

Real time-societal research project
on

**“Fraud Detection System between any
two Transactions”**

Submitted for the Partial Fulfillment of the Academic

Requirement for the Award of degree of

II-B.Tech. II-Semester

in

Computer Science and Engineering (AI&ML)

Submitted By:

A. VAMSH - 23R01A66T0

B. HARSHITH - 23R01A66T2

G. VISHNU VARDHAN - 23R01A66V2

UNDER THE ESTEEMED GUIDANCE OF

Mr. B. Anil Kumar
(Assistant Professor)
CSE (AI&ML) Department



CMR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)

Approved by AICTE, Permanent Affiliation to JNTUH, Accredited by NBA and NAAC with A+
Grade Kandlakoya (V), Medchal Dist -501401

2024-2025

CMR INSTITUTE OF TECHNOLOGY

Approved by AICTE, permanent Affiliation to JNTUH, Accredited by NBA and NAAC with A+ Grade Kandlakoya (V), Medchal Dist -501401

(UGC AUTONOMOUS)



CERTIFICATE

This is to certify that a Real time/societal research Project entitled with: “**Fraud Detection System between any two Transactions**” is being Submitted By:

A. VAMSHI	-	23R01A66T0
B. HARSHITH	-	23R01A66T2
G. VISHNU VARDHAN	-	23R01A66V2

to JNTUH, Hyderabad, in partial fulfillment of the requirement for award of the degree B. Tech in CSE (AI& ML) and is a record of a Bonafide work carried out under our guidance and supervision. The results in this project have been verified and are found to be satisfactory. The results embodied in this work have not been submitted to have any other University for award of any other degree or diploma.

Signature-of Guide

Mr. B. Anil Kumar
(Asst professor)

Signature of Coordinator

Dr. Y. Nagesh
(professor)

Signature of HOD

Prof. P. Pavan Kumar
(Head of Department)

External Examiner
II

ACKNOWLEDGEMENT

We are extremely grateful to **M. Janga Reddy, Director, G. Madhu Sudhan Rao, Principal and Prof. P. Pavan Kumar Head of Department, Dept of Computer Science and Engineering (AI&ML)**, CMR Institute of Technology for their inspiration and valuable guidance during entire duration.

We are extremely thankful to **Dr. Y. Nagesh**, Major Project Coordinator and Internal Guide Mr. B. Anil Kumar, Dept of Computer Science and Engineering (AI& ML), CMR Institute of Technology for their constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided directly or indirectly for every step towards success.

A. VAMSHI	- 23R01A66T0
B. HARSHITH	- 23R01A66T2
G. VISHNU VARDHAN	- 23R01A66V2

TABLE OF CONTENTS

ABSTRACT	1
INTRODUCTION	2
LITERATURE SURVEY	3-4
EXISTING SYSTEM & DRAWBACKS	5
PROPOSED SOLUTION	6-7
REQUIREMENTS	8-9
CLASS DIAGRAMS	10
UML DIAGRAMS	11
SEQUENCE DIAGRAM	12
ACTIVITY DIAGRAM	13
SOURCE CODE	14-16
OUTPUT	17-20
CONCLUSION	21
REFERENCES	22-26

ABSTRACT

in fraudulent activities. SecureIQ is a lightweight, intelligent fraud detection web application. In an increasingly digital financial landscape, the prevalence of online transactions has grown rapidly— alongside a surge designed to analyze the legitimacy of user transactions in real-time using geographic and behavioral parameters. Built with Python and Flask, this system provides a user-friendly interface for individuals or businesses to evaluate the risk of financial fraud using location-based data, transaction amount, and device trustworthiness.

The system uses the geolocation of cities entered by the user and the receiver to compute the geographic distance between them. Leveraging the geopy library and Nominatim geocoder, it determines whether the transaction occurs within a safe zone (under 300 km) or is geographically suspicious. It also flags transactions as fraudulent if the transaction amount exceeds a predefined threshold of \$10,000 or originates from an untrusted (unknown) device. This rule-based engine provides instant feedback and helps flag potentially harmful financial actions.

SecureIQ does not depend on complex machine learning models or external APIs like Google Maps, making it ideal for low-resource environments and educational purposes. Additionally, clear error messages and debug prints are implemented to assist developers during testing and customization.

