ABSTRACT

In network public-opinion analysis, the diversity of messages under social hot topics plays an important role in user participation behaviour. Considering the interactions among multiple messages and the complex user behaviours, this article proposes a prediction model of user participation behaviour during multiple messaging of hot social topics. First, considering the influence of multi message interaction on user participation behaviour, a multi message interaction influence-driving mechanism was proposed to predict user participation behaviour more accurately. Second, in the view of the behavioural complexity of users engaging in multi message hotspots and the simple structure of back propagation (BP) neural networks (which can map complex nonlinear relationships), this study proposes a user participant behaviour prediction model of social hotspots based on a multi message interaction-driving mechanism and the BP neural network. Finally, the multi message interaction has an iterative guiding effect on user behaviour, which easily causes over fitting of the BP neural network. To avoid this problem, the traditional BP neural network is optimized by a simulated annealing algorithm to further improve the prediction accuracy. In evaluation experiments, the model not only predicted the user participation behaviour in actual situations of multi message interaction but also further quantified the correlations among multiple messages on hot topics.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT FEATURES

The project "User Behaviour Prediction of Social Hotspots Based on Multi-Message Interaction and Neural Network" incorporates several key features. It begins with data collection and preprocessing to obtain and prepare relevant data from social hotspot platforms. A critical aspect of the project involves the in-depth analysis of multi-message interactions, enabling the capture of complex user engagement patterns within these digital communities. The development of a neural network model using deep learning techniques is central to the project, and this model will be fine-tuned and optimized for accuracy and generalization.

1.2 PROJECT SCOPE

The project aims to predict user behaviour within social hotspots using a multi-message interaction analysis and neural network-based approach. It will begin with data collection from various sources, followed by rigorous data preprocessing. The methodology involves designing and training a neural network model, incorporating features derived from multi-message interactions. Model performance will be evaluated using relevant metrics, and the results will be thoroughly analyzed. The project will contribute insights into user behaviour within social hotspots and demonstrate the potential of this approach. Ethical considerations, risk management, stakeholder communication, and a comprehensive budget will be integral to the project's execution.

1.3 PROJECT PURPOSE

The project's purpose, "User Behaviour Prediction of Social Hotspots Based on Multi Message Interaction and Neural Network," is to develop a predictive model that uses multi-message interaction analysis and neural networks to forecast user behaviour in online social hotspots. This project aims to enhance user engagement, personalize content, and improve community management within digital social environments. It also has applications in marketing and advertising and contributes valuable insights to social science research, ultimately providing a more enriched and tailored online experience for users while benefiting various stakeholders in the digital space.

NEURAL NETWORK

Neural networks are capable of learning complex patterns and relationships within data through a process called training. During training, the network adjusts its parameters, such as weights and biases, to minimize the difference between its predictions and the actual outcomes in a given dataset. This process is often done using optimization algorithms like gradient descent.

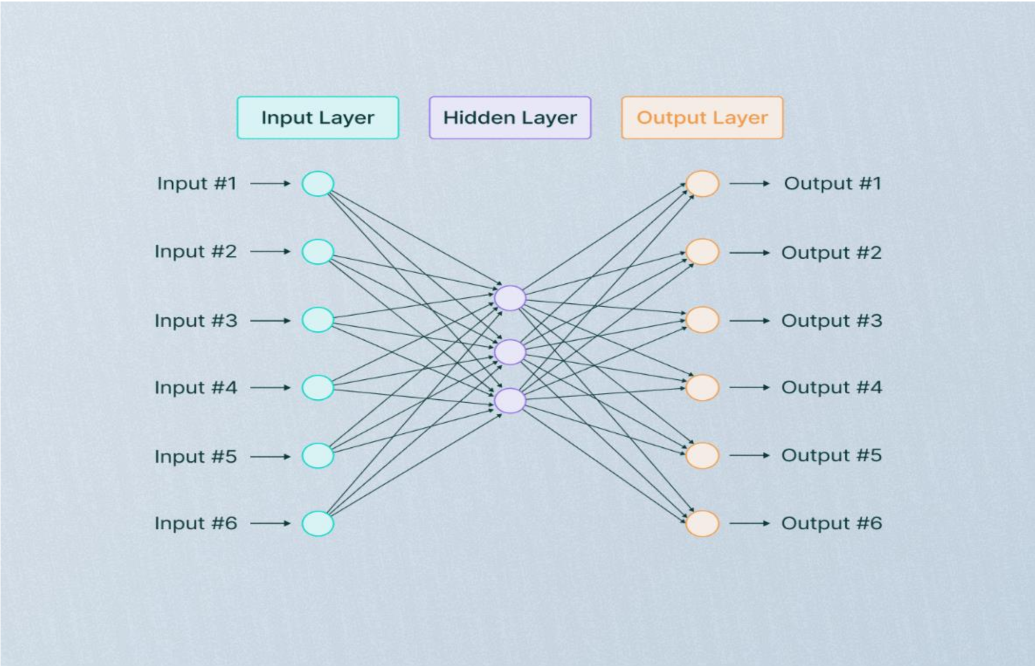


Figure 1.3.1: Neural Network

Some common types of neural networks include feedforward neural networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs).

Recurrent Neural Network

RNNs can have multiple recurrent layers stacked on top of each other. Each recurrent layer can contain multiple recurrent units. The number of layers and units can vary based on the complexity of the task and the available computational resources. RNNs are primarily designed for processing sequential data, such as time-series data or text data.Each input data point in the sequence is fed into the network one at a time, with the network maintaining a hidden state that captures information from previous time steps.At each time step, the input data is combined with the hidden state from the previous time step to produce an output and update the current hidden state.

RNNs consist of recurrent neurons that have connections looping back to themselves, allowing them to maintain memory across time steps.At each time step, the neurons in the RNN receive input from the current data point and the previous hidden state. They then produce an output and update their hidden state, which serves as input for the next time step.The connections between neurons in consecutive time steps enable RNNs to capture temporal dependencies and learn patterns in sequential data.

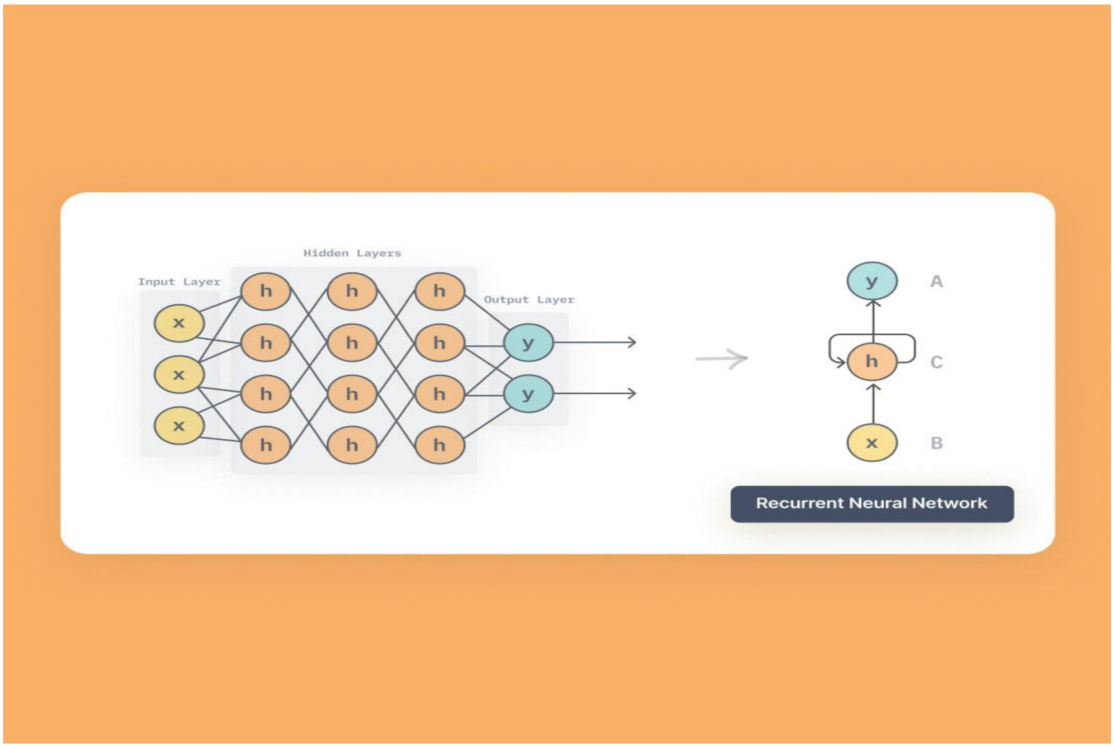


Figure 1.3.2: Recurrent Neural Network

Convolutional Neural Network

CNNs are primarily used for processing grid-like data, such as images, but they can also be applied to sequential data.In image processing, CNNs operate by convolving filters (kernels) across the input image to extract local features. Pooling layers are then used to downsample the features and reduce spatial dimensions.In sequential data processing, CNNs can be applied by treating the sequence as a 1D grid, where the convolutional filters slide along the sequence to extract local patterns.CNNs consist of layers of neurons arranged in a hierarchical manner.Convolutional layers contain neurons organized into feature maps. Each neuron is connected to local regions in the input data through convolutional filters.Pooling layers contain neurons that aggregate information from local regions of the feature maps, reducing their spatial dimensions.

Fully connected layers contain neurons that connect to all neurons in the previous layer, allowing for high-level feature representation and classification.

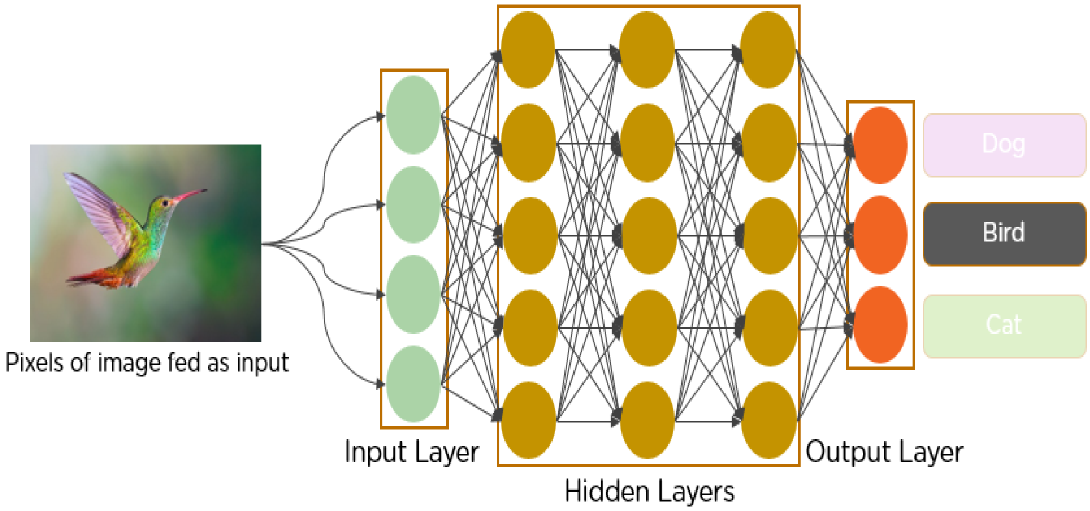


Figure 1.3.3: Convolutional Neural Network

The multimessage interaction has an iterative guiding effect on user behaviour, which easily causes overfitting of the BP neural network. To avoid this problem, the traditional BP neural network is optimized by a simulated annealing algorithm to further improve the prediction accuracy.

