A

Major Project

On

PROBABILISTIC INFERENCE AND TRUSTWORTHINESS EVALUATION OF ASSOCIATIVE LINKS TOWARDS MALICIOUS ATTACK DETECTION FOR ONLINE RECOMMENDATIONS

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

S. DIVYA (207R1A05B4)

B. LAHARI (207R1A0567)

M. SRI HARSHA (207R1A0592)

Under the Guidance of

TABEEN FATIMA

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGCAUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE,
NewDelhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V),
Medchal Road, Hyderabad-501401. **2020-2024**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **“PROBABILISTIC INFERENCE AND TRUSTWORTHINESS EVALUATION OF ASSOCIATIVE LINKS TOWARDS MALICIOUS ATTACK DETECTION FOR ONLINE RECOMMENDATIONS”** being submitted by **S. DIVYA (207R1A05B4), B. LAHARI (207R1A0567) and M. SRI HARSHA (207R1A0592)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of Bonafide work carried out by them under our guidance and supervision during the year 2023-24.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

TABEEN FATIMA
(Assistant Professor)
INTERNAL GUIDE

Dr. A. RAJI REDDY
DIRECTOR

Dr. K. SRUJAN RAJU
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ______________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express our profound gratitude and deep regard to our guide **Ms. Tabeen Fatima,** Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. J. Narasimha Rao, Mr. G. Vinesh Shanker, Ms. Shilpa & Dr. K. Maheshwari** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

S. DIVYA	**(207R1A05B4)**
B. LAHARI	**(207R1A0567)**
M. SRI HARSHA	**(207R1A0592)**

ABSTRACT

The increasing use of recommender systems as personalization recommendation services such as Amazon, Trip Advisor, and Yelp, has stressed the demand for secure and usable abnormality detection techniques, due to fundamental vulnerabilities of recommender systems and their openness. With the emergence of new attacks, how to defend diverse malicious attacks for online recommendations is a challenging issue. Moreover, characterizing and evaluating sparse rating behaviors are a long-standing problem that still remains open, leading to an upsurge of research, as well as real application.

This paper investigates probabilistic inference and trustworthiness evaluation of behavioral links according to coupled association networks converted from rating behaviors, and presents a unified detection framework from a novel perspective to spot diverse malicious threats. Firstly, an association graph is constructed from the original rating matrix based on both the inherent rating motivation of users and atomic propagation rules of coupled networks. Then, we evaluate the trustworthiness of link behaviors in the targeted network of coupled association network by exploiting a factor graph model of coupled network, and re-determine concerned links in the targeted network.

Finally, suspicious users and items can be empirically inferred by comprehensively evaluating the trustworthiness of both links and nodes in the targeted network. Extensive experiments on synthetic data for profile injection attacks and co-visitation injection attacks, as well as real-world data including Amazon and Trip Advisor, demonstrate the effectiveness of the proposed detection approach compared with competing benchmarks.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

Recommender systems play an increasing role in e- commerce systems such as Amazon, eBay, etc. In reality, they are susceptible to malicious attacks including profile injection attacks and co-visitation injection attacks. Defending these threats has attracted much attention from both academic and industry in the past two decades. Promising results can be obtained from previous efforts, which can be briefly summarized in the following aspects:

- Characterizing rating behaviors between users.
- Exploring the elimination of disturbed data and divide-and-conquer detection strategies.
- Developing detection approaches for real-world application. In what follows, we introduce related researches focused on the above aspects.

The project involves employing probabilistic inference to assess the trustworthiness of association links in online recommendations, aiming to detect potential malicious attacks. This would likely include developing models that evaluate the likelihood of associations being legitimate or malicious, enhancing the security of online recommendation systems.

1.2 PROJECT PURPOSE

The purpose of the project, "Probabilistic Inference and Trustworthiness Evaluation of Associative Links toward Malicious Attack Detection for Online Recommendations, "is to enhance the security, reliability, and trustworthiness of online recommendation systems.

By achieving these purposes, the project aims to create a more secure, trustworthy, and user-friendly online recommendation system that not only provides accurate suggestions but also protects users from potential malicious activities, ultimately enhancing the overall quality of the user experience.

By fulfilling these purposes, the project aims to create a state-of-the-art online recommendation system that not only provides accurate and valuable suggestions to users but also proactively safeguards against malicious activities, ultimately contributing to the advancement of secure and trustworthy online platforms.

1.3 PROJECT FEATURES

Associative Link Analysis: Develop algorithms to analyze and understand the relationships between items, users, and sources of recommendations. Consider factors such as the frequency of links, the diversity of sources, and the historical behavior of users.

Probabilistic Inference Model: Implement a probabilistic model that calculates the likelihood of an associative link being part of a malicious attack. Train the model using historical data on both legitimate and malicious link behaviors.

Real-time Monitoring: Enable real-time monitoring of online recommendations to promptly detect and respond to potential malicious activities. Implement mechanisms for continuous learning to adapt the probabilistic model over time.

Trustworthiness Evaluation Criteria: Define and integrate criteria for evaluating the trustworthiness of associative links. This might include user feedback, content quality, and reputation of the source.

User Behavior Analysis: Incorporate user behavior analysis to identify anomalies or deviations from normal patterns. Unusual user interactions could be indicative of malicious activities.

Explainability and Interpretability: Ensure that the probabilistic inference model is explainable and interpretable. This is crucial for gaining user trust and understanding the decision-making process of the system.

Privacy Protection: Design the system with privacy in mind. Consider techniques such as differential privacy to protect user data while still obtaining valuable insights for detection.

2. LITERATURE SURVEY

1. LITERATURE SURVEY

Probabilistic inference techniques are pivotal in modelling the complex relationships inherent in online recommendation systems. One prominent approach involves the use of Bayesian networks, which excel at capturing dependencies between different entities such as users, items, and contextual data. Research in this area focuses on leveraging Bayesian networks to infer the likelihood of malicious associations within recommendation systems. Additionally, probabilistic graphical models like Markov networks and factor graphs are explored for their ability to represent probabilistic dependencies and make inferences about potential malicious activities.

Trustworthiness evaluation methods play a crucial role in assessing the reliability and credibility of associative links in online recommendations. Reputation-based systems are a common avenue of exploration, where reputation scores or trust metrics are computed and utilized to evaluate the trustworthiness of users, items, and interactions. Anomaly detection techniques are also studied extensively, aiming to detect anomalous behavior or patterns that could indicate malicious activities, such as fake profiles or orchestrated attacks aimed at manipulating recommendations.

In the realm of malicious attack detection, researchers delve into various approaches to identify and mitigate potential threats to online recommendation systems. Adversarial modelling is employed to simulate attacks, such as profile injection attacks or shilling attacks, in order to develop robust defense mechanisms. Machine learning-based detection methods, including anomaly detection algorithms and supervised learning models, are investigated for their efficacy in recognizing suspicious patterns or behaviors.