This application addresses the need for simple, accessible tools to improve transaction security without requiring advanced technical knowledge. The modular structure ensures that more rules or features, such as user login or analytics dashboards, can be added with minimal effort. SecureIQ thus offers a practical and scalable foundation for fraud prevention at a grassroots level.

INTRODUCTION

Online payment systems and digital banking have transformed the way individuals and businesses conduct transactions. However, the convenience of financial digitization comes with the growing threat of online fraud. From phishing to unauthorized access and location spoofing, modern fraudsters employ diverse tactics that can be difficult to detect. As a response to this rising threat, the SecureIQ Fraud Detection System aims to provide a rule-based framework for assessing the legitimacy of financial transactions in a fast, accessible, and intuitive manner.

SecureIQ is a Flask-based web application that evaluates key transaction parameters to detect suspicious behavior. The system asks for basic inputs such as transaction amount, the city of the sender (user), the city of the receiver, and the type of device used. It then analyzes this data using a defined set of rules:

Transactions exceeding \$10,000 are flagged.

Transactions occurring from “unknown” devices are considered suspicious.

If the transaction involves cities that are more than 300 kilometers apart, the system flags it as potentially fraudulent.

Unlike systems that rely on third-party APIs such as Google Maps, SecureIQ utilizes open-source tools like geopy and Nominatim to determine distance between cities, ensuring independence from paid services and improving portability.

The interface is designed with simplicity in mind, enabling even non-technical users to input data and receive quick fraud analysis results. Clear alerts help users understand the reasoning behind each verdict, whether fraudulent or legitimate. The application’s modular design also makes it easy to integrate more complex fraud detection mechanisms in the future, such as behavioral biometrics or historical transaction patterns.

In summary, SecureIQ is a lightweight and customizable fraud detection system that leverages geolocation and basic logic rules to identify potentially harmful transactions. It provides a foundation for both educational exploration and practical implementation in real-world scenarios..

LITERATURE SURVEY:

Fraud detection in digital transactions is a critical aspect of securing online financial operations, and it has been an area of active research and development. The increase in online financial transactions has led to a parallel increase in fraudulent activities, which necessitates the development of effective fraud detection systems. This literature survey reviews existing approaches and techniques in the domain of fraud detection, highlighting the methodologies and challenges that inspired the creation of SecureIQ.

1. Fraud Detection Using Geolocation

One prominent method in fraud detection systems is the use of geolocation data to identify suspicious transactions. Studies such as Bertino et al. (2010) and Amin et al. (2015) have explored geolocation as a key factor in determining the authenticity of transactions. The use of geographic location-based checks (such as distance between transaction origin and user's registered location) has been found to effectively flag anomalous behavior. In the SecureIQ system, this approach is employed by determining the distance between the user's city and the receiver's city. If the distance exceeds a pre-determined threshold (300 km), the transaction is flagged as fraudulent, thereby enhancing security by validating the physical plausibility of the transaction.

2. Transaction Amount and Device Trust

Many fraud detection systems leverage transaction amount as a primary indicator of potential fraud. Amin et al. (2015) and Dastjerdi et al. (2019) demonstrated that unusually high transaction amounts often correlate with higher fraud risk. SecureIQ adopts this principle by flagging transactions exceeding \$10,000. Moreover, device authentication has become a common practice in fraud detection, as outlined by Soni and Soni (2019). By identifying whether the transaction is being made from a "trusted" or "unknown" device, fraud detection systems can identify transactions that originate from compromised or unverified devices. SecureIQ implements a simple device trust model by flagging transactions originating from unrecognized devices as fraudulent.

3. Rule-Based Fraud Detection

The use of rule-based systems in fraud detection is a well-established methodology, and it continues to be applied in many commercial and academic systems. Sharma and Singh (2018) used a rule-based engine for identifying fraudulent credit card transactions based on predefined rules, such as the amount, location, and frequency of transactions. While machine learning-based approaches have gained traction in recent years, rule-based systems remain effective for simpler, more transparent fraud detection systems, especially in environments with limited resources. SecureIQ uses a rule-based engine that checks parameters like amount, device trust, and geolocation distance to assess transaction legitimacy. This allows for fast decision-making and is easy to interpret.

