 Model Evaluation

Model evaluation is a critical step in machine learning that determines how well a trained model performs on unseen data. It ensures the model is not only accurate on training data but also generalizes well to new inputs**.**

```
def calculateMetrics(algorithm, testY,predict):
    testY = testY.astype('int')
    predict = predict.astype('int')
    p = precision_score(testY, predict,average='macro') * 100
    r = recall_score(testY, predict,average='macro') * 100
    f = f1_score(testY, predict,average='macro') * 100
    a = accuracy_score(testY,predict)*100
    accuracy.append(a)
    precision.append(p)
    recall.append(r)
    fscore.append(f)
    print(algorithm+' Accuracy    : '+str(a))
    print(algorithm+' Precision   : '+str(p))
    print(algorithm+' Recall      : '+str(r))
    print(algorithm+' FSCORE      : '+str(f))
    report=classification_report(predict, testY,target_names=labels)
    print('\n',algorithm+" classification report\n",report)
    conf_matrix = confusion_matrix(testY, predict)
    plt.figure(figsize =(5, 5))
```

Fig 6.4 model Evaluation

Data Splitting: Before evaluating a model, the dataset is typically divided into different subsets:

- Training Set: Used to train the model by learning patterns from data.
- Validation Set: Helps in tuning hyperparameters and selecting the best model configuration.
- Test Set: Used to measure the model's final performance on unseen data.

A common approach is an 80-20 split, where 80% of data is for training and 20% for testing. Another method is the 70-15-15 split, which includes a validation set. Proper data splitting prevents information leakage and ensures the model is evaluated on truly unseen data.

Cross-Validation

Cross-validation is a robust method to ensure that evaluation results are reliable and not dependent on a single data split. **K-fold cross-validation** is the most common technique, where the dataset is split into *K* subsets (folds). The model is trained *K* times, using a different fold as the test set in each iteration, and the final performance is averaged.

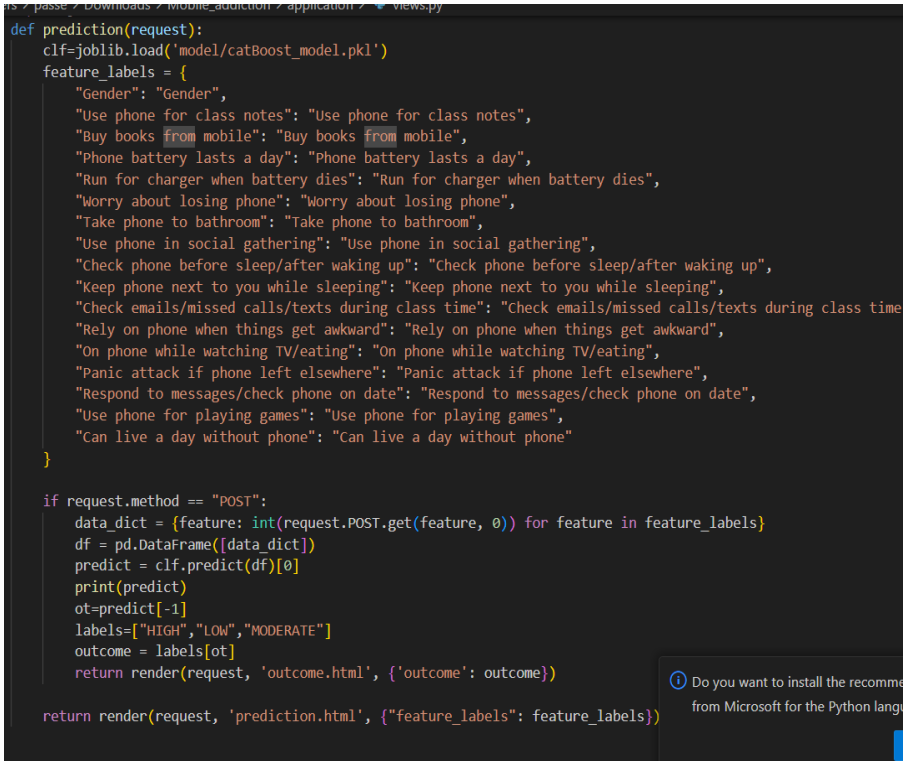
Benefits:

- Reduces bias from a single train-test split.
- Provides a more accurate estimate of model performance.