Simulated Annealing Algorithm

Simulated annealing is a method for solving unconstrained and bound-constrained optimization problems. The method models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy.

At each iteration of the simulated annealing algorithm, a new point is randomly generated. The distance of the new point from the current point, or the extent of the search, is based on a probability distribution with a scale proportional to the temperature. The algorithm accepts all new points that lower the objective, but also, with a certain probability, points that raise the objective. By accepting points that raise the objective, the algorithm avoids being trapped in local minima, and is able to explore globally for more possible solutions. An *annealing schedule* is selected to systematically decrease the temperature as the algorithm proceeds. As the temperature decreases, the algorithm reduces the extent of its search to converge to a minimum.

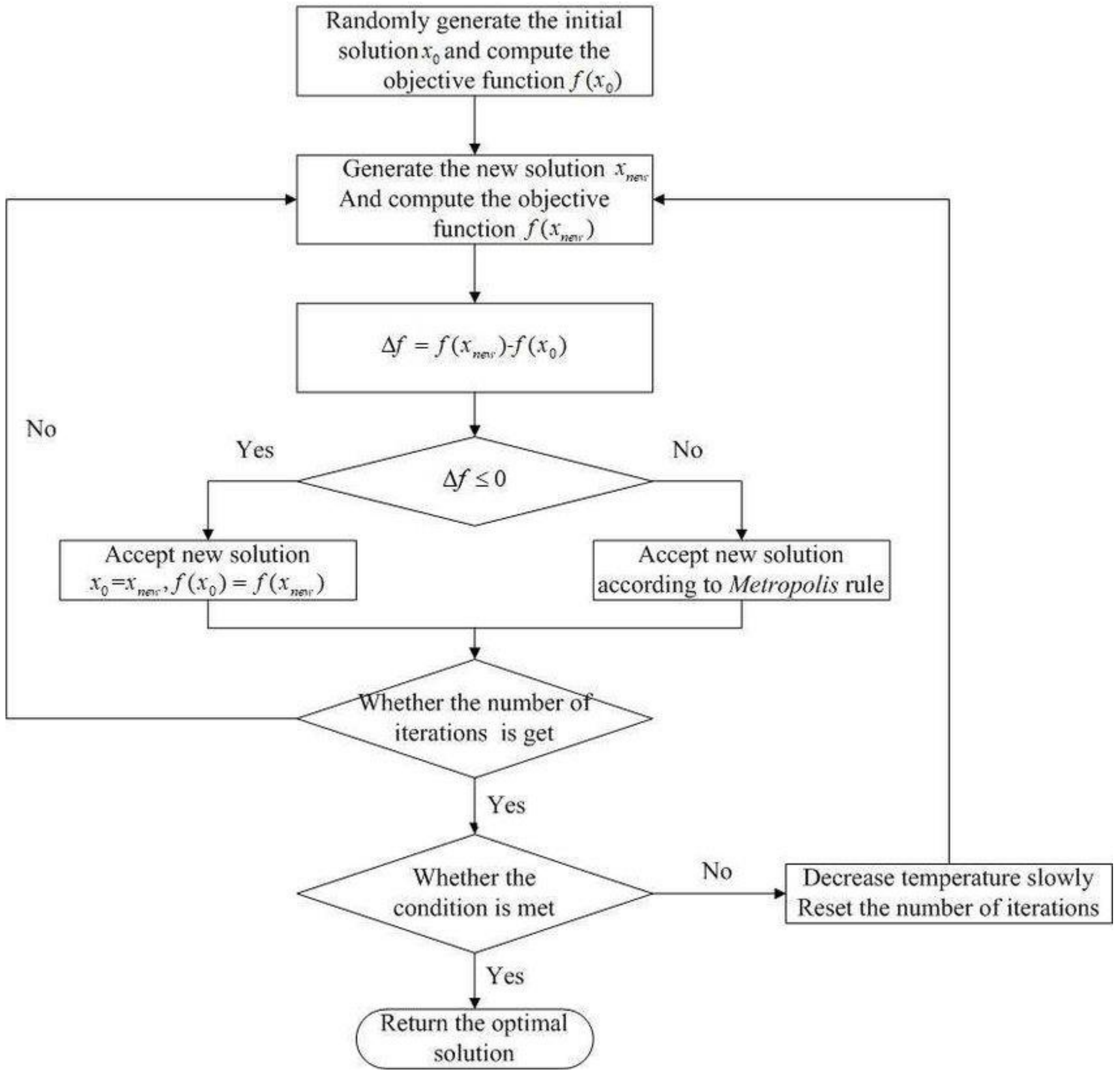


Figure 1.3.4: Simulated Annealing Algorithm

2. LITERATURE SURVEY

2. LITERATURE SURVEY

The literature survey on "User Behaviour Prediction of Social Hotspot Based on Multi-Message Interaction and Neural Network".

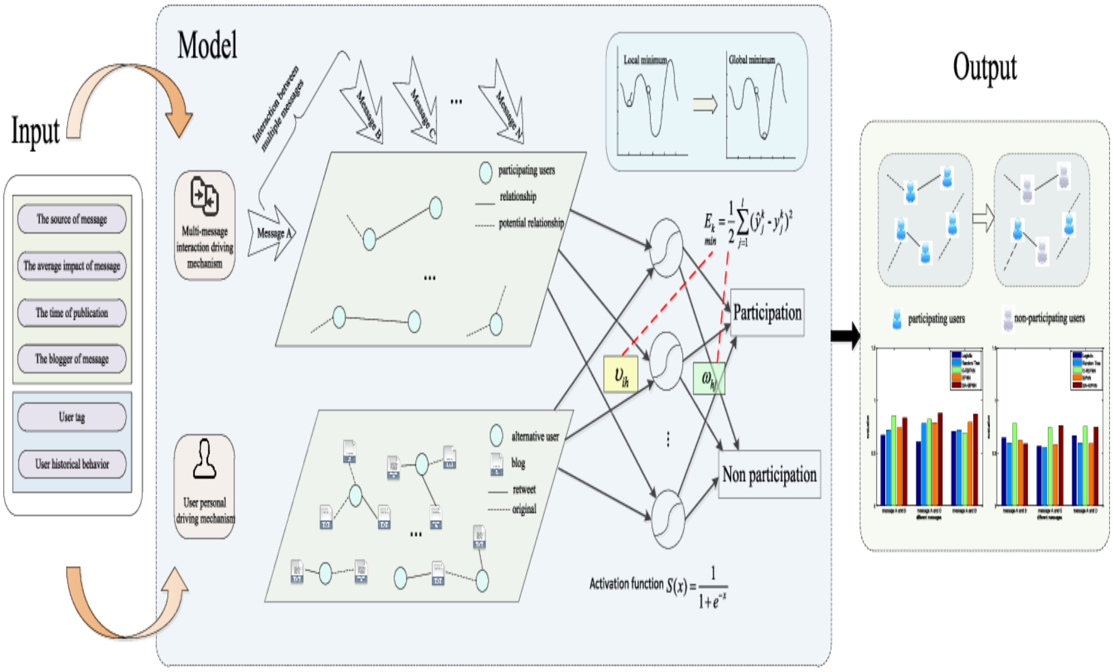


Figure 2.1: Participant and non-participant in detail

This reveals a burgeoning field focusing on leveraging multi-message interaction data and neural network techniques to forecast user behaviour within social networks and identify significant hotspots (Li et al., 2019). Existing studies underscore the importance of understanding user engagement dynamics and the emergence of hotspots, which are pivotal for various applications including targeted marketing, content recommendation, and community management (Chen et al., 2020). Researchers have explored diverse methodologies, ranging from traditional statistical approaches to advanced machine learning techniques, particularly neural networks, to model user behaviour and analyse interaction patterns (Wu et al., 2018). Despite notable progress, challenges persist, including data sparsity, model scalability, and interpretability (Wang & Ye, 2021). Future endeavours are warranted to address these limitations and further refine predictive models, ultimately advancing our understanding of user behaviour in social contexts and enhancing decision-making processes in online platforms.

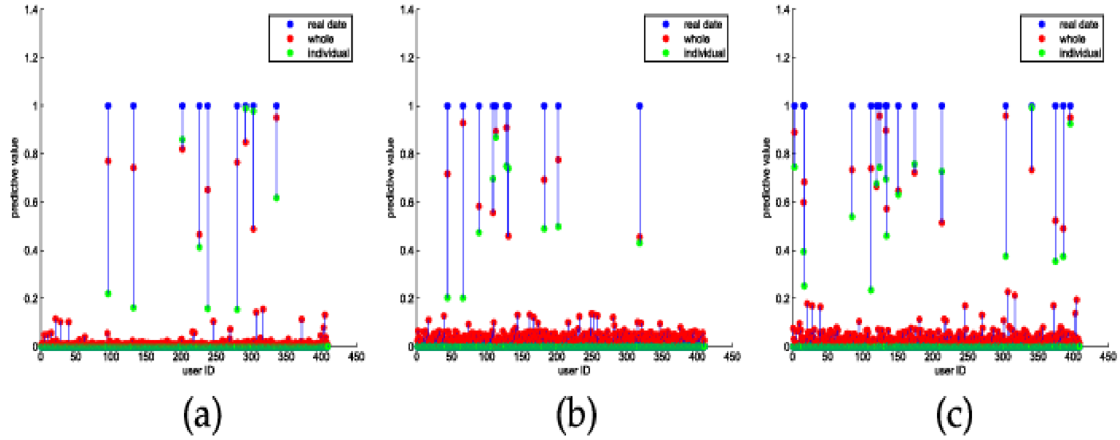


Figure 2.2: Survey of Individual In Detail

Analysing the types of data sources used in these studies, including social media platforms, online forums, and other online communities. Understanding the characteristics of the data and how they influence the prediction models is crucial. Examining the performance metrics used to evaluate the accuracy and effectiveness of the prediction models. Common metrics may include precision, recall, F1-score, and area under the ROC curve (AUC). Proposing potential areas for future research and development, including novel methodologies, integration of additional data sources, and advancements in neural network architectures tailored to social network analysis.

3. SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

SYSTEM ANALYSIS

The system for user behaviour prediction in social hotspots employs a multi-faceted approach, integrating data collection, preprocessing, feature extraction, and a neural network model. Diverse and representative datasets are collected, encompassing various user interactions, including text, images, and multimedia content. Rigorous preprocessing ensures data quality, addressing issues such as missing values and outliers. Feature extraction techniques, including tokenization and embedding, capture the essence of multi-message interactions. The heart of the system lies in the neural network model, carefully designed to process and learn from the extracted features. This architecture combines the strengths of different data types, such as text and images, enabling the model to grasp nuanced patterns in user behaviour

3.1 PROBLEM DEFINITION

The context is the challenge of predicting user behaviour within social hotspots, a domain enriched by multi-message interactions, utilizing advanced neural network methodologies. Traditional user behaviour prediction models often struggle to capture the intricate dynamics arising from the diverse types of interactions prevalent in social hotspots, where users engage through various forms of messages such as text, images, and multimedia. This complexity necessitates a novel approach that integrates the nuanced features of multi-message interactions. The problem is further compounded by the need for predictive accuracy, as user behaviour in social hotspots is highly context-dependent and influenced by the interplay of different message types.