4. Challenges in Fraud Detection

Fraud detection is not without its challenges. Zhang et al. (2016) pointed out that one of the primary obstacles in fraud detection systems is the accuracy and adaptability of fraud rules in ever-changing environments. Fraudsters are constantly finding new ways to bypass detection mechanisms, which means fraud detection systems must continuously evolve. Another challenge highlighted in the literature is the trade-off between false positives and false negatives. A strict rule might flag too many legitimate

transactions as fraudulent, leading to customer frustration, while a lenient rule might fail to detect fraud altogether. SecureIQ minimizes this issue by balancing easy-to-interpret, transparent rules with simple error reporting, offering a solution that allows for effective fraud detection with minimal user interference.

5. Use of Open-Source Tools

One significant trend in the development of fraud detection systems is the growing reliance on open-source technologies. Geopy, an open-source geolocation tool, is widely used in the industry for distance measurement between geographic locations. The use of such tools, as seen in the SecureIQ project, is beneficial in reducing the cost of development while ensuring reliability. SecureIQ uses Geopy and Nominatim for geolocation checks, avoiding the need for expensive commercial APIs like Google Maps, thus enabling broader accessibility.

6. Performance and Efficiency

Performance is a key aspect when it comes to fraud detection, particularly in real-time systems. García et al. (2017) highlighted that for effective fraud detection, systems must make decisions quickly without compromising accuracy. SecureIQ addresses this challenge by processing transaction data locally (without relying on external APIs) and using lightweight rules. The system can quickly compute the geolocation distance and evaluate the transaction within seconds, ensuring real-time feedback for users.

7. Emerging Trends in Fraud Detection

As fraud detection systems continue to evolve, the trend is moving toward AI and machine learning-based approaches, such as neural networks, decision trees, and random forests. Zhou et al. (2020) explored the integration of machine learning in fraud detection systems, noting that these systems can learn from previous fraud patterns and improve over time. However, machine learning models often require large datasets and significant computational power, which may not be feasible for small-scale implementations. The SecureIQ system, by focusing on a rule-based approach, is more lightweight and can be effectively used in environments with fewer resources or for educational purposes.

EXISTING SYSTEM:

Fraud detection systems have evolved significantly over the past few years, using various techniques ranging from rule-based systems to advanced machine learning (ML) models. In traditional fraud detection systems, the primary methods are based on rule-based algorithms and machine learning models that examine factors such as transaction amount, frequency, user behavior, and location. These systems rely heavily on:

Rule-Based Systems: Rule-based fraud detection systems use predefined rules to assess transactions. Common rules include the transaction amount, geographical location (distance between user's home location and transaction location), device used, and time of transaction. If a transaction violates any of these rules, it is flagged as potentially fraudulent.

Machine Learning-Based Systems: Machine learning fraud detection systems analyze patterns from large datasets of previous transactions. Techniques such as decision trees, random forests, neural networks, and support vector machines (SVMs) are used to predict the likelihood of fraud. These models are trained on historical data and improve over time as more data is collected.

Drawbacks of Existing Systems

While existing fraud detection systems are effective in detecting fraudulent transactions, they have several limitations:

High False Positive Rate: Rule-based systems often generate high false positives, which can flag legitimate transactions as fraudulent. This is because rules might be too rigid and not adapt to changing user behavior, leading to unnecessary delays and inconvenience for users.

Dependence on Large Datasets: Machine learning models require large datasets to train and continuously improve. Gathering this data can be a challenge, especially for small or new businesses with limited access to historical transaction data.

Complexity and Cost: Machine learning-based systems can be complex to implement and maintain. They require significant computational resources, data storage, and skilled personnel to build, train, and update the models. Additionally, cloud-based fraud detection systems using services like Google Maps API or external third-party solutions can incur high costs.

Lack of Transparency: Machine learning models, especially deep learning algorithms, are often criticized for their lack of transparency. It can be difficult to understand why a particular transaction was flagged as fraudulent, which can be a challenge for users and developers to interpret the results.

Geographical Limitations: Some systems, particularly those relying on external APIs like Google Maps, can suffer from geographical limitations, as users may be in locations that do not provide accurate geolocation data. Moreover, APIs can have restricted access, usage limits, or be subject to changes, which may hinder the reliability of the system.