Classification

Classification is a type of supervised learning in machine learning where the goal is to categorize data into predefined classes or labels. The model learns patterns from labeled training data and predicts the class of new, unseen data. Common classification problems include spam detection (spam or not spam), medical diagnosis (disease or no disease), and sentiment analysis. There are different types of classification algorithms, including logistic regression, decision trees, random forests, support vector machines (SVMs),CatBoost Classifier and deep learning models.

Model Evaluation



```
def prediction(request):
    clf=joblib.load('model/catBoost_model.pkl')
    feature_labels = {
        "Gender": "Gender",
        "Use phone for class notes": "Use phone for class notes",
        "Buy books from mobile": "Buy books from mobile",
        "Phone battery lasts a day": "Phone battery lasts a day",
        "Run for charger when battery dies": "Run for charger when battery dies",
        "Worry about losing phone": "Worry about losing phone",
        "Take phone to bathroom": "Take phone to bathroom",
        "Use phone in social gathering": "Use phone in social gathering",
        "Check phone before sleep/after waking up": "Check phone before sleep/after waking up",
        "Keep phone next to you while sleeping": "Keep phone next to you while sleeping",
        "Check emails/missed calls/texts during class time": "Check emails/missed calls/texts during class time
        "Rely on phone when things get awkward": "Rely on phone when things get awkward",
        "On phone while watching TV/eating": "On phone while watching TV/eating",
        "Panic attack if phone left elsewhere": "Panic attack if phone left elsewhere",
        "Respond to messages/check phone on date": "Respond to messages/check phone on date",
        "Use phone for playing games": "Use phone for playing games",
        "Can live a day without phone": "Can live a day without phone"
    }

    if request.method == "POST":
        data_dict = {feature: int(request.POST.get(feature, 0)) for feature in feature_labels}
        df = pd.DataFrame([data_dict])
        predict = clf.predict(df)[0]
        print(predict)
        ot=predict[-1]
        labels=["HIGH","LOW","MODERATE"]
        outcome = labels[ot]
        return render(request, 'outcome.html', {'outcome': outcome})

    return render(request, 'prediction.html', {"feature_labels": feature_labels})
```

Do you want to install the recomme
from Microsoft for the Python langu

Fig 6.5 **Model Prediction**

Model evaluation is a crucial step in machine learning that determines how well a trained model performs on unseen data. It helps ensure that the model is not just memorizing the training data but is capable of making accurate predictions on new inputs. A properly evaluated model prevents issues like overfitting or underfitting, leading to better real-world performance. The evaluation process involves testing the model on a separate dataset, using performance metrics to assess its accuracy, precision, recall, and overall effectiveness.

6.3 Classification Metrics

Classification metrics are essential for evaluating the performance of a classification model. They help determine how well a model distinguishes between different categories and provide insights into areas where the model may need improvement. These metrics are especially important in applications like medical diagnosis, spam detection, and fraud detection, where misclassifications can have

significant consequences.

1.Accuracy

Accuracy is the most used metric and measures the proportion of correctly classified instances out of the total instances. It is calculated as:

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

where **TP** (True Positives) and **TN** (True Negatives) are correctly classified instances, while **FP** (False Positives) and **FN** (False Negatives) are misclassified instances. Although useful, accuracy can be misleading in imbalanced datasets, where one class dominates the others.

2.Precision

Precision measures how many of the predicted positive instances are actually correct.

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

A high precision score indicates that the model makes fewer false positive errors, making it an important metric when false positives are costly, in email spam detection

3.Recall

Recall measures how many actual positive instances the model correctly identifies. It is given by:

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

A high recall score indicates that the model captures most of the actual positive instances, making it useful in situations where false negatives are critical, such as in disease detection or fraud identification.

4.F1-Score

F1-Score is the harmonic mean of precision and recall, balancing both metrics. It is calculated as:

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

F1-score is particularly useful when dealing with imbalanced datasets because it considers both false positives and false negatives.

5. Confusion Matrix

A Confusion Matrix provides a detailed breakdown of a model's predictions, showing the number of true positives, false positives, true negatives, and false negatives. It helps visualize model performance and calculate other metrics like precision, recall, and F1-score.