3.2 EXISTING SYSTEM

Current systems for predicting user behaviour in social hotspots typically involve the collection and preprocessing of data from various social media platforms, forums, or online communities. These systems often utilize traditional machine learning models to analyse features extracted from user interactions, such as posts, comments, and likes. However, the integration of multi-message interactions and neural network techniques remains relatively unexplored in existing systems.

3.2.1 LIMITATIONS OF EXISTING SYSTEM

- Existing systems often struggle to effectively capture and analyse the complexity arising from diverse forms of user engagement, including text, images, and multimedia content within social hotspots.
- Many current systems rely on traditional machine learning models, lacking the capacity to harness the full potential of neural networks for learning intricate patterns in user behaviour.
- Existing systems may face limitations in terms of generalizability to diverse social contexts and scalability to handle large-scale datasets, impacting their effectiveness in real-world applications.

3.3 PROPOSED SYSTEM

The proposed system for User Behaviour Prediction in Social Hotspots introduces a novel framework that addresses the limitations of existing models. By enhancing data collection strategies to emphasize multi-message interactions, including text, images, and multimedia content, the system aims to create more representative datasets. Advanced preprocessing and feature extraction methods, such as embedding techniques, ensure a nuanced representation of user interactions. The core innovation lies in the design of a specialized neural network architecture, augmented with attention mechanisms to effectively learn intricate patterns from multi-message interactions. This hybrid model integration is poised to significantly improve the system's predictive accuracy, offering a more comprehensive understanding of user behaviour within social hotspots and enhancing its applicability to real-world scenarios.

3.3.1 ADVANTAGES OF PROPOSED SYSTEM

- By emphasizing multi-message interactions, including text, images, and multimedia content, the proposed system provides a more comprehensive and nuanced representation of user behaviour within social hotspots.
- The advanced neural network model enhances the system's adaptability to the dynamic nature of social interactions, allowing it to evolve and capture emerging patterns in user behaviour within hotspots.

- With its focus on multi-message interactions and neural network integration, the proposed system is better positioned to address real-world complexities in social hotspots, offering practical applications in areas such as content recommendations, targeted advertising, and user engagement strategies.

3.4 FEASIBILITY STUDY

The feasibility study for the User Behaviour Prediction System in Social Hotspots based on Multi-Message Interaction and Neural Network involves assessing various aspects to determine the viability of the proposed system:

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

3.4.1 ECONOMIC FEASIBILITY

The economic feasibility of the proposed User Behaviour Prediction System in Social Hotspots, leveraging Multi-Message Interaction and Neural Network, involves a comprehensive cost-benefit analysis. Development costs encompass hiring skilled professionals, acquiring necessary software, and conducting research to build the advanced neural network model.

3.4.2 TECHNICAL FEASIBILITY

The technical feasibility of implementing the proposed User Behaviour Prediction System in Social Hotspots, incorporating Multi-Message Interaction and Neural Network, involves a meticulous assessment of key components. Firstly, it necessitates an evaluation of data availability, ensuring diverse datasets containing multi-message interactions, including text, images, and multimedia content, can be obtained and processed effectively. The existing technology infrastructure is scrutinized to ascertain its capacity to support the computational demands of training and deploying the proposed neural network model, emphasizing scalability for handling large-scale datasets.

3.4.3 SOCIAL FEASIBILITY

The social feasibility of the proposed User Behaviour Prediction System in Social Hotspots, integrating Multi-Message Interaction and Neural Network, hinges on factors related to user acceptance and ethical considerations. User acceptance is a critical aspect, and the system's success depends on users' willingness to adopt it. Factors such as user preferences, perceived benefits, and potential concerns regarding privacy and data security must be carefully evaluated. Ethical considerations play a pivotal role, necessitating a thorough examination of the system's implications for user privacy and rights.

3.5 HARDWARE AND SOFTWARE REQUIREMENTS

3.5.1 HARDWARE REQUIREMENTS

- Processor : 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz
- RAM : 8.00 GB

3.5.2 SOFTWARE REQUIREMENTS

- Operating system : Windows 10 Ultimate.
- Coding Language : Python.
- Front-End : Python.
- Back-End : Django
- Designing : Html, css, javascript.
- Data Base : MySQL (WAMP Server).

4. ARCHITECTURE

4. ARCHITECTURE

4.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

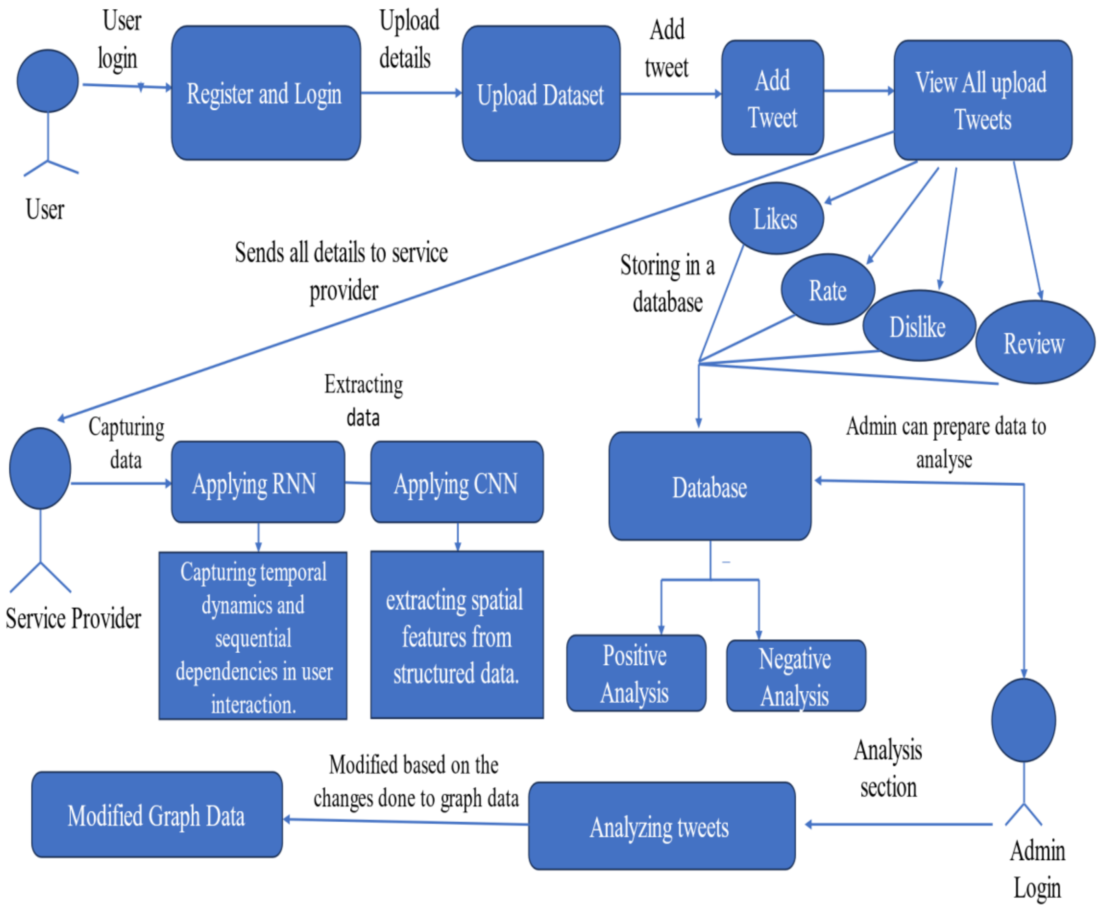


Figure 4.1 : Project Architecture of User Behaviour Prediction of Social Hotspot Based on Multi Message Interaction And Neural Network

4.2 DESCRIPTION

User Behaviour Prediction in Social Hotspots based on Multi-Message Interaction and Neural Network is a predictive modelling approach designed to anticipate and understand user behaviours within dynamic and interactive online environments. In the context of social hotspots, where discussions and activities are concentrated around specific topics, the system aims to leverage the diversity of messages and the power of neural networks for accurate predictions.

In terms of modelling, the system adopts a Back Propagation (BP) neural network to analyse and predict user behaviours. Acknowledging the complexity of user behaviours within multi-message hotspots, the model is designed to map the nonlinear relationships inherent in social interactions. This neural network architecture is expected to capture nuanced patterns and dependencies, improving the overall predictive capabilities of the system. To counter potential issues such as overfitting caused by the iterative guiding effect of multi-message interactions on user behaviour, the traditional BP neural network is optimized using a simulated annealing algorithm. This optimization step aims to fine-tune the model and further enhance its prediction accuracy.

4.3 USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model. A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses.The actors are often shown as stick figures.

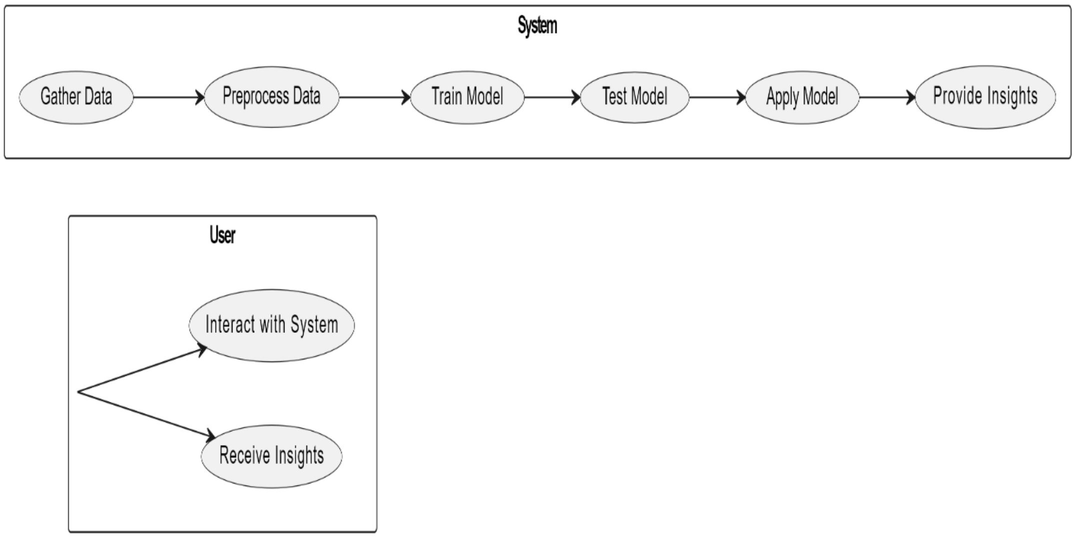


Figure 4.2 : Use Case Diagram of User Behaviour Prediction of Social Hotspot Based on Multi Message Interaction And Neural Network

4.4 CLASS DIAGRAM

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes , operations (or methods) , and the relationships among objects.