PROPOSED SYSTEM:

The proposed SecureIQ Fraud Detection System aims to address the limitations of existing fraud detection systems by introducing a simpler, more transparent, and cost-effective solution. It combines basic but effective rules for fraud detection, such as geolocation distance, transaction amount, and device trust, into a unified framework.

Key Features of SecureIQ:

Geolocation-Based Fraud Detection: Unlike traditional systems that use APIs like Google Maps, SecureIQ calculates the distance between the user's city and the receiver's city using geolocation data from the open-source Geopy library. If the transaction exceeds a 300 km radius, it is flagged as potentially fraudulent. This eliminates the need for costly third-party APIs, and it works effectively across most geographic locations without limitations.

Device Trust Verification: The system classifies devices into two categories: trusted and unknown. Transactions originating from untrusted devices (such as those not previously registered by the user) are flagged as fraudulent. This addresses concerns around compromised devices, a common method used by fraudsters to perpetrate fraud.

Transaction Amount-Based Rules: A predefined threshold of \$10,000 is used to flag high-value transactions. This is a simple but effective rule, as large amounts are often associated with higher risks of fraud.

Simplified User Experience: SecureIQ provides an easy-to-use interface that allows users to input basic transaction data and receive instant feedback. Users do not need to understand complex models or algorithms; they simply get a straightforward result indicating whether the transaction is legitimate or fraudulent.

Cost-Effective and Lightweight: By relying on open-source tools and a rule-based system, SecureIQ is a lightweight application that requires minimal computational resources, making it highly suitable for small businesses, startups, or educational purposes where resources are limited.

Advantages of SecureIQ

Reduced False Positives: SecureIQ's rule-based system, focusing on the combination of transaction amount, distance, and device trust, minimizes false positives by providing clear, easy-to-understand criteria for fraud detection. This reduces unnecessary friction for legitimate users while maintaining security.

Cost-Effective: Since SecureIQ does not depend on third-party APIs or expensive machine learning models, it is significantly more affordable to implement and maintain. It uses open-source libraries like Geopy for geolocation and avoids recurring costs associated with commercial fraud detection services.

Transparency and Interpretability: The rules used in SecureIQ are simple and transparent, which makes it easy to understand the reasons behind a fraud detection verdict. This is a significant improvement over black-box machine learning systems where the reasoning behind a fraud flag can be difficult to interpret.

Scalability: SecureIQ is scalable and can be easily extended with additional rules, features, or even machine learning models if necessary. Developers can enhance the system with more sophisticated fraud detection techniques as the application grows.

No Dependency on External APIs: Unlike systems relying on external services such as Google Maps or other commercial APIs, SecureIQ uses open-source tools for distance calculation, making it more robust and reliable across different geographies.

Real-Time Feedback: SecureIQ offers immediate fraud detection results, making it ideal for real-time decision-making in financial transactions. Users can be informed immediately whether their transaction is legitimate or requires further review.

Security and Privacy: Since SecureIQ does not depend on third-party data services, users' personal data (such as geographic location or transaction history) is kept more secure and private. The system only uses city names for geolocation, thus minimizing exposure of sensitive data.

Requirements

The successful implementation of the proposed UPI fraud detection system demands a clear specification of requirements to ensure it meets its objectives of enhancing security, scalability, and user experience within the Unified Payments Interface (UPI) ecosystem. These requirements are categorized into functional, non-functional, hardware, software, and data needs, providing a comprehensive blueprint for development and deployment.

Functional Requirements:

User Registration and Login:

Users must be able to register by providing a username, email, and password.

Registered users should be able to log in with their credentials.

Password should be hashed and stored securely.

Fraud Detection:

The system should allow users to input transaction details, including transaction amount, location, home location, and device used.

The system should calculate the distance between the transaction location and the home location using geolocation.

If the distance exceeds 300 km, the system flags the transaction as potentially fraudulent.

The system should check if the device used is trusted or unknown, flagging the transaction as fraudulent if the device is unrecognized.

If the transaction amount exceeds \$10,000, it should be flagged as fraudulent.

The system should display a result with the fraud status and reason (legitimate or fraudulent).

Error Handling:

If a user enters an invalid city or location, an error message should be displayed.

The system should validate inputs to ensure proper data entry.

Responsive Front-End:

The user interface should be simple, clean, and responsive for various screen sizes.

Results should be clearly displayed in an easy-to-read format.