Furthermore, the security of online recommendation systems is a paramount concern, leading to the exploration of privacy-preserving techniques and robustness analysis methodologies. Privacy-preserving techniques like differential privacy and homomorphic encryption are studied to protect user data while maintaining recommendation accuracy. Robustness analysis involves evaluating recommendation algorithms' resilience against various types of attacks, ensuring system integrity and user trust. Additionally, trust-aware recommendation algorithms are proposed, integrating trust-related features or constraints to enhance system security and mitigate potential vulnerabilities.

Case studies and experimental evaluations provide valuable insights into the real-world applicability and performance of probabilistic inference and trustworthiness evaluation techniques. These studies assess the effectiveness, scalability, and computational complexity of different approaches in detecting and mitigating malicious attacks. Moreover, future directions and challenges in the field are discussed, highlighting emerging trends, open research questions, and potential avenues for enhancing the security and trustworthiness of online recommendation systems.

3. SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

System Components: Identify and define the major components of your system, such as the probabilistic inference module, the trustworthiness evaluation module, real-time monitoring, user feedback mechanisms, and the adaptive learning system.

Data Flow: Map out the flow of data through the system. How is data collected from online recommendations? What preprocessing steps are involved before it reaches the probabilistic inference and trustworthiness evaluation modules?

Data Sources: Specify the sources of data, including user interactions, recommendation sources, and any additional contextual information. Consider how these sources contribute to the overall understanding of link trustworthiness.

Probabilistic Inference Model: Provide a detailed analysis of the probabilistic inference model. What algorithms or statistical methods are being employed? How is the model trained and updated over time? What features are considered in the inference process?

Trustworthiness Evaluation Criteria: Define the criteria used to evaluate the trustworthiness of associative links. This could involve a combination of user feedback, source reputation, content quality, and other relevant factors.

3.1 PROBLEM DEFINITION

The project involves developing a system for detecting malicious attacks in online recommendations by employing probabilistic inference and evaluating the trustworthiness of association links. This includes assessing the reliability of connections between entities to enhance the accuracy of identifying potential threats in the recommendation system.

The problem addressed in this project is the vulnerability of online recommendation systems to malicious attacks, where adversaries manipulate association links to deceive the system. The challenge lies in implementing probabilistic inference techniques to model the uncertainty in link associations and developing a trustworthiness evaluation mechanism. This is crucial for detecting and mitigating malicious activities, ensuring the integrity and security of online recommendations.

The goal of detection is, given a rating dataset D which is generally combined with fake injection data, to discover anomalous information. Facing with the imbalanced distribution between authentic and abnormal data, G is empirically pruned from the perspective of removing disturbed information. Based on a well-pruned graph, the association of links and the trustworthiness of nodes are probabilistically evaluated in order to further shrink the scope of detection. It is noteworthy that we exploit the trustworthiness of both nodes (users or items) and links to filter out disturbed nodes and untrusted links. Concretely, the spam probability of each user and item based on the transition probability is used to evaluate the trustworthiness of nodes in G and G0. Similarly, the trustworthiness of links in GI0 is re-determined according to the link prediction of coupled networks in order to further prune the association between nodes. In other words, how to eliminate untrusted links in GI0 needs to be focused. Formally, the ultimate task is to intelligently prune G0\Gf as well as keep Gf unchanged as far as possible.

3.2 EXISTING SYSTEM

With the development of machine learning and online social network, researchers have begun to investigate detection models from the perspectives of supervised, unsupervised and semi-supervised learning. Naturally, designing representative rating features extracted from the original rating data is a primary task. To this end, on the one hand, Chirita et al developed statistical metrics to reveal rating patterns of shilling attackers. Then, Burke et al analyzed diverse features such as generic features and model specific features to detect shilling attackers.

On the other hand, graph-based detection methods for spotting fake accounts or Sybil nodes have been well developed. Investigating attributes of link behaviors in association graphs has also attracted much attention Cao et al. graph properties to rank users according to their perceived likelihood of being fake. Then, Gong et al. proposed a semi-supervised learning framework to detect Sybil nodes.

They also investigated algorithms for both link prediction and attribute inference. Additionally, Dong et al. presented a unified link prediction framework (termed CLP) to deal with the problem of heterogeneous interactions in coupled networks. In particular, they mathematically analyzed the feasibility for the inference of associative links in coupled networks.

Link prediction in the target network, however, depends heavily on the structure of both the source and cross networks. In this sense, improving the performance of link prediction may be easily constrained by disturbed information (e.g., an inactive user rated an unpopular item in the user-item cross network). After that, Wang et al. investigated a guilt-by-association method to detect fraudulent users in online social networks.

3.2.1 DISADVANTAGES OF EXISTING SYSTEM

Following are the disadvantages of existing system:

- Refinement of association graphs is not implemented for Determining target items (or anchor items) is a crucial task for injection attack detection.

- Determination of suspicious nodes are not implemented Based on the re determined links in target network and refined cross network.

3.3 PROPOSED SYSTEM

- We develop a unified representation of injection behaviors to deal with different injection attacks, and propose to combine the iteration propagation of behavior and atomic propagation rules of coupled networks for both the trustworthiness evaluation of nodes and the determination of disturbed information.
- We propose to incorporate a graphical representation of coupled networks for probabilistic inference of association links, and propose to globally refine association links using both the coupled factor graph model and the recursive propagation of transition probability for determining target items and anchor items.
- We propose a unified detection approach to identify both profile injection attacks and co- visitation injection attacks. Additionally, discovering spam users on Trip Advisor and Amazon data is also investigated and analyzed. Extensive experiments on synthetic data and real-world data demonstrate the effectiveness of the proposed detection approach.

3.3.1 ADVANTAGES OFTHE PROPOSED SYSTEM

- Unified representation of injection behaviors: Different injection attacks have different raw data. Concretely, profile injection attacks in collaborative recommendations are experimentally conducted on real data (e.g., Movie Lens-100).

- Construction of association links: Based on the weighted graphs, we evaluate propagation behaviors between nodes (users and items) in order to construct associative links.

3.4 FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase and 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, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

3.4.1 ECONOMICAL FEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies use dare freely available. Only the customized products had to be purchased.

3.4.2 TECHNICAL FEASIBILITY:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.4.3 SOCIAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.5 HARDWARE & SOFTWARE REQUIREMENTS

3.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- **Processor** : Pentium –IV
- **RAM** : 4 GB (min)
- **Hard Disk** : 20 GB
- **Key Board** : Standard Windows Keyboard
- **Mouse** : Two or Three Button Mouse
- **Monitor** : SVGA

3.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements.

3.5.2.1 **Operating System :** Windows 7 Ultimate.

3.5.2.2 **Coding Language :** Python.

3.5.2.3 **Front-End :** Python.

3.5.2.4 **Back-End :** Django-ORM

3.5.2.5 **Designing :** Html, CSS, JavaScript

3.5.2.6 **Data Base :** MySQL (WAMP Server).