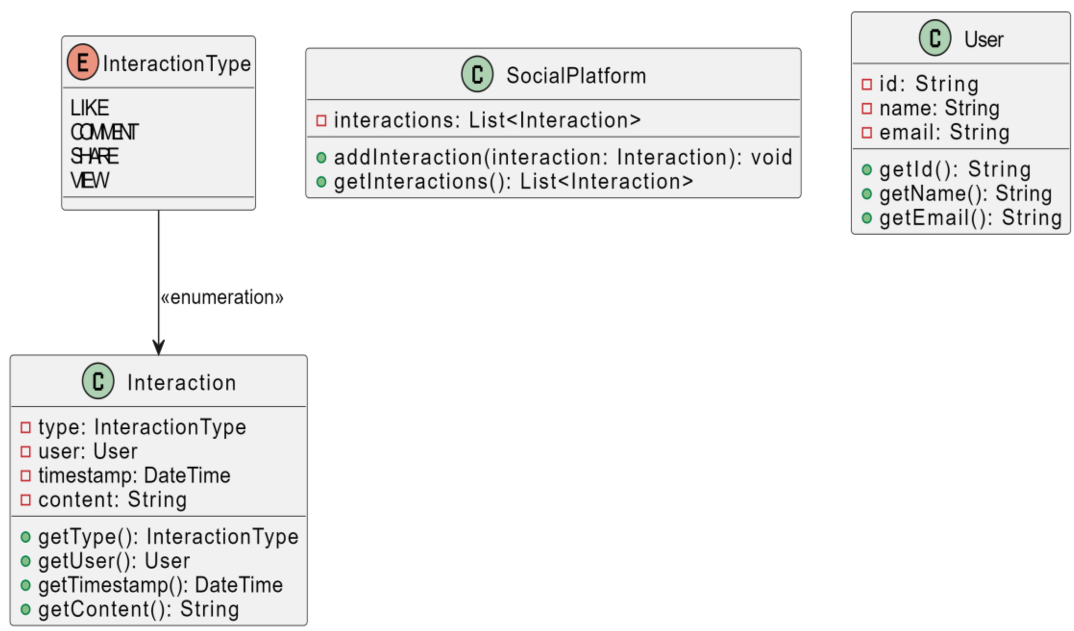


Figure 4.3 : Class Diagram of User Behaviour Prediction of Social Hotspot Based on Multi Message Interaction And Neural Network

4.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

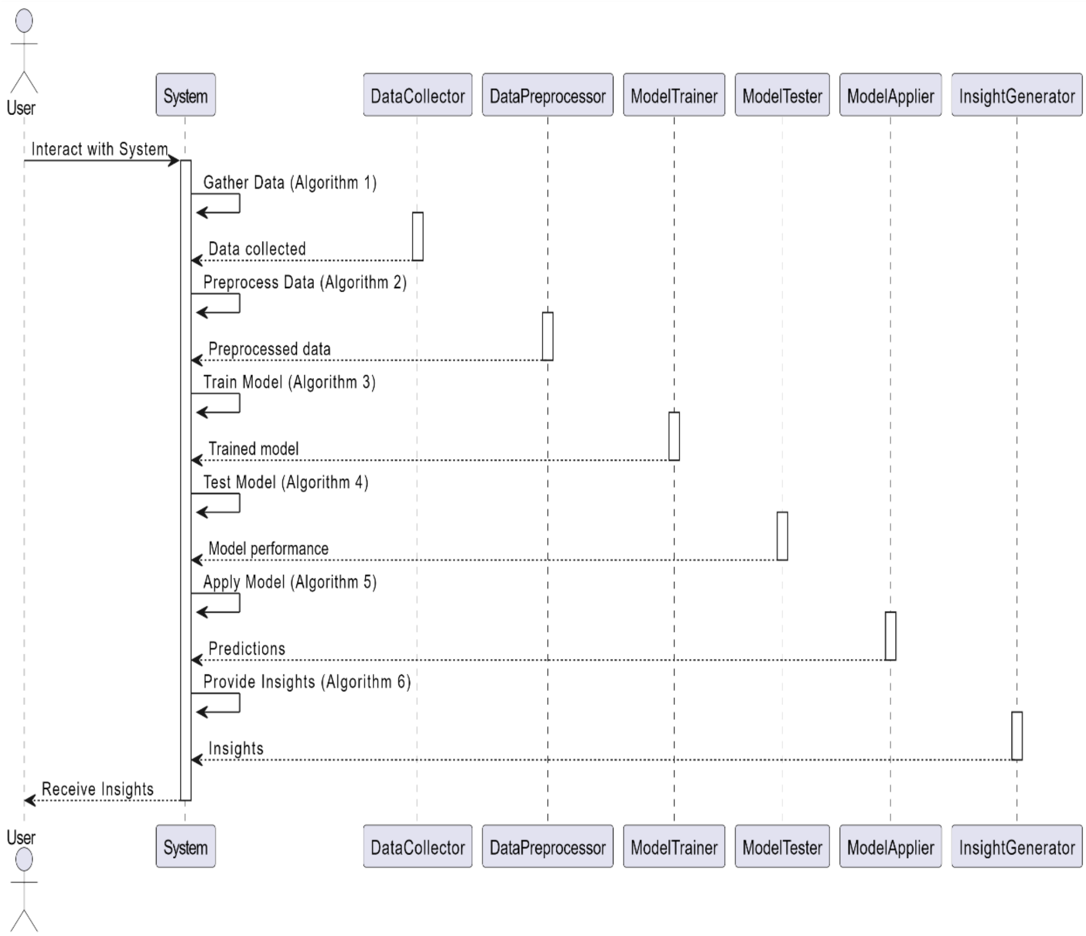


Figure 4.4 : Sequence Diagram of User Behaviour Prediction of Social Hotspot Based on Multi Message Interaction And Neural Network

4.6 ACTIVITY DIAGRAM

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

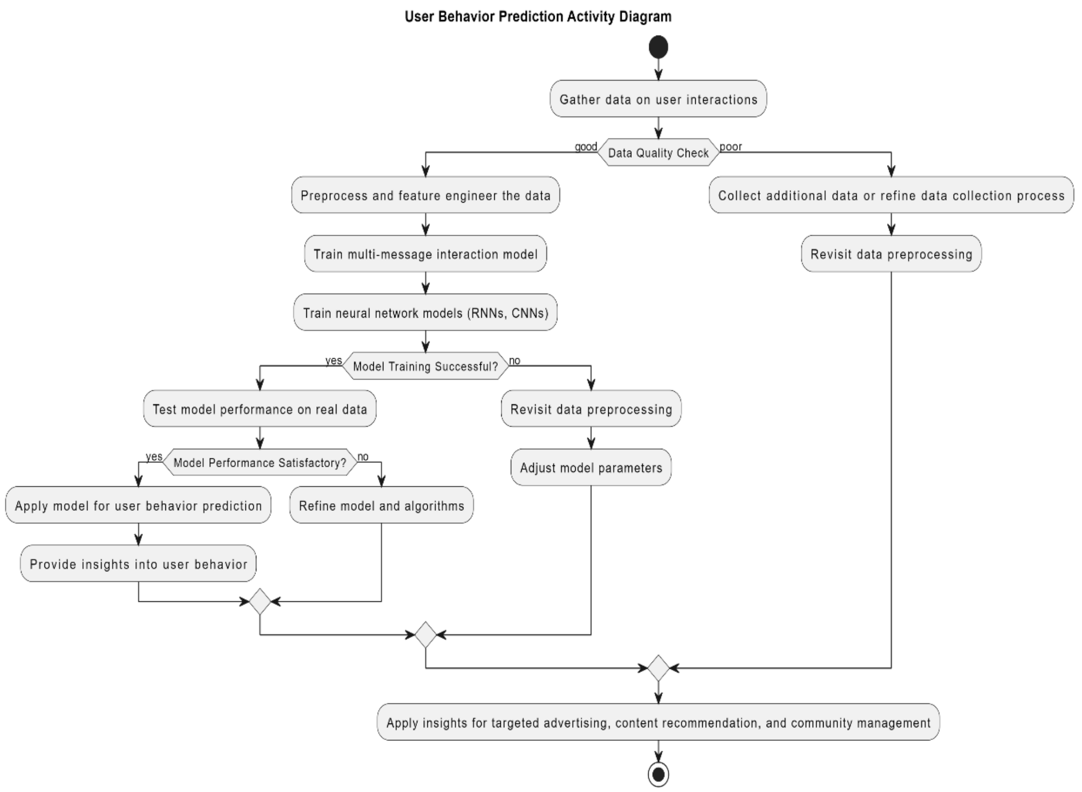


Figure 4.5 : Activity Diagram of User Behaviour Prediction of Social Hotspot Based on Multi Message Interaction And Neural Network