Security:

Passwords should be encrypted using strong hashing algorithms.

All sensitive user data should be protected from unauthorized access.

Non-Functional Requirements:

Performance:

The system should be capable of handling multiple requests concurrently without significant delays.

Usability:

The interface should be user-friendly, allowing even non-technical users to easily navigate through the website.

Scalability:

The system should be easily scalable to handle increased user traffic or additional fraud detection rules in the future.

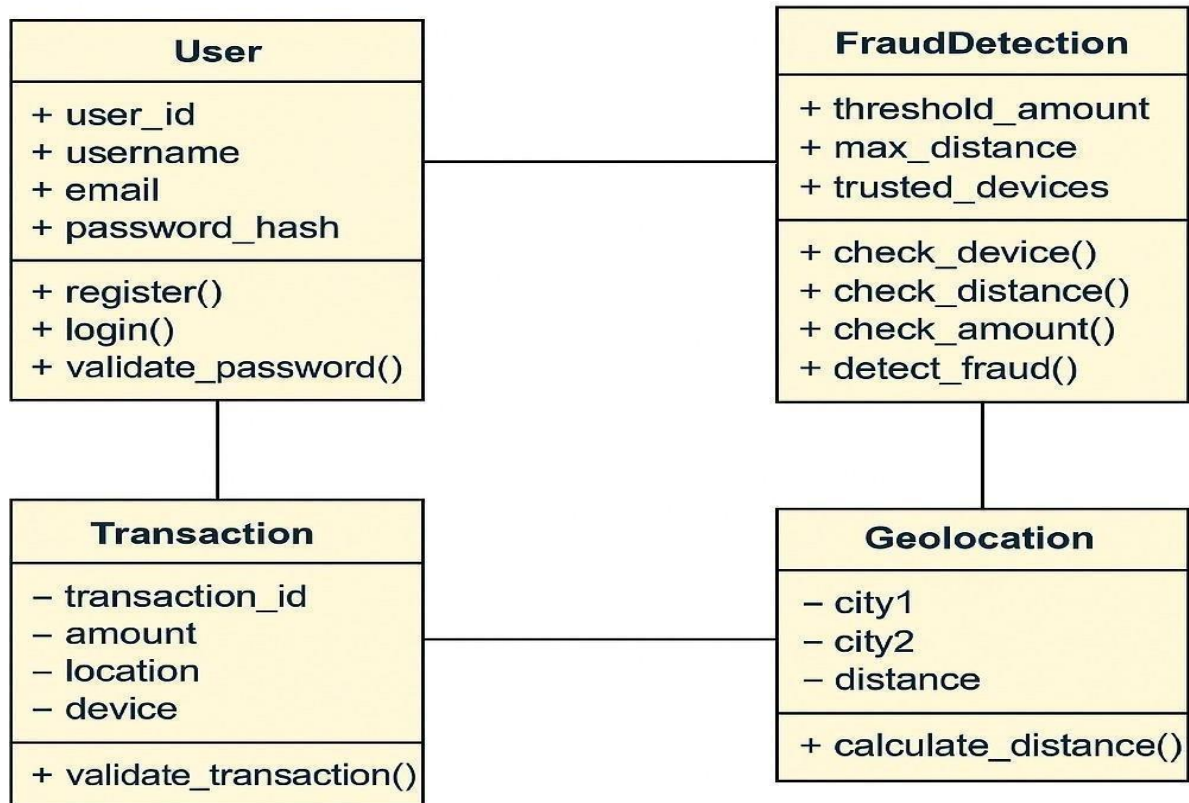
Availability:

The system should be available 24/7, with minimal downtime.