4. ARCHITECTURE

4.ARCHITECTURE

4.1 PROJECT ARCHITECTURE

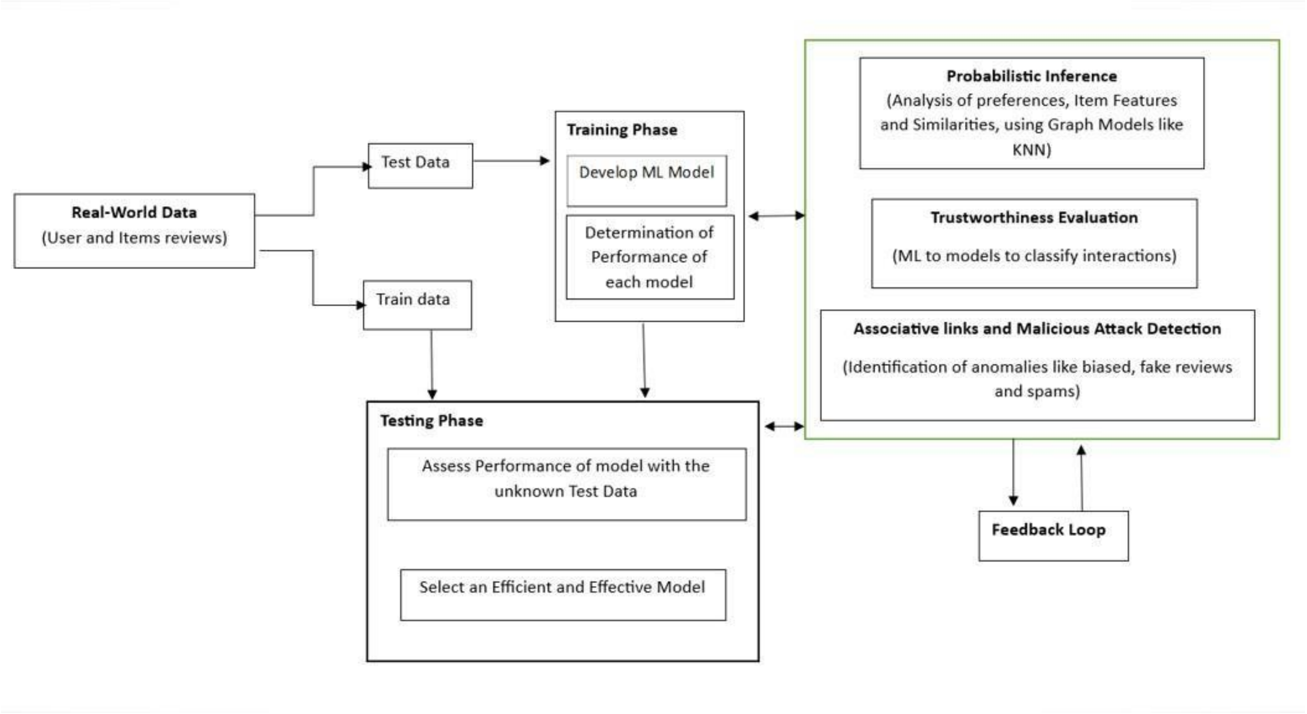

Real-World Data
(User and Items reviews)
Test Data
Train data
Training Phase
Develop ML Model
Determination of Performance of each model
Testing Phase
Assess Performance of model with the unknown Test Data
Select an Efficient and Effective Model
Probabilistic Inference
(Analysis of preferences, Item Features and Similarities, using Graph Models like KNN)
Trustworthiness Evaluation
(ML to models to classify interactions)
Associative links and Malicious Attack Detection
(Identification of anomalies like biased, fake reviews and spams)
Feedback Loop

Figure 1: Architecture

4.2 USECASE 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.

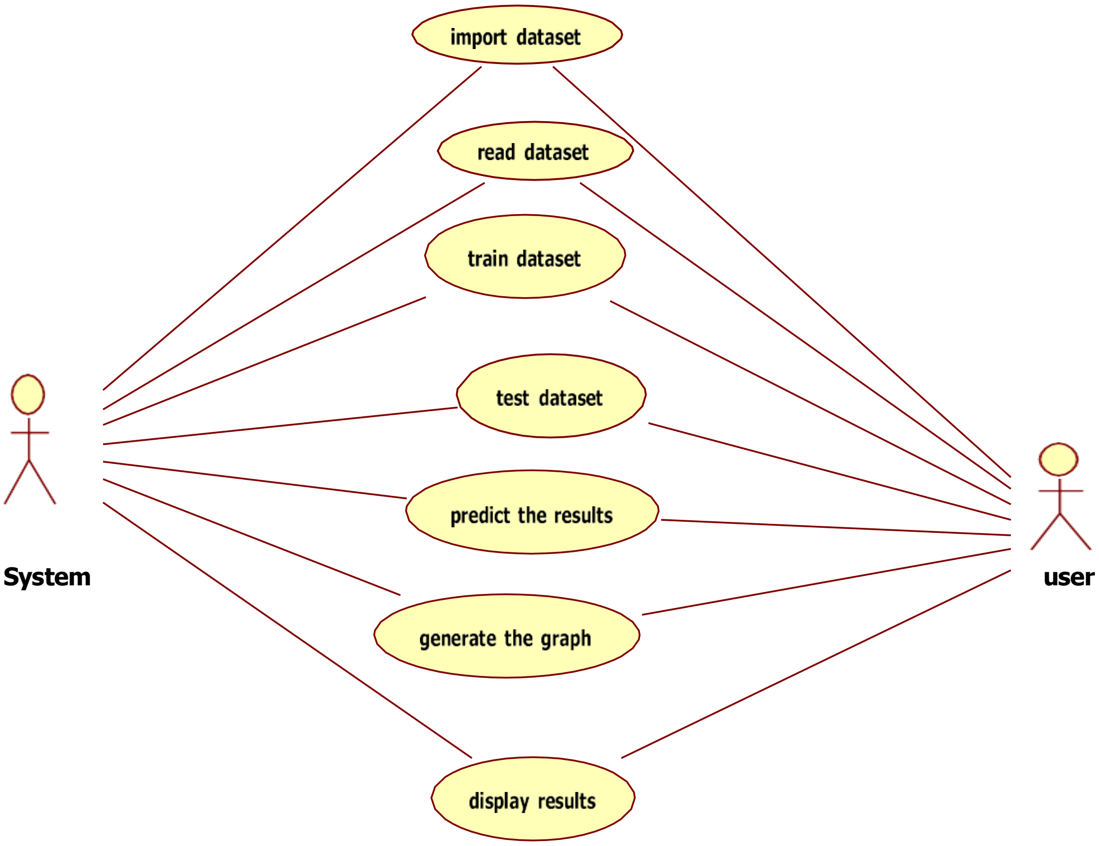


Figure 2: Use Case Diagram

4.3 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML)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 the classes. It explains which class contains information.