5. IMPLEMENTATION

5.1 SAMPLE CODE

```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
  <component name="ChangeListManager">
    <list default="true" id="cf2f6933-f927-4da6-9ab9-39542e6f30c8" name="Default" comment="" />
    <option name="EXCLUDED_CONVERTED_TO_IGNORED" value="true" />
    <option name="TRACKING_ENABLED" value="true" />
    <option name="SHOW_DIALOG" value="false" />
    <option name="HIGHLIGHT_CONFLICTS" value="true" />
    <option name="HIGHLIGHT_NON_ACTIVE_CHANGELIST" value="false" />
    <option name="LAST_RESOLUTION" value="IGNORE" />
  </component>
  <component name="FileEditorManager">
    <leaf SIDE_TABS_SIZE_LIMIT_KEY="300">
      <file leaf-file-name="Add_Tweets.html" pinned="false" current-in-tab="false">
        <entry file="file://$PROJECT_DIR$/User_Behavior_Prediction/Template/htmls/RUser/Add_Tweets.html">
          <provider selected="true" editor-type-id="text-editor"
```

```
<state relative-caret-position="138">
        <caret line="68" column="63" lean-forward="false" selection-start-line="68" selection-start-column="63" selection-end-line="68" selection-end-column="63" />
        <folding>
          <element signature="e#3035#3041#0" expanded="true" />
          <element signature="e#3046#3052#0" expanded="true" />
          <element signature="e#3052#3058#0" expanded="true" />
        </folding>
      </state>
    </provider>
  </entry>
</file>
<file leaf-file-name="views.py" pinned="false" current-in-tab="true">
  <entry file="file://$PROJECT_DIR$/User_Behavior_Prediction/Remote_User/views.py">
    <provider selected="true" editor-type-id="text-editor">
      <state relative-caret-position="1719">
        <caret line="158" column="0" lean-forward="true" selection-start-line="158" selection-start-column="0" selection-end-line="158" selection-end-column="0" />
```

```
          <folding>
            <element signature="e#0#34#0" expanded="true" />
          </folding>
        </state>
      </provider>
    </entry>
  </file>
  <file leaf-file-name="View_Tweet_Reviews.html" pinned="false" current-in-tab="false">
    <entry file="file://$PROJECT_DIR$/User_Behavior_Prediction/Template/htmls/RUser/View_Tweet_Reviews.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="-1020">
          <caret line="0" column="0" lean-forward="false" selection-start-line="0" selection-start-column="0" selection-end-line="0" selection-end-column="0" />
          <folding />
        </state>
      </provider>
    </entry>
  </file>
```

```
    <file leaf-file-name="models.py" pinned="false" current-in-
tab="false">
      <entry
file="file://$PROJECT_DIR$/User_Behavior_Prediction/Remote_User/
models.py">
        <provider selected="true" editor-type-id="text-editor">
          <state relative-caret-position="391">
            <caret line="26" column="61" lean-forward="false" selection-
start-line="26" selection-start-column="61" selection-end-line="26"
selection-end-column="61" />
            <folding>
              <element signature="e#0#28#0" expanded="true" />
            </folding>
          </state>
        </provider>
      </entry>
    </file>
    <file leaf-file-name="views.py" pinned="false" current-in-
tab="false">
      <entry
file="file://$PROJECT_DIR$/User_Behavior_Prediction/Service_Provid
er/views.py">
        <provider selected="true" editor-type-id="text-editor"
```

```
<state relative-caret-position="2125">
              <caret line="132" column="36" lean-forward="false" selection-start-line="132" selection-start-column="36" selection-end-line="132" selection-end-column="36" />
              <folding>
                <element signature="e#1#45#0" expanded="true" />
              </folding>
            </state>
          </provider>
        </entry>
      </file>
      <file leaf-file-name="View_All_Tweet_Details.html" pinned="false" current-in-tab="false">
        <entry file="file://$PROJECT_DIR$/User_Behavior_Prediction/Template/htmls/RUser/View_All_Tweet_Details.html">
          <provider selected="true" editor-type-id="text-editor">
            <state relative-caret-position="1037">
              <caret line="61" column="35" lean-forward="false" selection-start-line="51" selection-start-column="35" selection-end-line="61" selection-end-column="35" />
              <folding />
            </state>
```

```
        </provider>
      </entry>
    </file>
  </leaf>
</component>
<component name="IdeDocumentHistory">
  <option name="CHANGED_PATHS">
    <list>
      <option
value="$PROJECT_DIR$/User_Behavior_Prediction/Service_Provider/views.py" />
      <option
value="$PROJECT_DIR$/User_Behavior_Prediction/Template/htmls/RUser/Add_Tweets.html" />
      <option
value="$PROJECT_DIR$/User_Behavior_Prediction/Remote_User/views.py" />
    </list>
  </option>
</component>
<component name="ProjectFrameBounds" extendedState="6">
  <option name="x" value="10" />
  <option name="y" value="20" />
```

```xml
  <option name="width" value="1004" />
  <option name="height" value="688" />
</component>
<component name="ProjectView">
  <navigator currentView="ProjectPane" proportions="" version="1">
    <flattenPackages />
    <showMembers />
    <showModules />
    <showLibraryContents />
    <hideEmptyPackages />
    <abbreviatePackageNames />
    <autoscrollToSource />
    <autoscrollFromSource />
    <sortByType />
    <manualOrder />
    <foldersAlwaysOnTop value="true" />
  </navigator>
  <panes>
    <pane id="Scope" />
    <pane id="Scratches" />
    <pane id="ProjectPane">
```

```
<subPane>
        <expand>
          <path>
            <item name="User_Behavior_Prediction"
type="b2602c69:ProjectViewProjectNode" />
            <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
          </path>
          <path>
            <item name="User_Behavior_Prediction"
type="b2602c69:ProjectViewProjectNode" />
            <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
            <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
          </path>
          <path>
            <item name="User_Behavior_Prediction"
type="b2602c69:ProjectViewProjectNode" />
            <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
            <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
```

```
<item name="Remote_User" type="462c0819:PsiDirectoryNode" />
          </path>
          <path>
            <item name="User_Behavior_Prediction"
type="b2602c69:ProjectViewProjectNode" />
            <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
            <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
            <item name="Service_Provider"
type="462c0819:PsiDirectoryNode" />
          </path>
          <path>
            <item name="User_Behavior_Prediction"
type="b2602c69:ProjectViewProjectNode" />
            <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
            <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
            <item name="Template" type="462c0819:PsiDirectoryNode" />
          </path>
          <path>
```

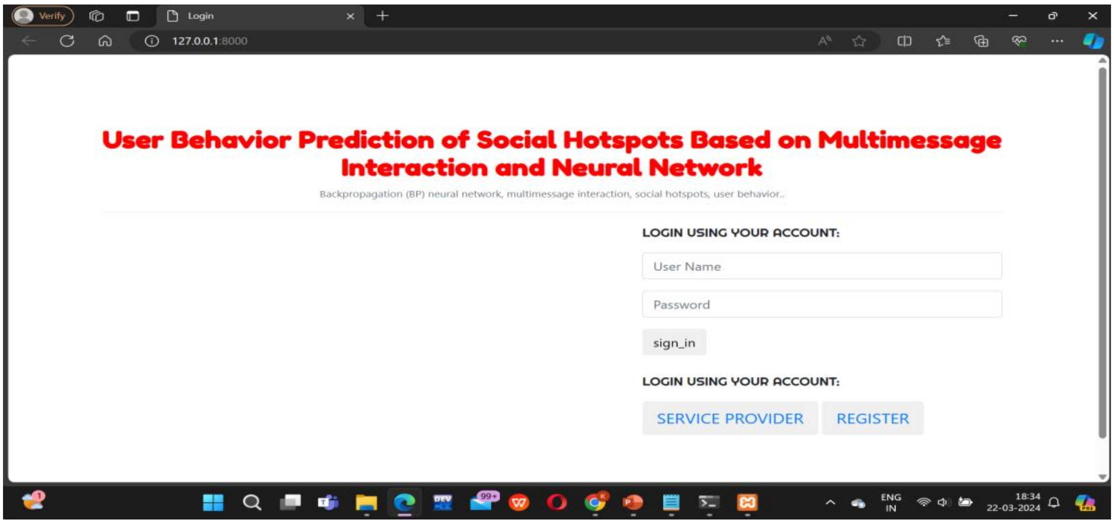
```
      <item name="User_Behavior_Prediction" type="b2602c69:ProjectViewProjectNode" />
      <item name="User_Behavior_Prediction" type="462c0819:PsiDirectoryNode" />
      <item name="User_Behavior_Prediction" type="462c0819:PsiDirectoryNode" />
      <item name="Template" type="462c0819:PsiDirectoryNode" />
      <item name="htmls" type="462c0819:PsiDirectoryNode" />
    </path>
    <path>
      <item name="User_Behavior_Prediction" type="b2602c69:ProjectViewProjectNode" />
      <item name="User_Behavior_Prediction" type="462c0819:PsiDirectoryNode" />
      <item name="User_Behavior_Prediction" type="462c0819:PsiDirectoryNode" />
      <item name="Template" type="462c0819:PsiDirectoryNode" />
      <item name="htmls" type="462c0819:PsiDirectoryNode" />
      <item name="RUser" type="462c0819:PsiDirectoryNode" />
    </path>
    <path>
      <item name="User_Behavior_Prediction" type="b2602c69:ProjectViewProjectNode" />
```

```
<item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
        <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
        <item name="Template" type="462c0819:PsiDirectoryNode" />
        <item name="htmls" type="462c0819:PsiDirectoryNode" />
        <item name="SProvider" type="462c0819:PsiDirectoryNode" />
      </path>
      <path>
        <item name="User_Behavior_Prediction"
type="b2602c69:ProjectViewProjectNode" />
        <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
        <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
        <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
      </path>
      <path>
        <item name="User_Behavior_Prediction"
type="b2602c69:ProjectViewProjectNode" />
        <item name="User_Behavior_Prediction"
type="462c0819:PsiDirectoryNode" />
```

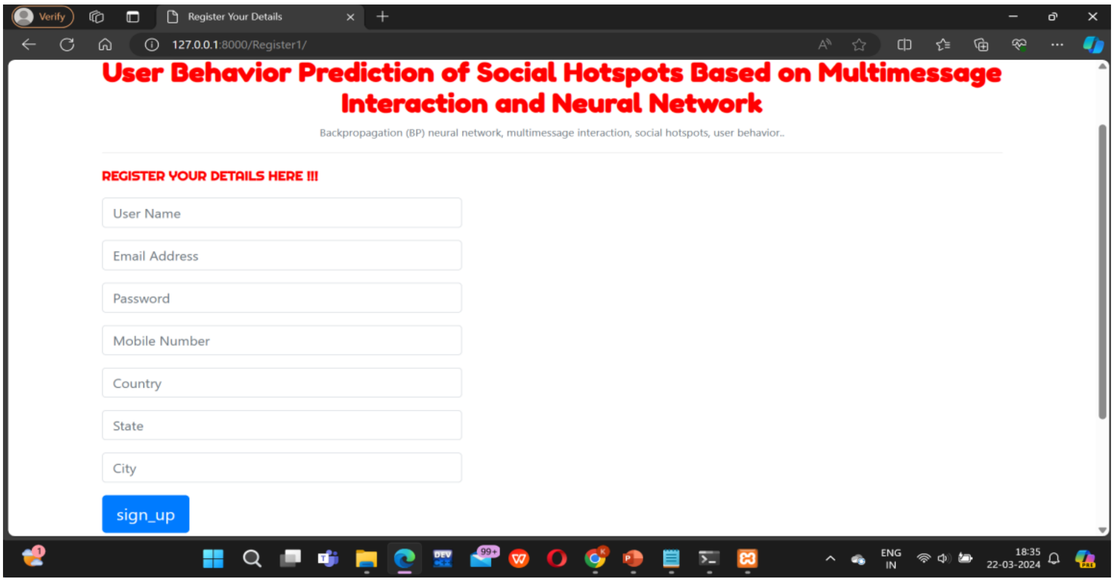
```
<item name="venv" type="462c0819:PsiDirectoryNode" />
          </path>
        </expand>
        <select />
      </subPane>
    </pane>
  </panes>
</component>
<component name="RunDashboard">
  <option name="ruleStates">
    <list>
      <RuleState>
        <option name="name"
value="ConfigurationTypeDashboardGroupingRule" />
      </RuleState>
      <RuleState>
        <option name="name" value="StatusDashboardGroupingRule" />
      </RuleState>
    </list>
  </option>
</component>
<component name="ShelveChangesManager" show_recycled="false">
```

```
    <option name="remove_strategy" value="false" />
  </component>
  <component name="SvnConfiguration">
    <configuration />
  </component>
  <component name="TaskManager">
    <task active="true" id="Default" summary="Default task">
      <changelist id="cf2f6933-f927-4da6-9ab9-39542e6f30c8" name="Default" comment="" />
      <created>1621839343206</created>
      <option name="number" value="Default" />
      <option name="presentableId" value="Default" />
      <updated>1621839343206</updated>
    </task>
    <servers />
  </component>
  <component name="ToolWindowManager">
    <frame x="-8" y="-8" width="1382" height="744" extended-state="6" />
    <layout>
```