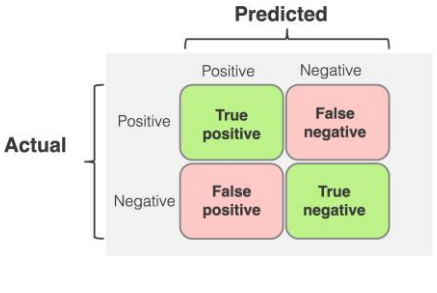


Fig 6.6 Confusion matrix

6.4 Web app UI

The web app UI for your project serves as the central platform for users to interact with document ingestion, embedding creation, and semantic search functionalities. It features an intuitive interface with sections for document ingestion and semantic search The semantic search interface includes a search bar, search results display, document preview, and filtering options for refining search results. Settings and configuration options allow users to customize embedding models and search parameters.

Additionally, the UI includes user feedback and help resources, such as feedback form and documentation links. With user authentication, responsive design, and error handling, the web app UI ensures a seamless and user-friendly experience for users to explore and interact with the project's functionalities.

Fig 6.7 Web Interface

CHAPTER 7

System Study and Testing

7.1 System Study

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

Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

7.1.1 Economic Feasibility

Economic feasibility in a feasibility study of a prediction system evaluates whether the project is financially viable by analyzing costs, expected benefits, and long-term sustainability. It involves a cost-benefit analysis (CBA) to compare the total expenses, including hardware, software, training, and operational costs, against the projected savings or revenue. A positive net benefit ensures that the system is worth the investment. Additionally, Return on Investment (ROI) is calculated to determine the profitability of the system, helping stakeholders decide whether to proceed with the implementation. A higher ROI indicates a more economically feasible solution, making the prediction system an asset for decision-making and resource allocation.

7.1.2 Technical Feasibility

Technical feasibility assesses whether the proposed prediction system can be successfully developed and implemented using available technology, infrastructure, and expertise. It ensures that the system meets performance requirements, integrates with existing platforms, and operates efficiently without excessive technical risks.

A key aspect of technical feasibility is evaluating hardware and software requirements. The system must have sufficient processing power, storage, and network capabilities to handle large datasets, especially for AI-based prediction models. Compatibility with existing databases and integration with other software systems are also critical. Additionally, the choice of prediction algorithms (e.g., Machine Learning, Deep Learning, or Statistical Models) must align with the available computational resources.

7.1.3 Social Feasibility

Social feasibility evaluates the acceptance, impact, and ethical implications of implementing a prediction system within a community or organization. It ensures that the system aligns with user expectations, cultural norms, and societal values while minimizing resistance to adoption. A key factor is assessing how the system will affect different user groups, such as employees, customers, or the general public. If a prediction system significantly alters workflows or decision-making processes, proper training and awareness programs may be needed to encourage user acceptance.

Another important aspect is the ethical and privacy concerns surrounding data collection and usage. Users must trust that their data is being handled securely and transparently, following ethical AI principles and data protection laws. Additionally, social feasibility examines whether the system creates any social inequalities by disproportionately benefiting or disadvantaging certain groups. Ensuring that the prediction model is fair, unbiased, and inclusive helps gain public trust and enhances its long-term success. Ultimately, a socially feasible system is one that is widely accepted, ethically responsible, and beneficial to all stakeholders without causing significant resistance or negative societal impact.

7.2 System testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product.

Software systems meet their specified requirements and functions as expected. Testing involves creating and executing test cases to verify that the software behaves correctly under various conditions and scenarios.

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. The testing process can be manual or automated and can be conducted at various stages of the software development life cycle, such as unit testing, integration testing, system testing, and acceptance testing. Overall, testing plays a crucial role in ensuring the quality and reliability of software products.

7.3 Types of testing

7.3.1Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application it is done after the completion of an individual unit before integration. This is structural testing, that relies on knowledge of its construction and is invasive.

Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contain clearly defined inputs and expected results. unit testing is an essential part of the software development process, which helps developers identify defects early, improve the quality of the software, and reduce the cost of bug fixes.

7.3.2 Integration testing

Integration testing is a type of software testing that involves testing the interactions and interfaces between different software components that have been integrated into a larger system. The purpose of integration testing is to ensure that these components work together as expected and that the integrated system meets its functional requirements.

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is even driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.3.3 Functional Test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified, and the effective value of current tests is determined.

7.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and points.

7.4.1 White Box Testing

White Box Testing is a test in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.3.5 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a test in which the software under test is treated, as a black box you cannot "see" into it. The test provides input and responds to outputs without considering how the software works. Field testing will be performed manually, and functional tests will be written in detail, Features to be tested.

- Verify that there is correct format of input images.
- Ensure that the optimizer and loss function are set up correctly for training.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or one step up- software applications at the company level-interact without error.