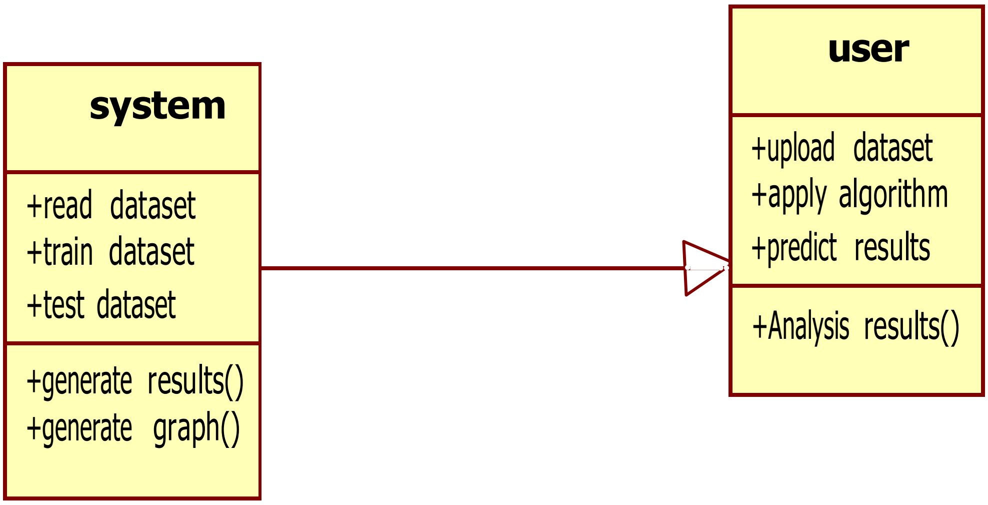


Figure 3: Class Diagram

4.4 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 a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

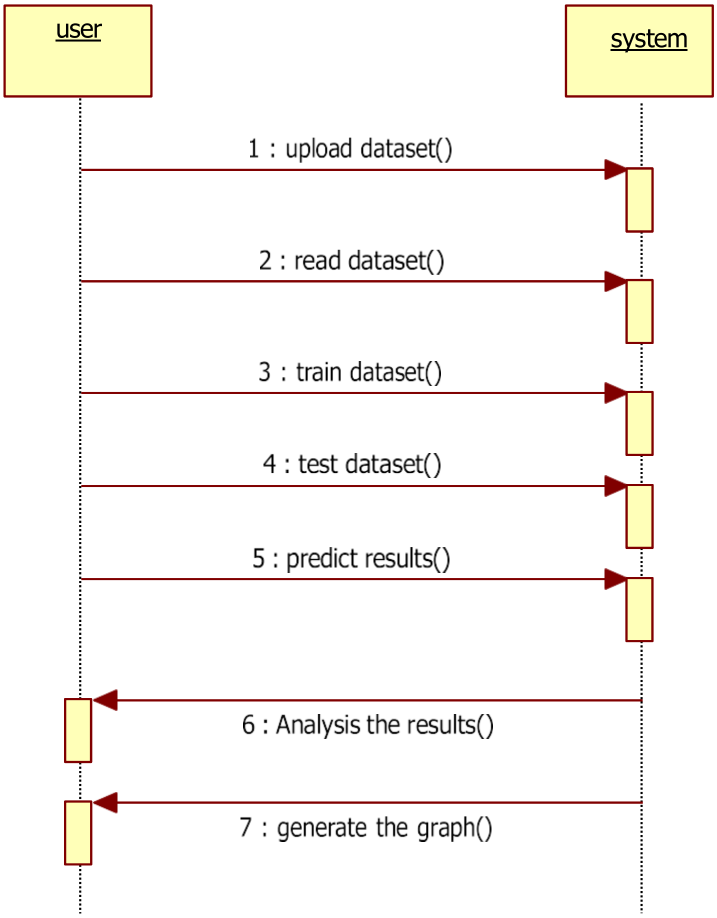


Figure 4: Sequence Diagram

4.5 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.

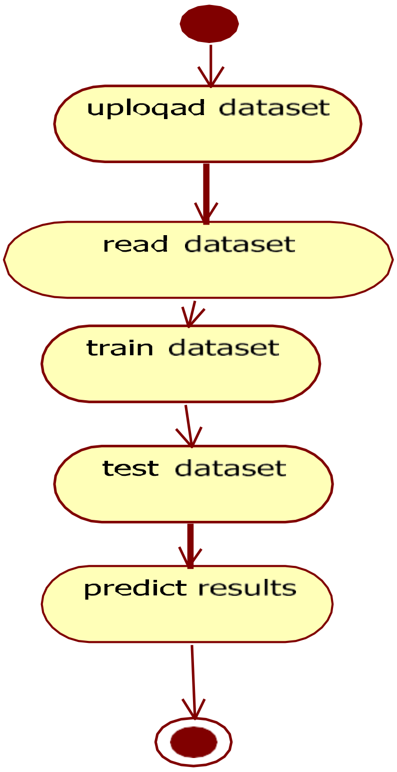


Figure 5: Activity Diagram

5. IMPLEMENTATION

5 IMPLEMENTATION

5.1 SAMPLE CODE

```
from django.db.models import Count, Avg
from django.shortcuts import render, redirect
from django.db.models import Count
from django.db.models import Q
import datetime
import xlwt
from django.http import HttpResponse


import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier

# Create your views here.
from Remote_User.models import
ClientRegister_Model,detect_trust_type,detection_ratio,detection_accuracy


def serviceproviderlogin(request):
    if request.method  == "POST":
        admin = request.POST.get('username')
        password = request.POST.get('password')
        if admin == "Admin" and password =="Admin":
            detection_accuracy.objects.all().delete()
            return redirect('View_Remote_Users')

    return render(request,'SProvider/serviceproviderlogin.html')

def View_Prediction_Of_Product_Review_Trust_Ratio(request):
    detection_ratio.objects.all().delete()
    ratio = ""
    kword = 'No Trust'
    print(kword)
    obj = detect_trust_type.objects.all().filter(Q(Prediction=kword))
    obj1 = detect_trust_type.objects.all()
    count = obj.count();
    count1 = obj1.count();
    ratio = (count / count1) * 100
    if ratio != 0:
        detection_ratio.objects.create(names=kword,  ratio=ratio)

    ratio12 = ""
    kword12 = 'Average Trust'
    print(kword12)
    obj12 = detect_trust_type.objects.all().filter(Q(Prediction=kword12))
```