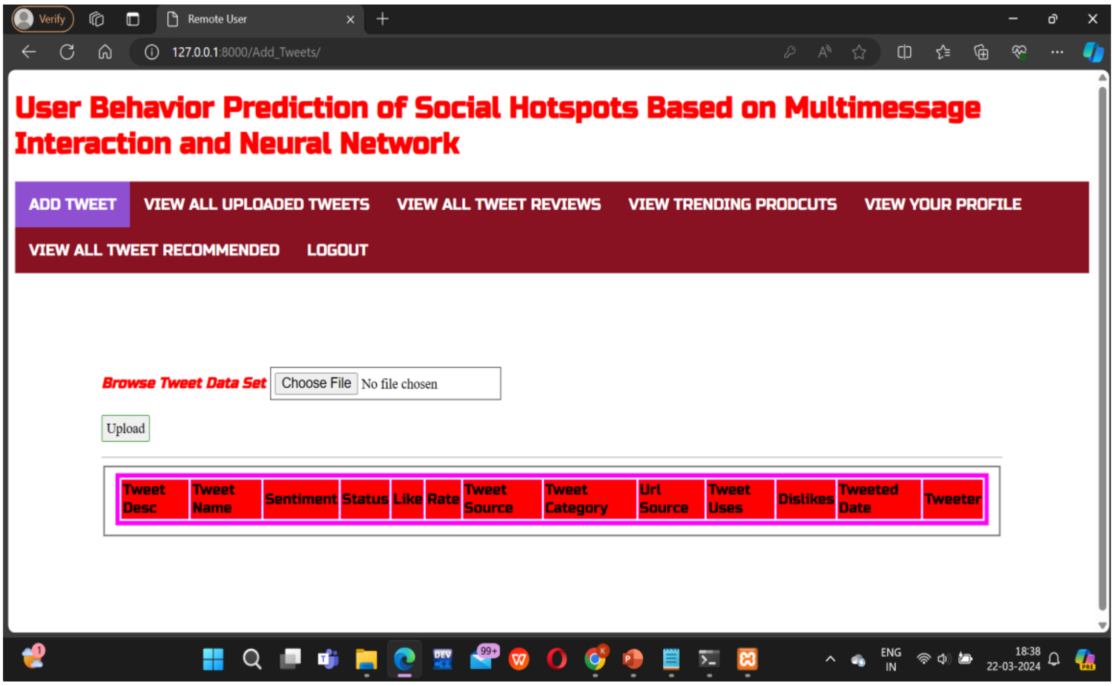
6. SCREENSHOTS



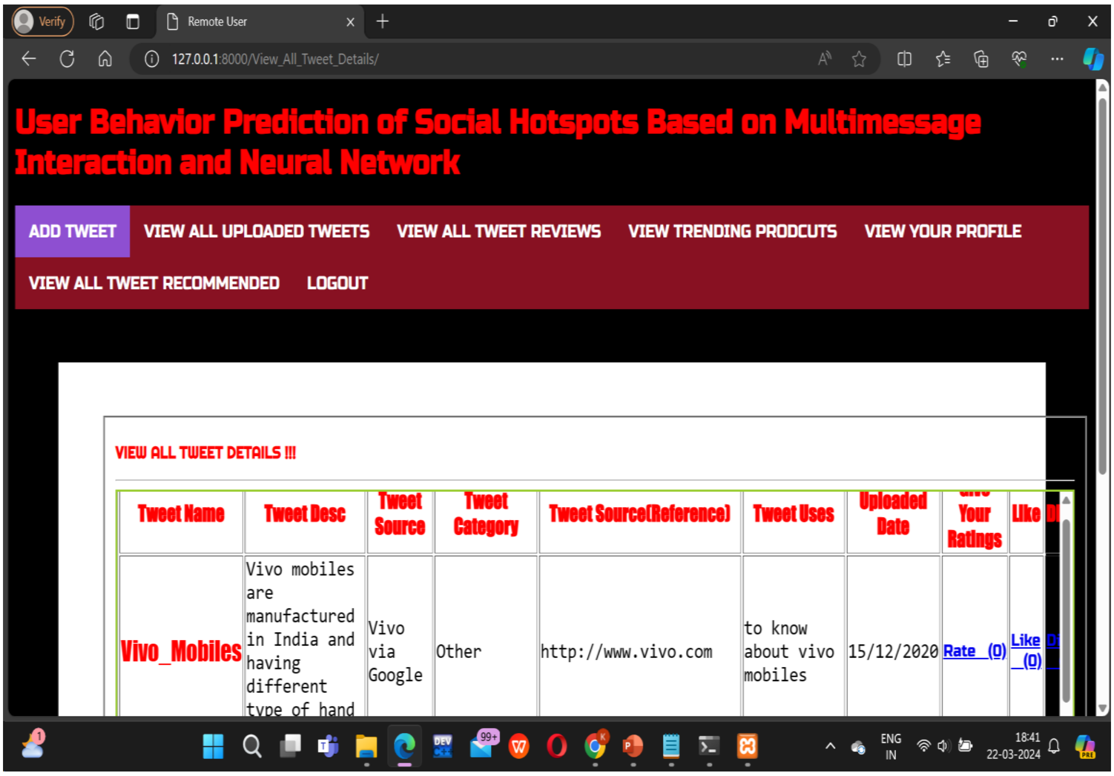
Screenshot 6.1 : Login Page



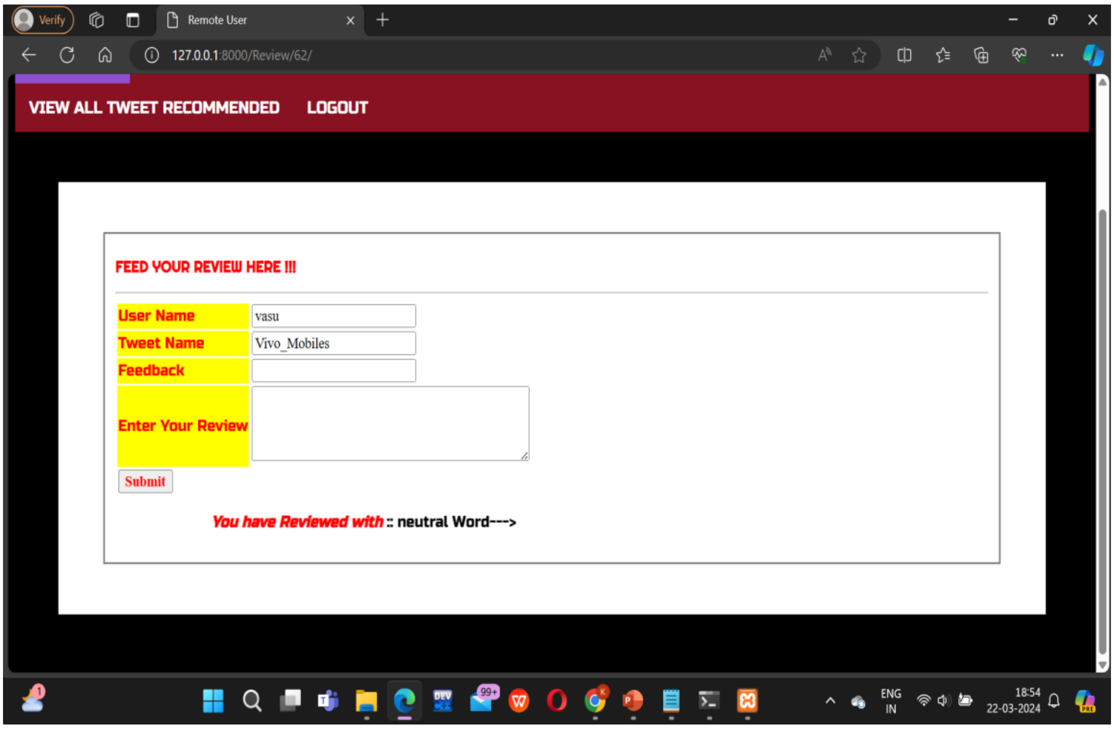
Screenshot 6.1 : Registration Page



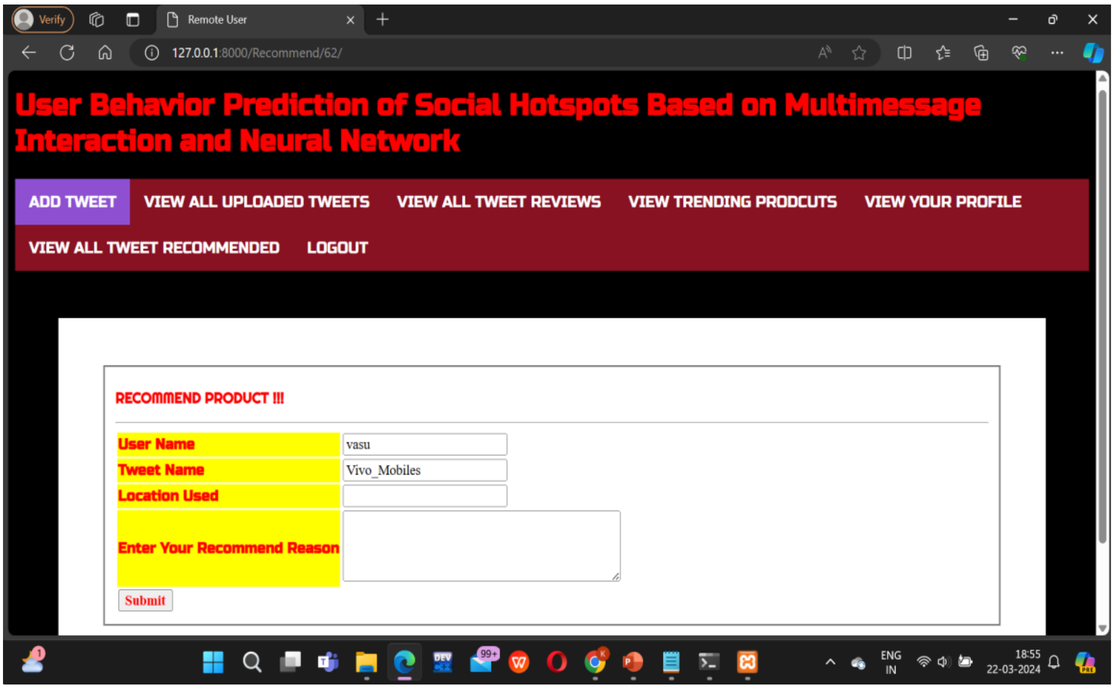
Screenshot 6.2 : Add Tweet



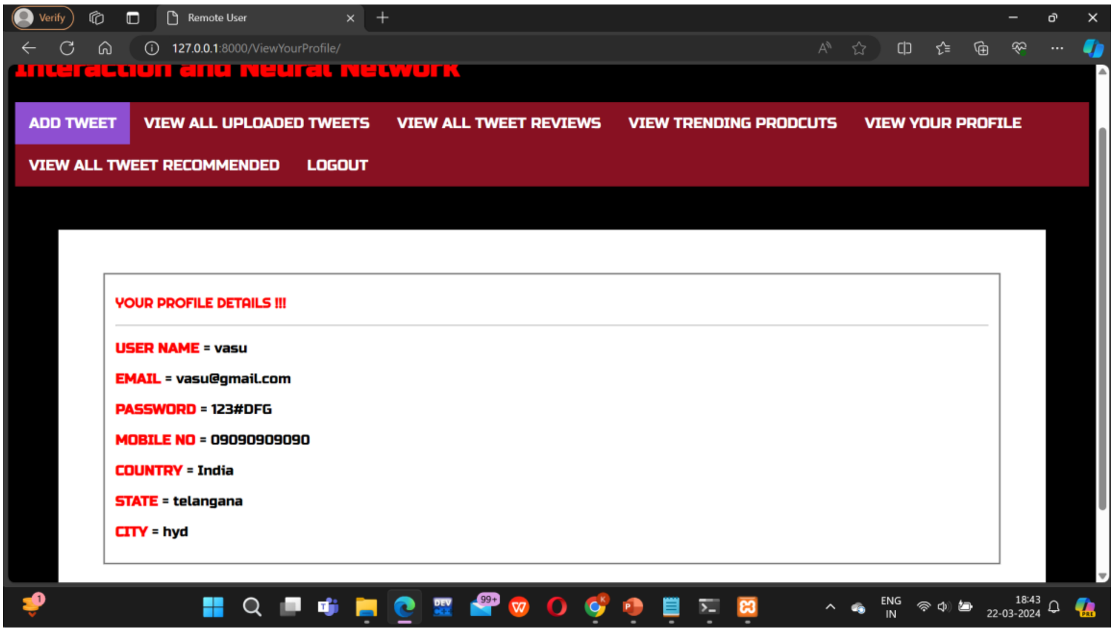
Screenshot 6.3 : View all uploaded tweet