Acceptance Testing

User Acceptance Testing 1s a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 8

Results and Analysis

6.1 Evaluation of algorithms:

6.1.1 Catboost Classifier

Evaluation metrics such as Accuracy, RMSE, MAE, and Log Loss show that CatBoost provides stable and reliable predictions, especially for small to medium-sized datasets.

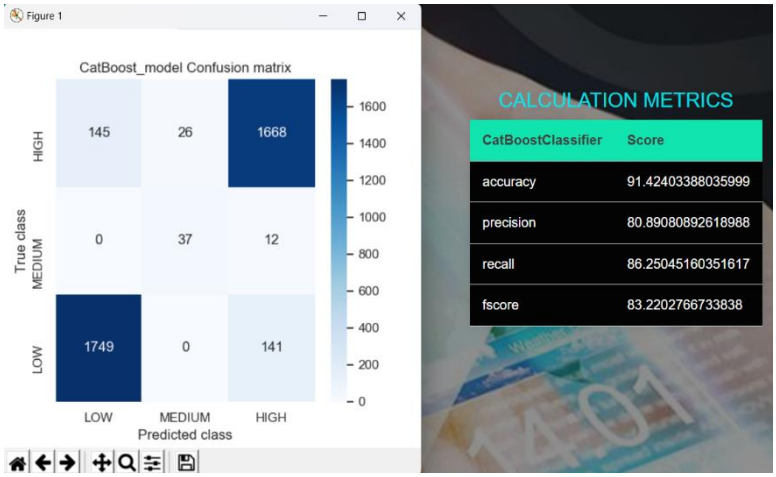


Fig 8.1 catboost confusion matrix

6.1.2 Random Forest classifer

Random Forest requires manual encoding (one-hot or label encoding) for categorical variables, which may lead to information loss, reducing accuracy compared to CatBoost, which handles categorical data natively.

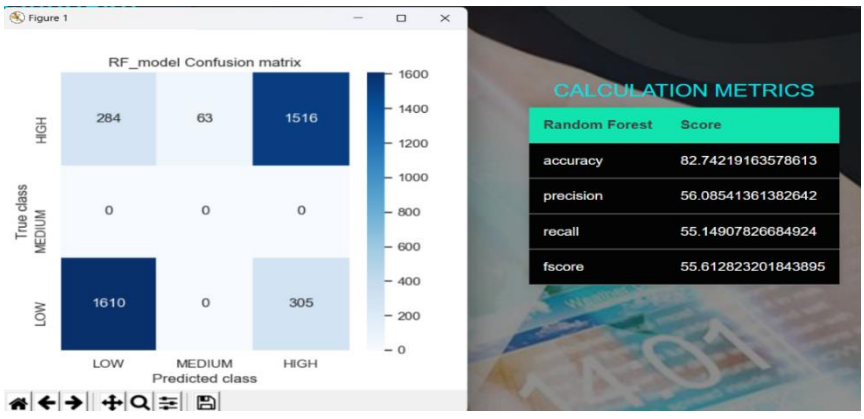


Fig 8.2 Random Forest Confusion Matrix

6.1.3 Decision Tree Classifier

Decision trees tend to overfit on training data, making them less generalizable, A single decision tree has limited predictive power small changes in data can drastically alter the tree structure, leading to unstable predictions.