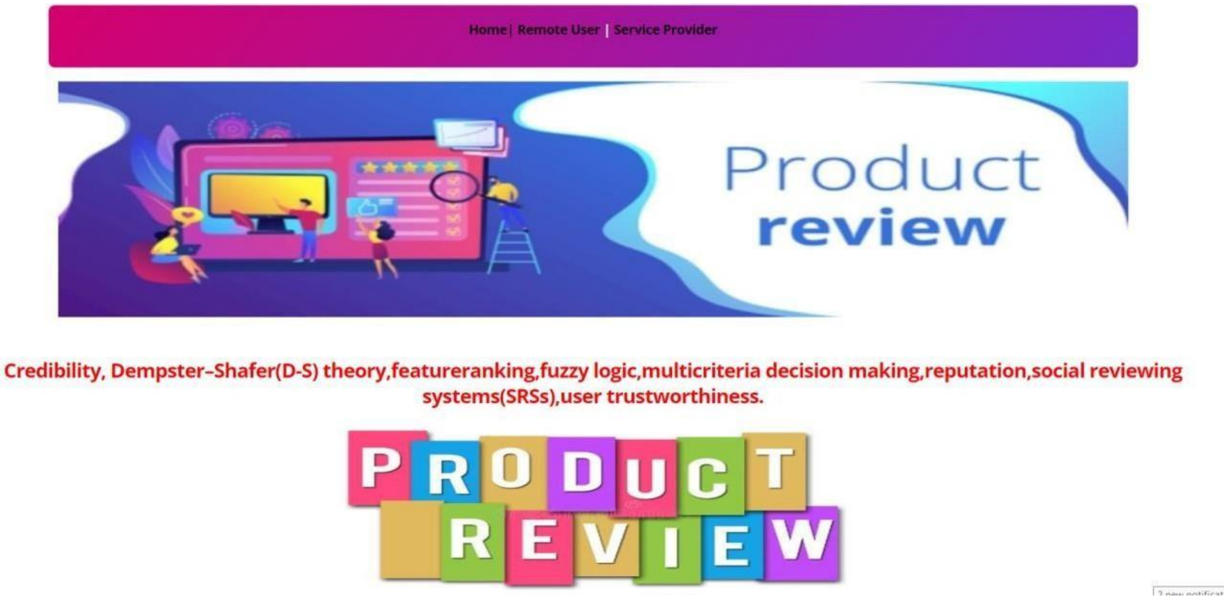
```
   obj112 = detect_trust_type.objects.all()
    count12 = obj12.count();
    count112 = obj112.count();
    ratio12 = (count12 / count112) * 100
    if ratio12 != 0:
        detection_ratio.objects.create(names=kword12,  ratio=ratio12)

    ratio12 = ""
    kword12 = 'Good Trust'
    print(kword12)
    obj12 = detect_trust_type.objects.all().filter(Q(Prediction=kword12))
    obj112 = detect_trust_type.objects.all()
    count12 = obj12.count();
    count112 = obj112.count();
    ratio12 = (count12 / count112) * 100
    if ratio12 != 0:
        detection_ratio.objects.create(names=kword12,  ratio=ratio12)

    obj = detection_ratio.objects.all()
    return render(request,
'SProvider/View_Prediction_Of_Product_Review_Trust_Ratio.html',  {'objs':  obj})

def View_Remote_Users(request):
    obj=ClientRegister_Model.objects.all()
    return render(request,'SProvider/View_Remote_Users.html',{'objects':obj})

def charts(request,chart_type):
    chart1 = detection_ratio.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts.html", {'form':chart1,
'chart_type':chart_type})

def charts1(request,chart_type):
    chart1 = detection_accuracy.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts1.html", {'form':chart1,
'chart_type':chart_type})

def View_Prediction_Of_Product_Review_Trust(request):
    obj =detect_trust_type.objects.all()
    return render(request, 'SProvider/View_Prediction_Of_Product_Review_Trust.html',
{'list_objects': obj})

def likeschart(request,like_chart):
    charts =detection_accuracy.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/likeschart.html", {'form':charts,
'like_chart':like_chart})

def Download_Predicted_DataSets(request):

    response = HttpResponse(content_type='application/ms-excel')
    # decide file name
    response['Content-Disposition'] = 'attachment; filename="Predicted_Datasets.xls"'
    # creating workbook
    wb = xlwt.Workbook(encoding='utf-8')

  # adding sheet
    ws = wb.add_sheet("sheet1")
    # Sheet header, first row
    row_num = 0
    font_style = xlwt.XFStyle()
    # headers are bold
    font_style.font.bold = True
    # writer = csv.writer(response)
    obj = detect_trust_type.objects.all()
    data = obj # dummy method to fetch data.
    for my_row in data:
        row_num = row_num + 1

        ws.write(row_num, 0, my_row.reviewerID, font_style)
        ws.write(row_num, 1, my_row.Product_Id, font_style)
        ws.write(row_num, 2, my_row.reviewerName, font_style)
        ws.write(row_num, 3, my_row.helpful, font_style)
        ws.write(row_num, 4, my_row.reviewText, font_style)
        ws.write(row_num, 5, my_row.overall_rating, font_style)
        ws.write(row_num, 6, my_row.summary, font_style)
        ws.write(row_num, 7, my_row.unixReviewTime, font_style)
        ws.write(row_num, 8, my_row.reviewTime, font_style)
        ws.write(row_num, 9, my_row.day_diff, font_style)
        ws.write(row_num, 10, my_row.helpful_yes, font_style)
        ws.write(row_num, 11, my_row.total_vote, font_style)
        ws.write(row_num, 12, my_row.Prediction, font_style)

    wb.save(response)
    return response

def train_model(request):
    detection_accuracy.objects.all().delete()

    df = pd.read_csv('Product_Reviews.csv')

    def apply_response(total_vote):
        if (total_vote == 0):
            return 0  # No Trust
        elif (total_vote>0 and total_vote<50):
            return 1  # Average Trust
        elif(total_vote>50):
            return 2  # Good Trust

    df['Results'] = df['total_vote'].apply(apply_response)

    cv = CountVectorizer()
    X = df['reviewText'].apply(str)
    y = df['Results']
```

```
print("Review")

print(X)
print("Results")
print(y)

X = cv.fit_transform(X)

models = []
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
X_train.shape, X_test.shape, y_train.shape


print(X_test)

print("Naive Bayes")

from sklearn.naive_bayes import MultinomialNB
NB = MultinomialNB()
NB.fit(X_train, y_train)
predict_nb = NB.predict(X_test)
naivebayes = accuracy_score(y_test, predict_nb) * 100
print(naivebayes)
print(confusion_matrix(y_test, predict_nb))
print(classification_report(y_test, predict_nb))
models.append(('naive_bayes', NB))
detection_accuracy.objects.create(names="Naive Bayes", ratio=naivebayes)

# SVM Model
print("SVM")
from sklearn import svm
lin_clf = svm.LinearSVC()
lin_clf.fit(X_train, y_train)
predict_svm = lin_clf.predict(X_test)
svm_acc = accuracy_score(y_test, predict_svm) * 100
print(svm_acc)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, predict_svm))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, predict_svm))
models.append(('svm', lin_clf))
detection_accuracy.objects.create(names="SVM", ratio=svm_acc)

print("Logistic Regression")

from sklearn.linear_model import LogisticRegression
reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train, y_train)
y_pred = reg.predict(X_test)
print("ACCURACY")
```

```
    print(accuracy_score(y_test, y_pred) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, y_pred))

    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, y_pred))
    models.append(('logistic', reg))
    detection_accuracy.objects.create(names="Logistic Regression",
ratio=accuracy_score(y_test, y_pred) * 100)
    print("Decision Tree Classifier")
    dtc = DecisionTreeClassifier()
    dtc.fit(X_train, y_train)
    dtcpredict = dtc.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, dtcpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, dtcpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, dtcpredict))
    models.append(('DecisionTreeClassifier', dtc))
    detection_accuracy.objects.create(names="Decision Tree Classifier",
ratio=accuracy_score(y_test, dtcpredict) * 100)

    print("KNeighborsClassifier")
    from sklearn.neighbors import KNeighborsClassifier
    kn = KNeighborsClassifier()
    kn.fit(X_train, y_train)
    knpredict = kn.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, knpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, knpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, knpredict))
    models.append(('KNeighborsClassifier', kn))
    detection_accuracy.objects.create(names="KNeighborsClassifier",
ratio=accuracy_score(y_test, knpredict) * 100)

    print("Gradient Boosting Classifier")

    from sklearn.ensemble import GradientBoostingClassifier
    clf = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0,
max_depth=1, random_state=0).fit(
        X_train,
        y_train)
    clfpredict = clf.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, clfpredict) * 100)
    print("CLASSIFICATION REPORT")
print(classification_report(y_test, clfpredict))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, clfpredict))

models.append(('GradientBoostingClassifier', clf))


detection_accuracy.objects.create(names="Gradient Boosting Classifier",
                                  ratio=accuracy_score(y_test, clfpredict) * 100)


csv_format = 'Results.csv'
df.to_csv(csv_format, index=False)
df.to_markdown

obj = detection_accuracy.objects.all()
return render(request,'SProvider/train_model.html', {'objs': obj})
```