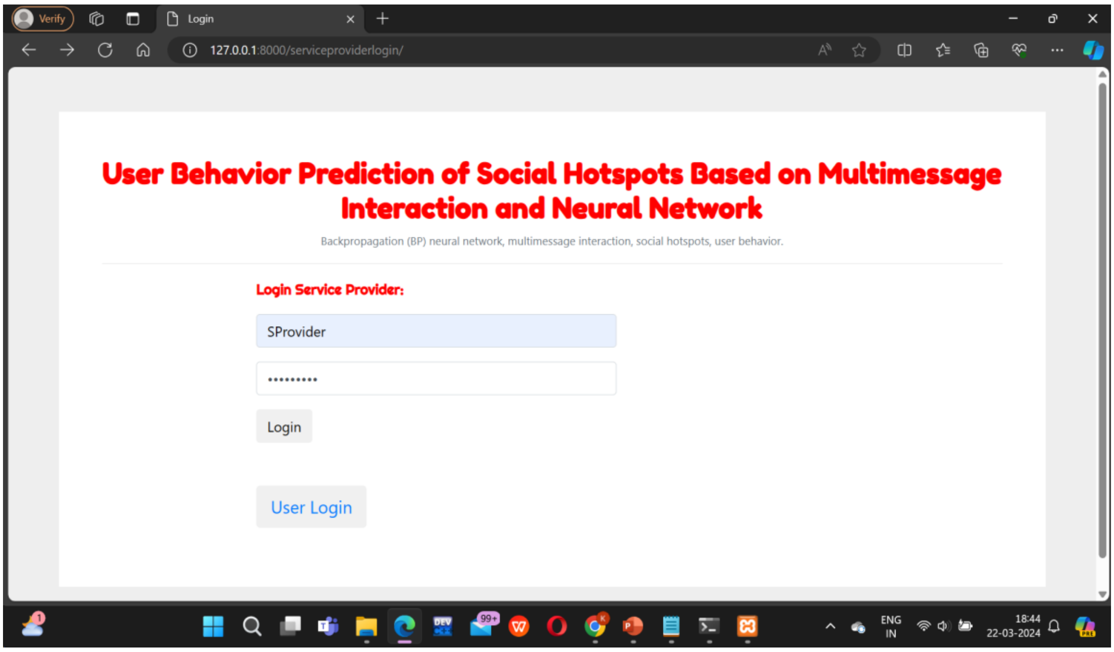
Screenshot 6.4 : Review Product



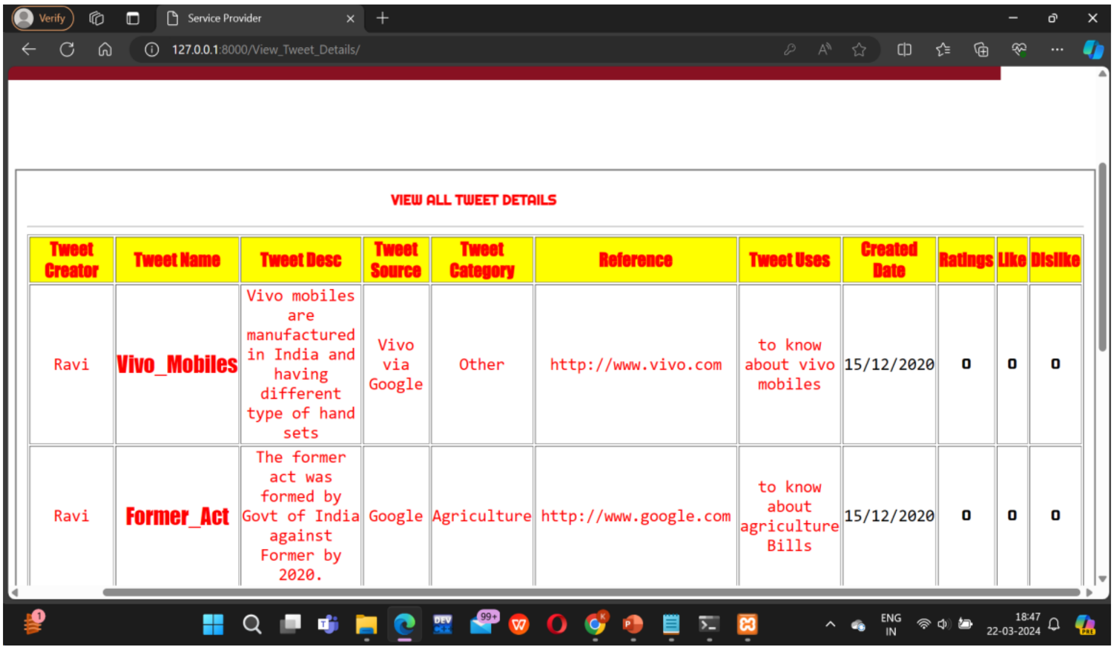
Screenshot 6.5 : Recommend Product



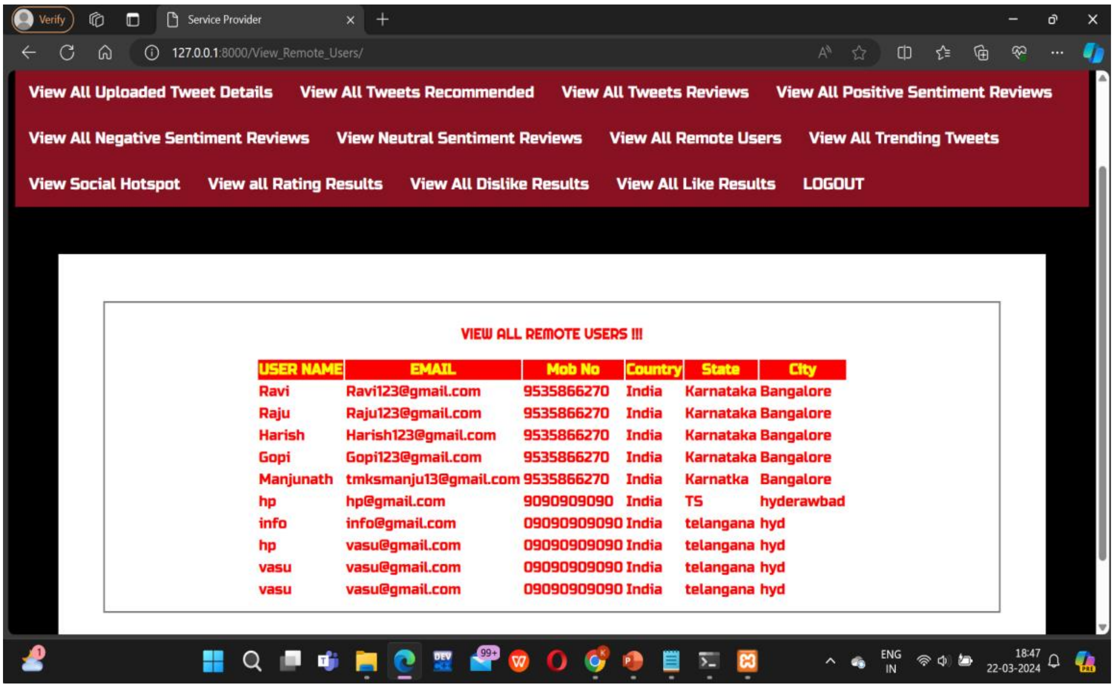
Screenshot 6.6 : Profile Details



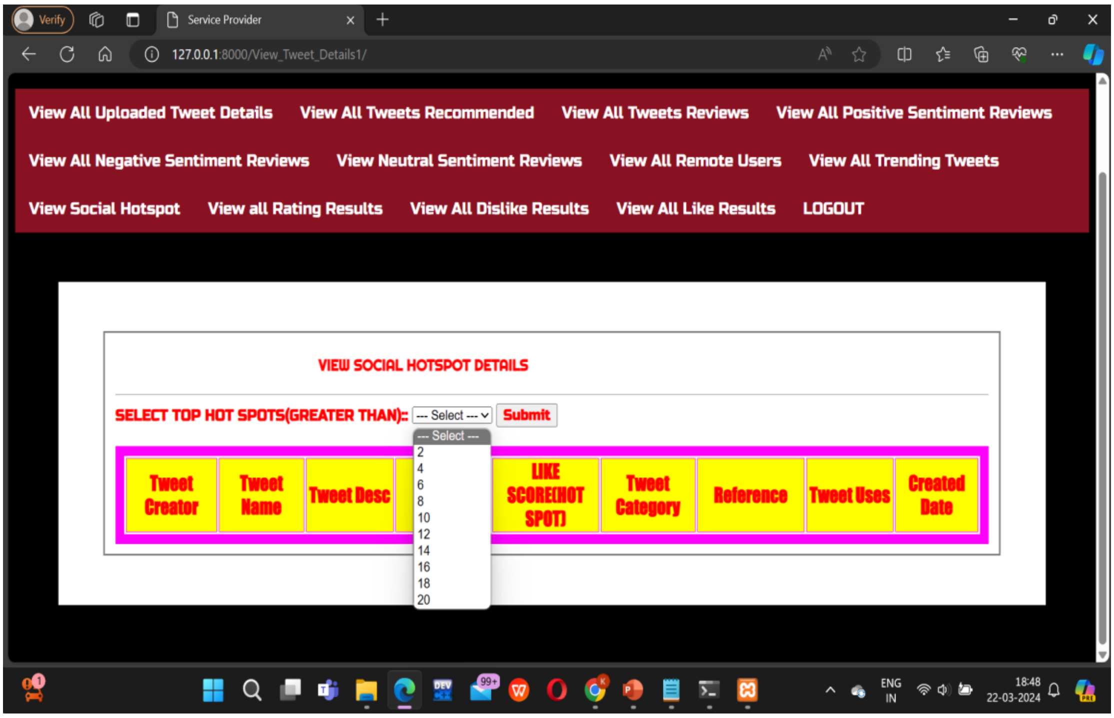
Screenshot 6.7 : Service Provider Login



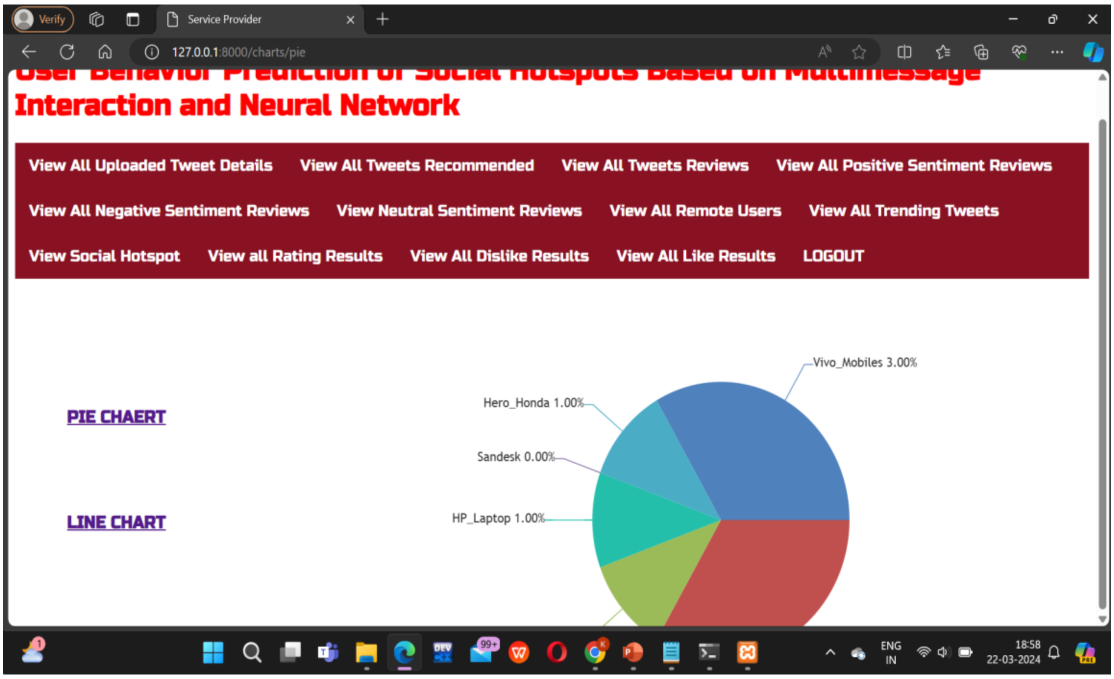
Screenshot 6.8 : View All Tweet Details



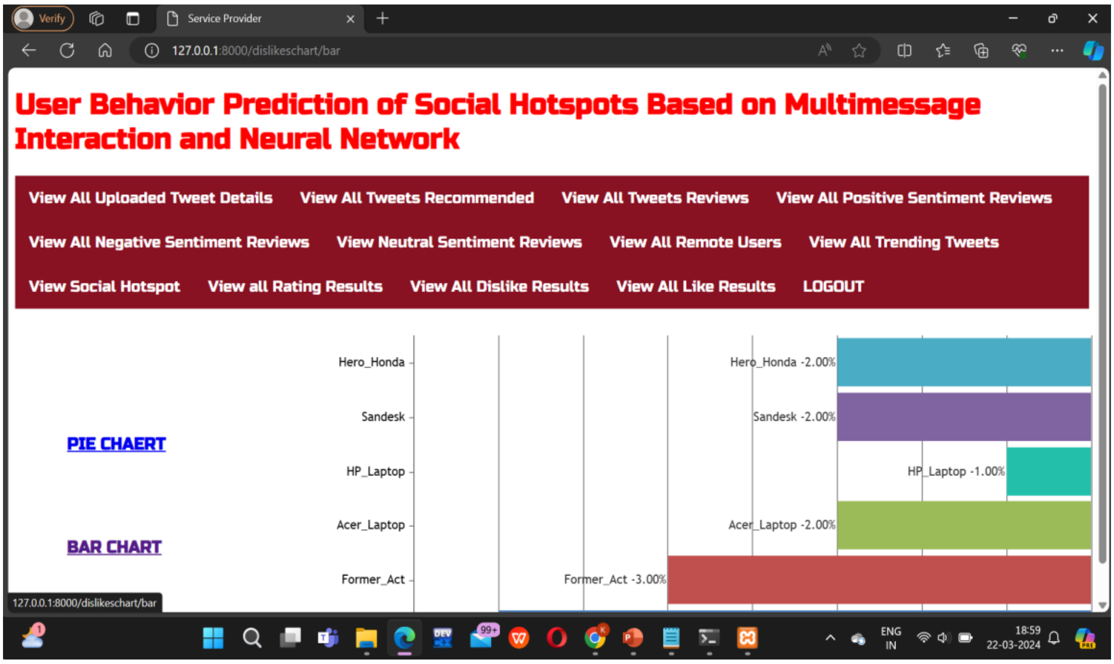
Screenshot 6.9 : View All Remote Users



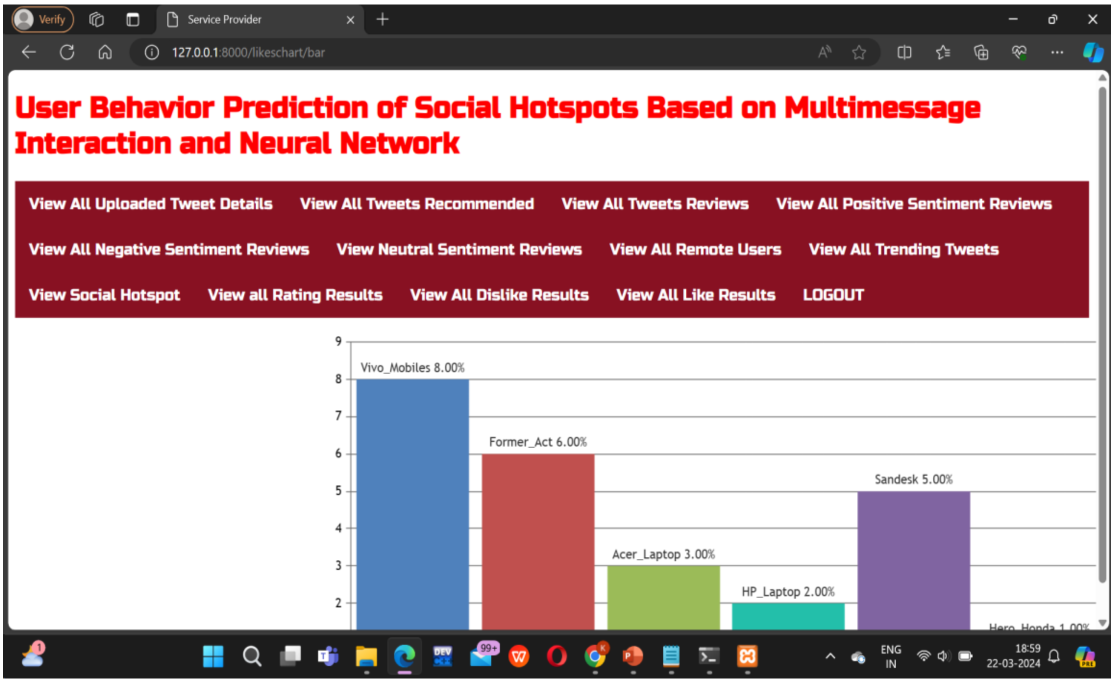
Screenshot 6.10 : View Social Hotspot Details



Screenshot 6.11 : View All Rating Results



Screenshot 6.12 : View All Dislike Results



Screenshot 6.13 : View All Like Results

7. TESTING

7. TESTING

7.1 INTRODUCTION TO 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, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.2 TYPES OF TESTING

7.2.1 UNIT 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 a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, 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.

7.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

7.3 TEST CASES

7.3.1 CLASSIFICATION

Test Case ID	Test Case Name	Purpose	Input	Output
1	User Register	User gets Registered	The user gives the input in the form of data	The data gets registered in dataset
2	User Login	User gets login	The user provides the login details	Successfully logged in
3	View All Uploaded tweets	Need to get no of likes, dislikes,review	Gives the likes, dislikes,review for each product	Successfully directed to graph in server page
4	View all sentiments on reviews	Go through positive, negative , neutral reviews	Depends on selection of either of three	Successfully gives the output depends on selection
5	View like & dislike result	Analysis of tweets made by person on post	Go through the dataset	Like gives piechart& dislike gives Barchart
6	View all Rating Results	Ratings recommended for product	With the dataset	Best product for rating

8. CONCLUSION

8. CONCLUSION & FUTURE SCOPE

8.1 PROJECT CONCLUSION

The User Behaviour Prediction System in Social Hotspots based on Multi-Message Interaction and Neural Network represents a comprehensive and innovative approach to understanding and forecasting user behaviours within dynamic online environments. The integration of diverse message types and the utilization of advanced neural network architectures contribute to the system's ability to capture the intricate dynamics of social interactions. The adoption of a Back Propagation (BP) neural network, designed to handle the behavioural complexity of users in multi-message hotspots, demonstrates a commitment to modelling the nonlinear relationships inherent in social interactions. The proposed neural network not only captures complex patterns but is also optimized using a simulated annealing algorithm to mitigate overfitting issues.