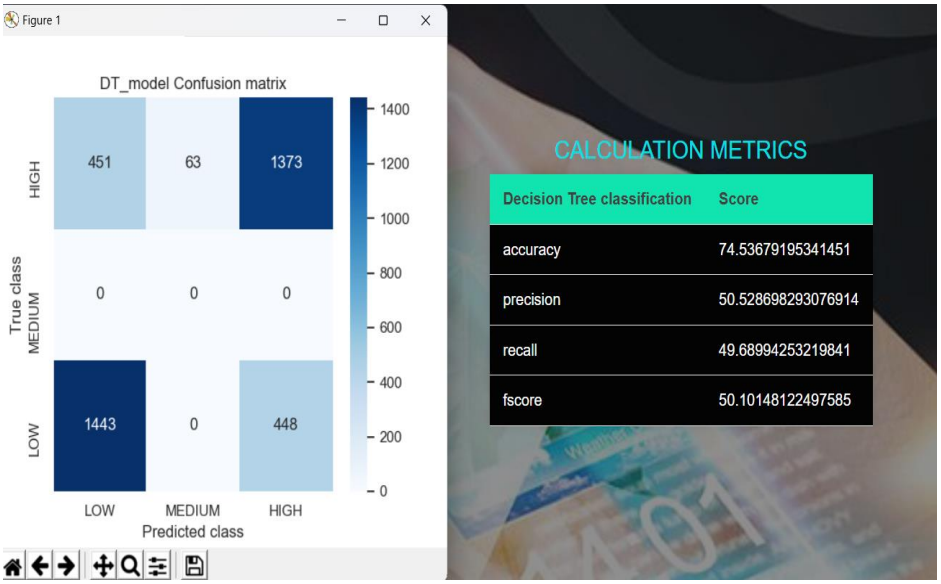


Fig 8.3 Decision Tree confusion matrix

6.2 performance Comparison of Model

Decision Trees are simple, fast, and easy to interpret but suffer from high overfitting and low generalization. They are sensitive to small changes in data, which can lead to drastically different tree structures, making predictions unstable. While Decision Trees work well for small datasets, they struggle with large datasets and categorical features, requiring manual preprocessing.

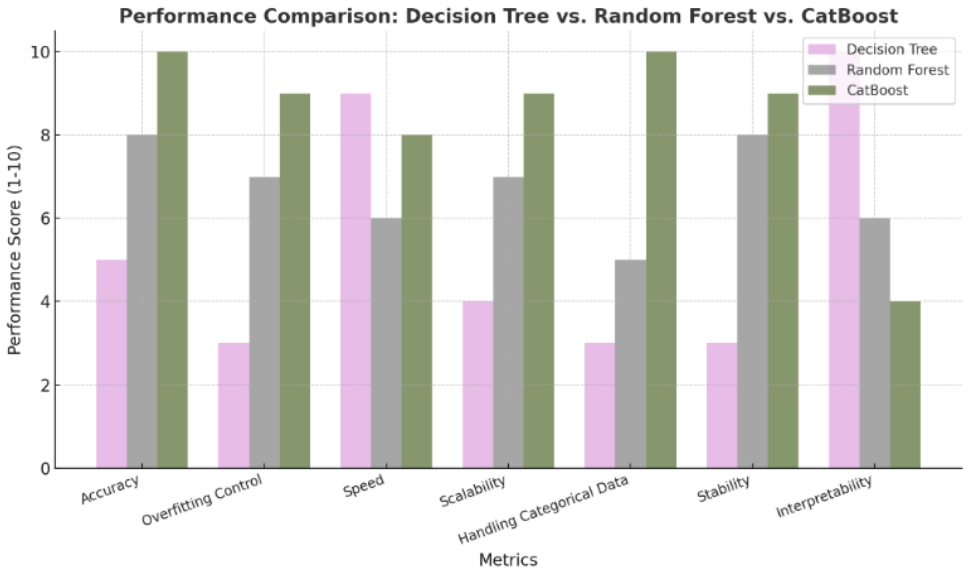


Fig 8.4 comparison Graph

Random Forest combines multiple trees to improve accuracy and reduce overfitting, making it more stable than a single Decision Tree.It is computationally expensive and requires categorical encoding. CatBoost, on the other hand, excels in handling categorical features natively and offers high accuracy with minimal overfitting.

8.3 WebApp Results

The web application dashboard for predicting mobile addiction levels is designed with a clean and intuitive interface. The first image displays an input form where users can answer a set of questions related to their mobile usage habits. The left side of the screen features a navigation menu with options such as "Home," "Prediction," and "Logout," ensuring ease of access to different sections of the application. The main content area presents a questionnaire with multiple questions, each offering a simple "Yes" or "No" dropdown for responses.