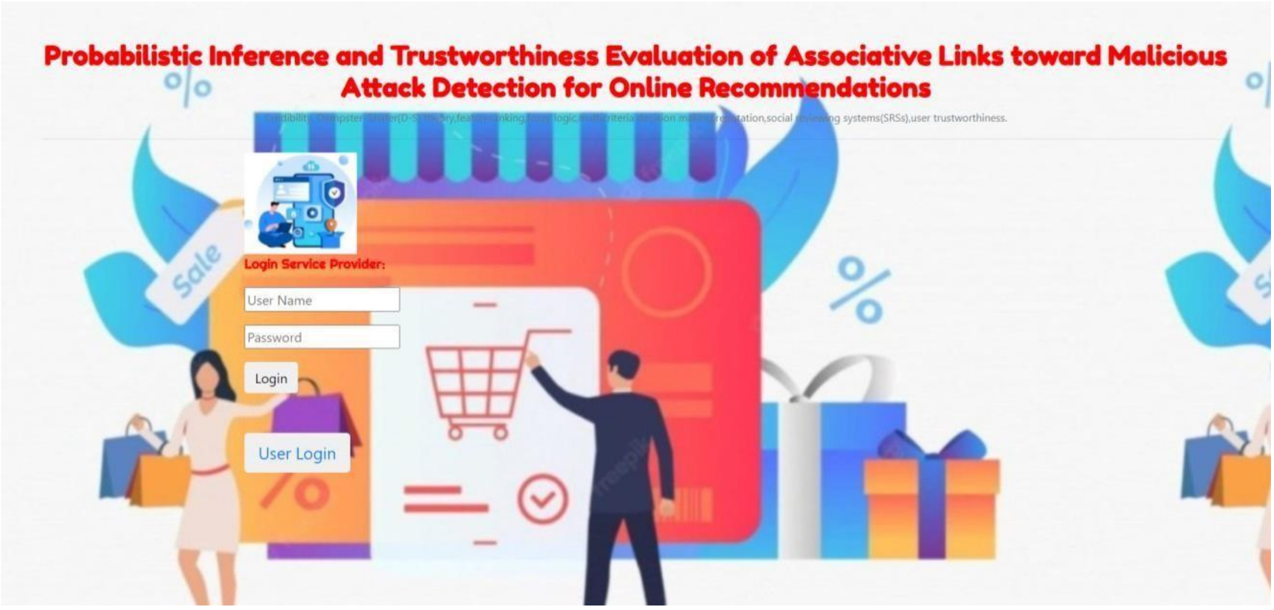
6. SCREENSHOTS

6 SCREENSHOTS



Credibility, Dempster–Shafer(D-S) theory,featureranking,fuzzy logic,multicriteria decision making,reputation,social reviewing systems(SRSs),user trustworthiness.

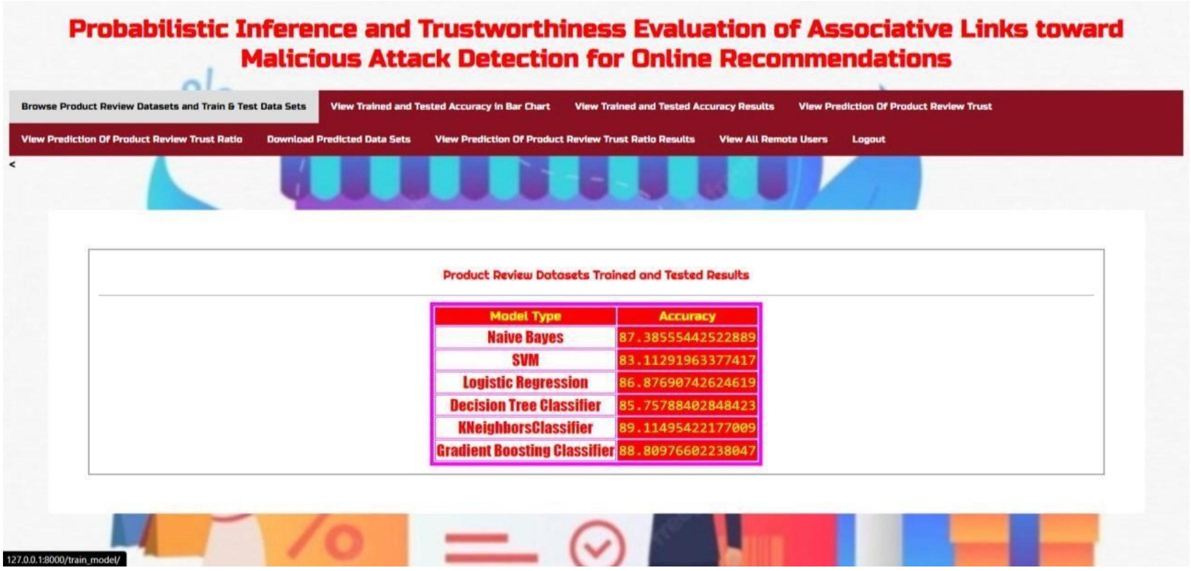
Screenshot 6.1: Home Page



Probabilistic Inference and Trustworthiness Evaluation of Associative Links toward Malicious Attack Detection for Online Recommendations

Credibility, Dempster-Shafer(D-S) theory,featureranking,fuzzy logic,multicriteria decision making,reputation,social reviewing systems(SRSs),user trustworthiness.