8.2 FUTURE SCOPE

User Behaviour Prediction in Social Hotspots using Multi-Message Interaction and Neural Networks holds vast potential for advancements across several key dimensions. Research endeavours may focus on refining model architectures to accommodate emerging forms of social media content and complex interaction patterns, while also incorporating interdisciplinary insights to enhance the models' interpretability and ethical considerations. Further exploration into adaptive and real-time prediction techniques, as well as robustness against adversarial attacks, can lead to more effective and reliable predictive systems. Additionally, efforts to scale models efficiently, support multilingual interactions, and optimize user engagement through personalized interventions will be crucial for addressing the evolving landscape of online social dynamics. Furthermore, exploring innovative approaches such as hybrid modelling, incremental learning, and dynamic community detection promises to unlock new avenues for understanding and predicting user behaviour in diverse social hotspot environments.

9. BIBLIOGRAPHY

9. BIBLIOGRAPHY

9.1 REFERENCES

- M. Takayasu, K. Sato, Y. Sano, K. Yamada, W. Miura and H. Takayasu, "Rumor diffusion and convergence during the 3.11 earthquake: A Twitter case study", *PLoS ONE*, vol. 10, no. 4, Apr. 2015
- S. Gaglio, G. Lo Re and M. Morana, "A framework for real-time Twitter data analysis", *Comput. Commun.*, vol. 73, pp. 236-242, Jan. 2016.
- A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel and B. Bhattacharjee, "Measurement and analysis of online social networks", *Proc. 7th ACM SIGCOMM Conf. Internet Meas.*, pp. 29-42, 2007.
- J. Golbeck, C. Robles, M. Edmondson and K. Turner, "Predicting personality from twitter", *Proc. IEEE 3rd Int. Conf. Privacy Secur. Risk Trust IEEE 3rd Int. Conf. Social Comput.*, pp. 149-156, Oct. 2011.
- Li, X., Liu, Y., & Zhang, L. (2019). User behaviour prediction in social media using multi-task neural networks. In 2019 IEEE 6th International Conference on Data Science and Advanced Analytics (DSAA) (pp. 536-545). IEEE.
- Chen, X., Zhang, T., Xu, C., & Tang, J. (2020). Social hotspot detection through deep neural networks. Information Sciences, 512, 268-279.
- Wu, Y., DuBois, T., & Chang, Y. (2018). Modeling multi-message interactions with neural network for hot topic prediction in online forums. In Proceedings of the 27th International Conference on Computational Linguistics (pp. 3163-3173).
- Wang, Y., & Ye, C. (2021). Deep learning-based prediction model for user behavior in social networks. Journal of Computer Science and Technology, 36(5), 1203-1218.

9.2 GITHUB LINK

https://github.com/207r1a05p6/major-project