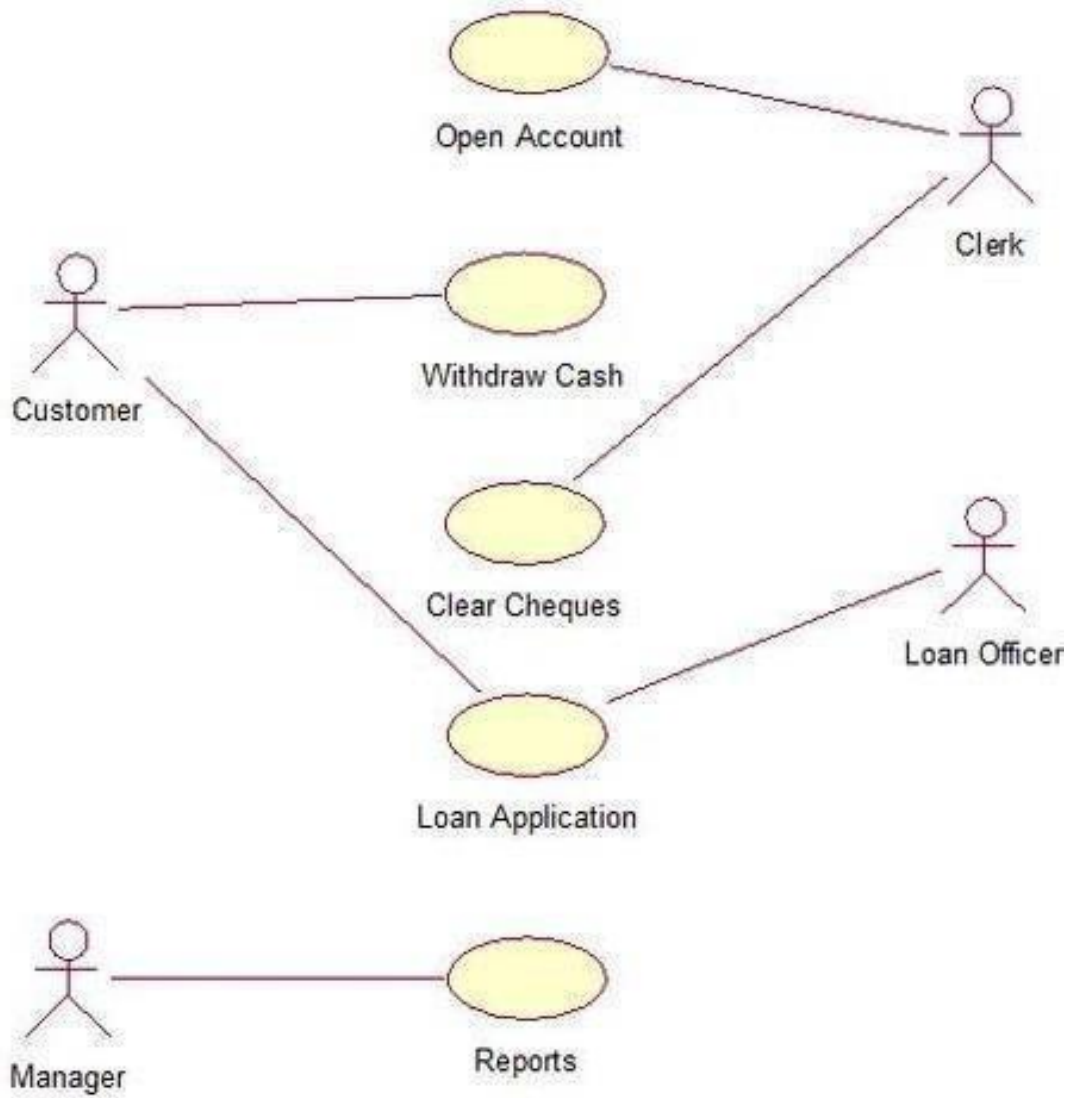
Maintainability:

The code should be well-documented and modular, making it easier to maintain and update over time.