Screenshot 6.2: Service Provider Login

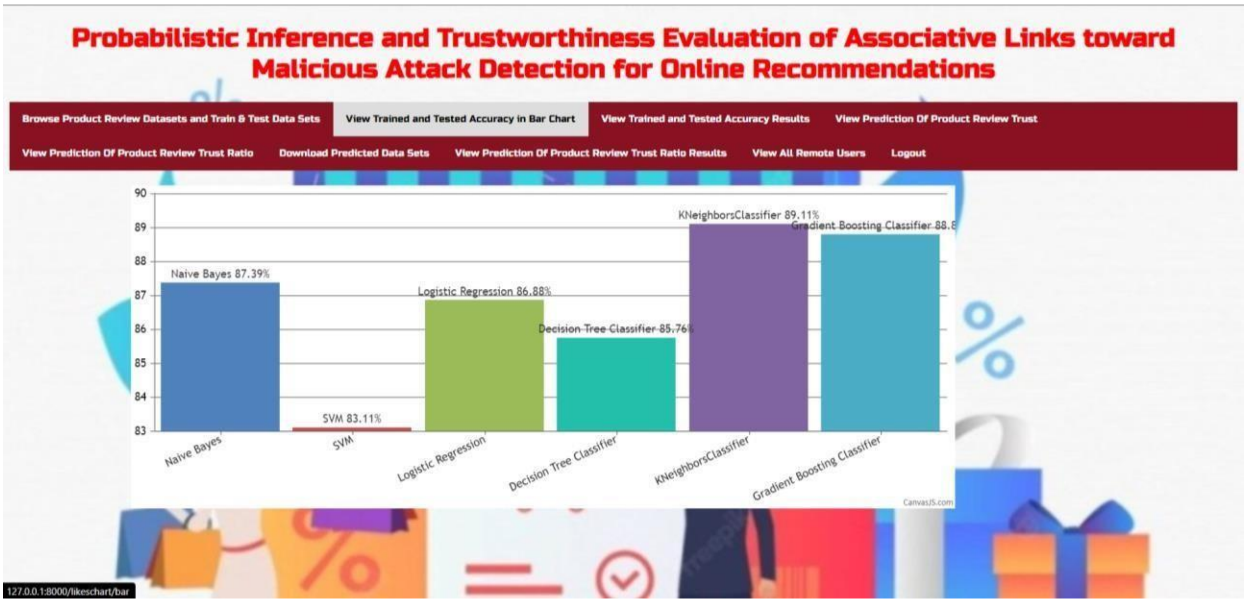

Browse Product Review Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Product Review Trust
View Prediction Of Product Review Trust Ratio
Download Predicted Data Sets
View Prediction Of Product Review Trust Ratio Results
View All Remote Users
Logout
<
127.0.0.1:8000/train_model/

Probabilistic Inference and Trustworthiness Evaluation of Associative Links toward Malicious Attack Detection for Online Recommendations

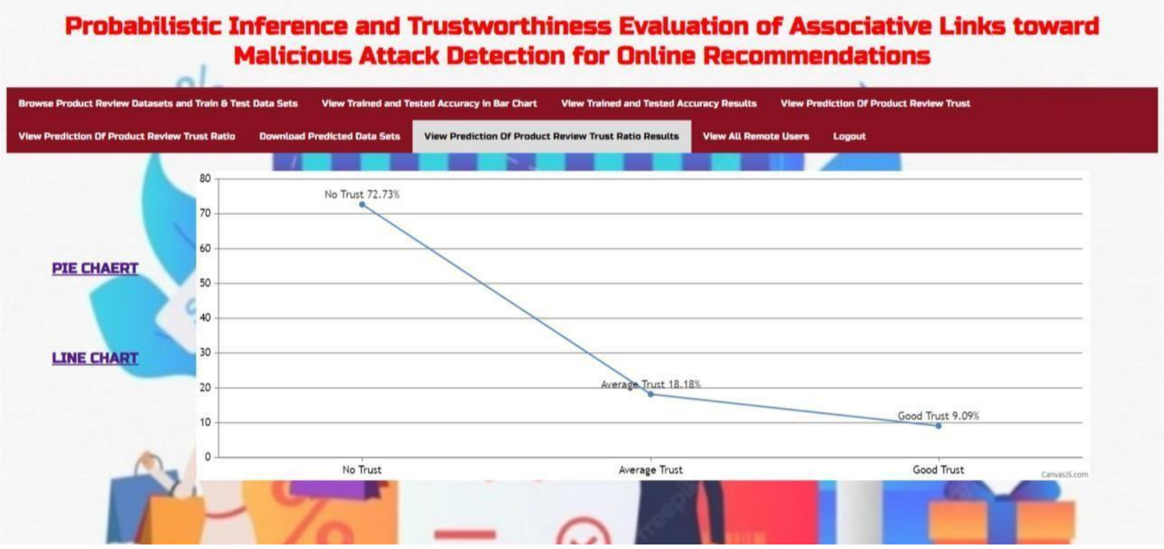
Product Review Datasets Trained and Tested Results

Model Type	Accuracy
Naive Bayes	87.38555442522889
SVM	83.11291963377417
Logistic Regression	86.87690742624619
Decision Tree Classifier	85.75788402848423
KNeighborsClassifier	89.11495422177009
Gradient Boosting Classifier	88.80976602238047