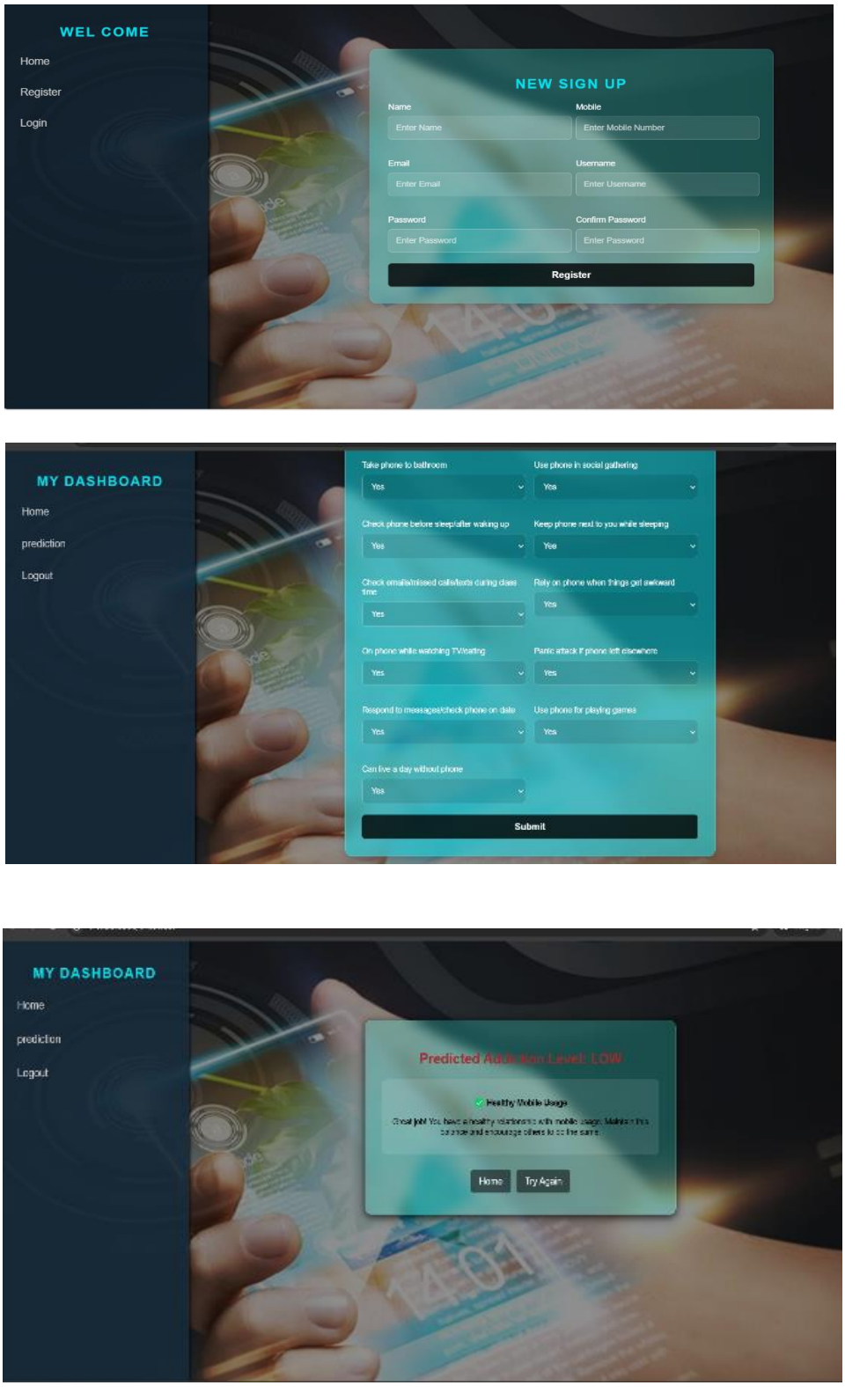


Fig 8.5 User Interface

The web application successfully collects user inputs, processes them, and provides meaningful feedback on mobile addiction levels. The futuristic background and modern UI elements contribute to an appealing visual experience. The system is functional and provides users with valuable insights into their mobile usage habits.

The questions focus on aspects like checking the phone before sleeping, using the phone in social gatherings, and feeling anxious when the phone is not around. At the bottom of the form, a "Submit" button allows users to process their responses and receive an analysis.

CONCLUSION

In conclusion, the proposed ML-based predictive model for mobile addiction offers a data-driven, scalable, and efficient solution to identify and address mobile addiction patterns. By utilizing machine learning algorithms, the system provides real-time monitoring, personalized predictions, and actionable insights based on user behavior. Unlike traditional methods, which often rely on self-reported data or manual interventions, this system delivers objective, automated, and timely assessments that can proactively guide users towards healthier mobile usage. With its ability to scale, provide tailored recommendations, and evaluate model performance through various metrics, the system presents a promising approach to tackling the growing concern of mobile addiction, fostering better mental health, and improving well-being in a technology-driven world.

Future Scope of Project

The future scope of this project lies in enhancing its predictive capabilities and expanding its reach. As mobile addiction becomes an increasing concern worldwide, there is a growing need for more personalized and accurate models. Future work can focus on integrating additional data sources, such as real-time mobile usage statistics, social media interactions, and user demographics, to improve prediction accuracy. The incorporation of deep learning techniques could also enable more complex behavioral pattern recognition. Furthermore, the project can be extended to offer real-time interventions or feedback based on predictive results, guiding users towards healthier mobile habits through personalized suggestions or notifications.

Additionally, the system could be adapted to work across various platforms (e.g., mobile apps, web apps) and integrated into health management systems, allowing for widespread use in clinical settings, schools, and workplaces. By continuously improving model accuracy, u ser engagement, and data privacy, the project could contribute significantly to combating mobile addiction and promoting mental well-being on a global scale.

REFERENCES

1. Choi, Y., & Park, H. (2019). "Mobile addiction and mental health problems among adolescents." *Journal of Child and Adolescent Psychiatric Nursing*, 32(4), 181-188.

2. Chen, L., & Zhao, S. (2018). "Behavioral prediction model for mobile app addiction using machine learning." Computers in Human Behavior, 81, 183-190.

3. Przybylski, A. K., & Weinstein, N. (2017). "Can you contact me now? How the presence of mobile communication technology influences face-to-face conversation quality." Journal of Social and Personal Relationships, 34(4), 626-643.

4. Kuss, D. J., & Griffiths, M. D. (2017). "Social networking sites and addiction: Ten lessons learned." International Journal of Environmental Research and Public Health, 14(3), 311-319.

5. Bian, M., & Leung, L. (2015). "The influence of mobile phone use on the mental health of young adults." Computers in Human Behavior, 48, 126-132.

6. Vallerand, R. J., et al. (2014). "Self-determination theory and the study of human motivation." Handbook of Motivation Science, 35-57.

7. González, P. D., et al. (2018). "Mobile phone addiction and psychological well-being: Evidence from a large cross-sectional study in China." Computers in Human Behavior, 81, 135-144.

8. Cao, H., et al. (2011). "The relationship between mobile phone addiction and depression among adolescents: A cross-sectional study." Journal of Affective Disorders, 132(1-2), 214-219.

9. Goh, C. F., & Chua, Y. P. (2019). "An analysis of mobile phone usage among adolescents and its implications for addiction prevention." *Computers in Human Behavior*, 92, 72-77.

10. Shah, D., & Shah, D. (2020). "Mobile addiction and its relationship with academic performance." Education and Information Technologies, 25(4), 3383-3400.

11. Pradeep, S., & Rani, S. (2020). "Prediction of addiction risk using machine learning models." International Journal of Advanced Research in Computer Science, 11(5), 42-48.

12. Hussain, S., & Anwar, M. (2021). "Exploring machine learning algorithms for prediction of mobile addiction." Journal of Intelligent Systems, 30(2), 453-463.

13. Harris, S., & Leung, M. (2018). "Addictive behaviors on mobile applications: A systematic review." Behavioral Research and Therapy, 112, 1-8.

14. Zhou, Y., et al. (2018). "Mobile phone addiction and the role of fear of missing out (FoMO)." Computers in Human Behavior, 85, 107-118.

15. Sundar, S. S., & Marathe, S. S. (2010). "Personalization in mobile devices: The role of contextual relevance." Human Communication Research, 36(4), 379-406.

16. Gao, Y., & Zhang, Y. (2020). "Exploring machine learning models for prediction of mobile phone addiction in adolescents." Computational Intelligence and Neuroscience, 2020.

17. Yang, Z., & Zhang, X. (2019). "Evaluating the relationship between mobile addiction and mental health: A machine learning perspective." Journal of Behavioral Addictions, 8(3), 320-329.

18. Ariely, D., & Shapira, Y. (2020). "Predicting addictive behaviors in mobile applications using machine learning." *Mobile Networks and Applications*, 25(1), 123-136.

19. Huang, C., & Zhang, W. (2019). "Predicting mobile addiction levels in adolescents using supervised machine learning." *Psychiatry Research*, 271, 133-139.

20. Kaur, R., & Singh, J. (2019). "Analyzing the relationship between mobile usage and addiction using machine learning techniques." *International Journal of Advanced Computing Science and Engineering*, 5(1), 15-28.

PUBLICATION CERTIFICATES

SCITEPRESS DIGITAL LIBRARY Scopus

ICRDICCT CONFERENCE PROCEEDINGS 2025.