Screenshot 6.3: Browse Product Review Datasets and Train and Test Datasets

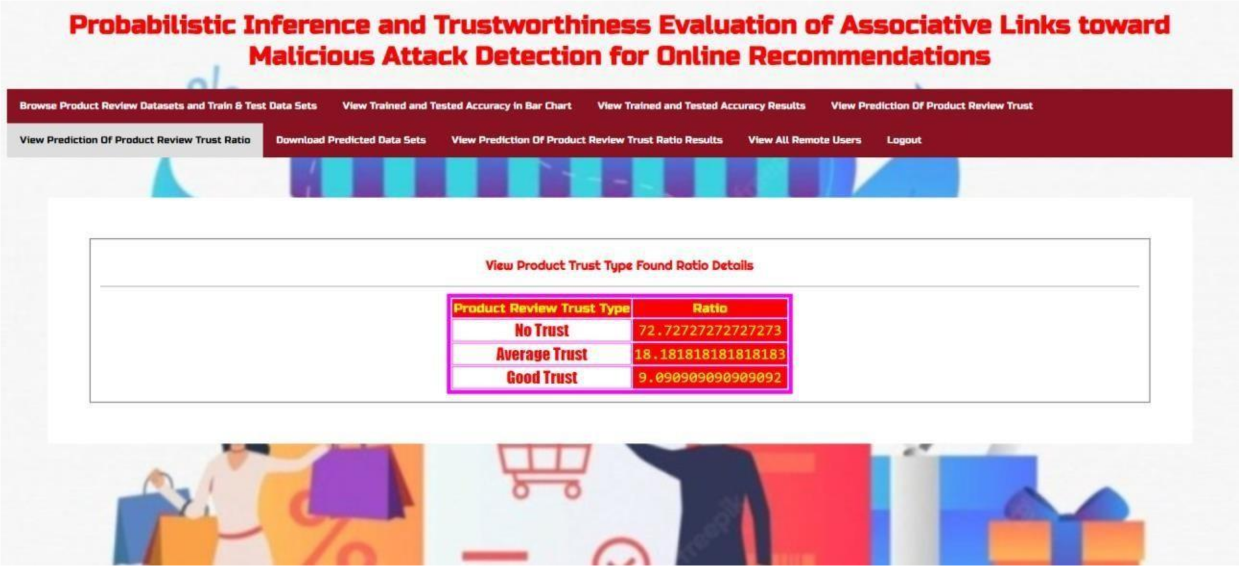

Probabilistic Inference and Trustworthiness Evaluation of Associative Links toward Malicious Attack Detection for Online Recommendations
Browse Product Review Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Product Review Trust
View Prediction Of Product Review Trust Ratio
Download Predicted Data Sets
View Prediction Of Product Review Trust Ratio Results
View All Remote Users
Logout
90
89
88
87
86
85
84
83
Naive Bayes 87.39%
SVM 83.11%
Logistic Regression 86.88%
Decision Tree Classifier 85.76%
KNeighborsClassifier 89.11%
Gradient Boosting Classifier 88.8
Naive Bayes
SVM
Logistic Regression
Decision Tree Classifier
KNeighborsClassifier
Gradient Boosting Classifier
CanvasJS.com
127.0.0.1:8000/likeschart/bar

Screenshot 6.4: Trained and Tested Accuracy in Bar Chart


Probabilistic Inference and Trustworthiness Evaluation of Associative Links toward Malicious Attack Detection for Online Recommendations

Browse Product Review Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Product Review Trust
View Prediction Of Product Review Trust Ratio
Download Predicted Data Sets
View Prediction Of Product Review Trust Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
80
70
60
50
40
30
20
10
0
No Trust 72.73%
Average Trust 18.18%
Good Trust 9.09%
No Trust
Average Trust
Good Trust
CanvasJS.com

Screenshot 6.5: Prediction of Product Review Trust Ratio Graph


Probabilistic Inference and Trustworthiness Evaluation of Associative Links toward Malicious Attack Detection for Online Recommendations

Browse Product Review Datasets and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Product Review Trust
View Prediction Of Product Review Trust Ratio
Download Predicted Data Sets
View Prediction Of Product Review Trust Ratio Results
View All Remote Users
Logout

View Product Trust Type Found Ratio Details

Product Review Trust Type	Ratio
No Trust	72.72727272727273
Average Trust	18.181818181818183
Good Trust	9.090909090909092

Screenshot 6.6: Prediction of Product Review Trust Ratio

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. Testing is event 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.

7.2.3 WHITE BOX TESTING

White Box Testing is a testing in which 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.

7.2.4 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 testing in which the software under test is treated, as a black box, you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

7.3 TEST CASES

7.3.1 CLASSIFICATION

S.NO	Test Case	Excepted Result	Result	Remarks(IF Fails)
1.	User Register	If User registration successfully.	Pass	If already user email exist then it fails.
2.	User Login	If Username and password is correct then it will getting valid page.	Pass	Un Register Users will not logged in.
3.	User View User	Show our dataset	Pass	If Data set Not Available fail.
4.	View Fast History Results	The Four Alarm Score Should be Displayed.	Pass	The Four Alarm Score Not Displaying fail
5.	User Prediction	Display Review with true results	Pass	Results not True Fail
6.	Show Detection process	Display Detection process	Pass	Results Not True Fail
7.	Show Eye Blink Process	Display Eye Blink Process	Pass	If Results not Displayed Fail.
8.	Admin login	Admin can login with his login credential. If success he get his home page	Pass	Invalid login details will not allowed here
9.	Admin can activate the register users	Admin can activate the register user id	Pass	If user id not found then it won't login
10.	Results	For our Four models the accuracy and F1 Score	Pass	If Accuracy And F1 Score Not Displayed fail

8. CONCLUSION & FUTURE SCOPE

8 CONCLUSION & FUTURE SCOPE

8.1 CONCLUSION

Defending malicious injection attacks for recommender systems faces a long-standing but unresolved issue, namely dimensionality reduction and the association representation of sparse rating data. Without understanding the nuances of fake injection profiles and authentic profiles, one can fall into the chasm of misleading correlation, probably leading to false discovery and insight. In this paper, we present a unified detection framework using a divide- and-conquer strategy to spot malicious attacks including co-visitation injection attacks, profile injection attacks, etc.

Experimental results demonstrated that disturbed data determined by user activity, item popularity, and transition probability between users and items can be effectively eliminated in advance, which is also favorable to shrink the scope of detection and also reduce the cost of computation and evaluating the trustworthiness of associative links in coupled networks after eliminating disturbed information can be improved, which significantly enhances the detection performance of the proposed approach compared with competing baselines.

In our future work, we will explore the sensitivity of parameters across different synthetic networks, such as scale free, small-world, etc. Additionally, facing with new threats focused on recommender systems, such as data poisoning attacks on factorization-based collaborative filtering, poisoning attacks to graph-based recommender systems etc., how to design a unified detection framework to deal with these diverse threats is also an open issue. Moreover, investigating abnormality forensics for real data is necessary to be focused.

8.2 FUTURE SCOPE

The future scope for probabilistic inference and trustworthiness evaluation of associative links for malicious attack detection in online recommendations is promising, with several potential areas of advancement:

- Integration of Advanced Machine Learning Techniques
- Privacy-Preserving Mechanisms
- Cross-Domain Application

9. BIBILIOGRAPHY

9 REFERENCES

9.1 REFERENCES

- M. Fang, G. Yang, N. Gong, and J. Liu, “Poisoning attacks to graph-based recommender systems,” ACSAC, arXiv preprint arX_x0002_iv:1809.04127, 2018.
- P. Chirita, W. Nejdl, and C. Zamfir, “Preventing shilling attacks in online recommender systems,” In Proceedings of the 7th annual ACM WIDM, pp. 67–74, 2005.
- C. A. Williams, B. Mobasher, R. Burke, and R. Bhaumik, “Detecting profile injection attacks in collaborative filtering: a classification_x0002_based approach,” Advances in Web Mining and Web Usage Analysis, pp. 167–186, 2007.
- J. Calandrino, A. Kilzer, A. Narayanan, E. Felten, and V. Shmatikov, “You might also like: Privacy risks of collaborative filtering,” IEEE Symposium on Security and Privacy (SP), pp. 231–246, 2011.

9.2 GITHUB LINK

https://github.com/bollepallylahari/Probabilistic-Inference