International Conference on

Research and Development in Information, Communication and Computing Technologies

(ICRDICCT'25)

This is to certify that

Kondanna Kanamaneni

has successfully presented the paper titled

Prediction Of Smart Phone Addiction Among Students Using Gradient Boosting Algorithm

in Proceedings of the International Conference on Research and Development in Information, Communication and Computing Technologies (ICRDICCT'25), held on April 4 & 5, 2025, at E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu, India.

Organizing Secretary
Dr.S.Manikandan

Industry Partner
Ms.Gugapriyaa Sivakumar
NTL Technology

Principal
Dr.S.Ramabalan

ORGANIZED BY

E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu, India.

&

NTL Technology
Erode, Tamil Nadu, India.

NTLTECHNOLOGY
New Thinking and Learning

SCITEPRESS DIGITAL LIBRARY Scopus

ICRDICCT CONFERENCE PROCEEDINGS 2025.

International Conference on

Research and Development in Information, Communication and Computing Technologies (ICRDICCT'25)

This is to certify that

P.Vasundhara

has successfully presented the paper titled

Prediction Of Smart Phone Addiction Among Students Using Gradient Boosting Algorithm

in Proceedings of the International Conference on Research and Development in Information, Communication and Computing Technologies (ICRDICCT'25), held on April 4 & 5, 2025, at E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu, India.

Organizing Secretary	Industry Partner	Principal
Dr.S.Manikandan	Ms.Gugapriyaa Sivakumar NTL Technology	Dr.S.Ramabalan

ORGANIZED BY

E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu, India.

&

NTL Technology
Erode, Tamil Nadu, India.

NTLTECHNOLOGY
New Thinking and Learning

SCITEPRESS DIGITAL LIBRARY Scopus

ICRDICCT CONFERENCE PROCEEDINGS 2025.

International Conference on

Research and Development in Information, Communication and Computing Technologies (ICRDICCT'25)

This is to certify that

B. Sruthi

has successfully presented the paper titled

Prediction Of Smart Phone Addiction Among Students Using Gradient Boosting Algorithm

in Proceedings of the International Conference on Research and Development in Information, Communication and Computing Technologies (ICRDICCT'25), held on April 4 & 5, 2025, at E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu, India.

Organizing Secretary	Industry Partner	Principal
Dr.S.Manikandan	Ms.Gugapriyaa Sivakumar NTL Technology	Dr.S.Ramabalan

ORGANIZED BY

E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu, India.

&

NTL Technology
Erode, Tamil Nadu, India.

NTLTECHNOLOGY
New Thinking and Learning

SCITEPRESS DIGITAL LIBRARY Scopus

ICRDICCT CONFERENCE PROCEEDINGS 2025.

International Conference on

Research and Development in Information, Communication and Computing Technologies
(ICRDICCT'25)

This is to certify that

M. Subahan

has successfully presented the paper titled

Prediction Of Smart Phone Addiction Among Students Using Gradient Boosting Algorithm

in Proceedings of the International Conference on Research and Development in Information, Communication and Computing Technologies (ICRDICCT'25), held on April 4 & 5, 2025, at E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu, India.

Organizing Secretary	Industry Partner	Principal
Dr.S.Manikandan	Ms.Gugapriyaa Sivakumar NTL Technology	Dr.S.Ramabalan

ORGANIZED BY

E.G.S. Pillay Engineering College,
Nagapattinam, Tamil Nadu, India.
&
NTL Technology
Erode, Tamil Nadu, India.

NTLTECHNOLOGY
New Thinking and Learning

SCITEPRESS DIGITAL LIBRARY Scopus

ICRDICCT CONFERENCE PROCEEDINGS 2025.

International Conference on

Research and Development in Information, Communication and Computing Technologies (ICRDICCT'25)

This is to certify that

K. Santosh Kumar

has successfully presented the paper titled

Prediction Of Smart Phone Addiction Among Students Using Gradient Boosting Algorithm

in Proceedings of the International Conference on Research and Development in Information, Communication and Computing Technologies (ICRDICCT'25), held on April 4 & 5, 2025, at E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu, India.

Organizing Secretary	Industry Partner	Principal
Dr.S.Manikandan	Ms.Gugapriyaa Sivakumar NTL Technology	Dr.S.Ramabalan

ORGANIZED BY

E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu, India.

&

NTL Technology Erode, Tamil Nadu, India.

NTLTECHNOLOGY
New Thinking and Learning