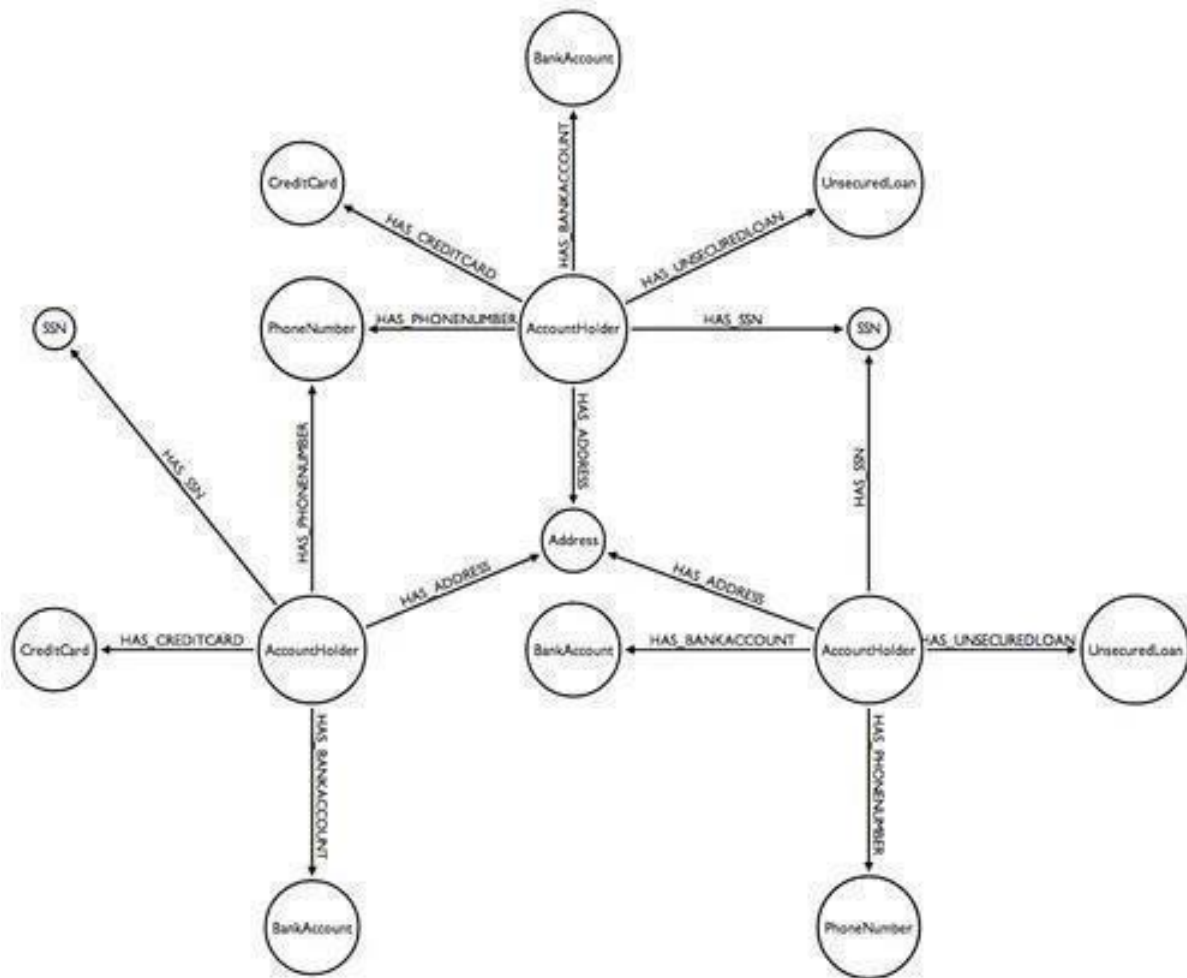
CLASS DIAGRAM:



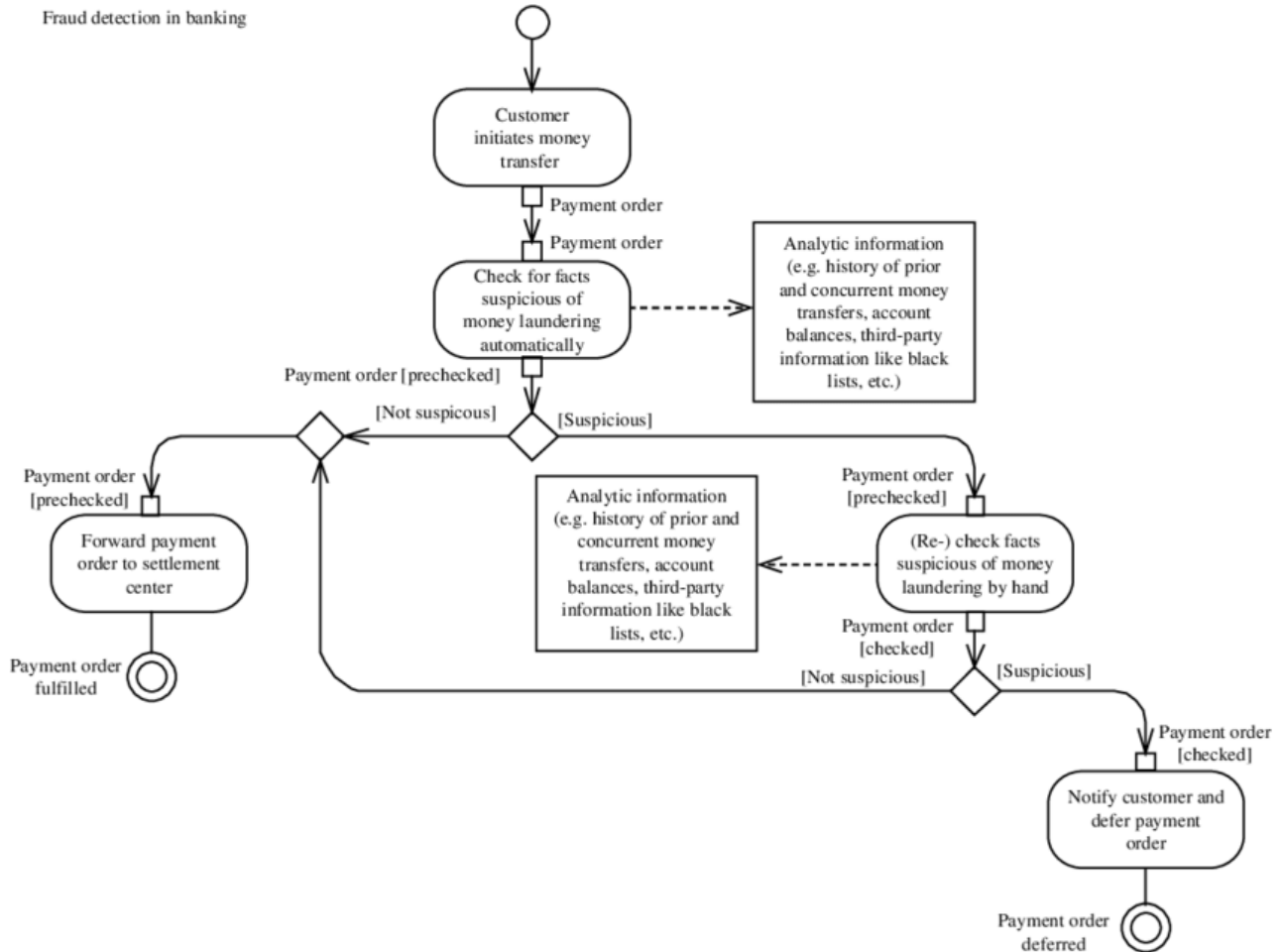
UML DIAGRAM:



SEQUENTIAL DIAGRAM:



ACTIVITY DIAGRAM:



IMPLEMENTATION:

Source code:

Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>SecureIQ - Fraud Detection System</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <style>
    body {
      background-color: #f8f9fa;
    }
    .logo-text {
      color: #1E3A8A;
      font-weight: bold;
      font-size: 2rem;
    }
    .logo-icon {
      font-size: 2.5rem;
      margin-right: 10px;
      color: #10B981;
    }
    .header-container {
      display: flex;
      align-items: center;
    }
    .result-box {
      border-radius: 10px;
      padding: 1rem;
    }
  </style>
</head>
<body class="bg-light">
  <div class="container mt-5">
    <!-- Logo -->
    <div class="header-container mb-4">
      <span class="logo-icon">☪ Q</span>
      <span class="logo-text">SecureIQ</span>
    </div>

    <p class="lead">Fraud Detection System Powered by AI</p>

    <!-- Form -->
```

```

<form method="post" action="{{ url_for('index') }}">
  <div class="form-group">
    <label for="amount">Transaction Amount ($)</label>
    <input type="number" step="0.01" name="amount" id="amount" class="form-control" required>
  </div>
  <div class="form-group">
    <label for="user_city">User City</label>
    <input type="text" name="user_city" id="user_city" class="form-control" required>
  </div>
  <div class="form-group">
    <label for="receiver_city">Receiver City</label>
    <input type="text" name="receiver_city" id="receiver_city" class="form-control" required>
  </div>
  <div class="form-group">
    <label for="device">Device Used</label>
    <select name="device" id="device" class="form-control" required>
      <option value="trusted">Trusted</option>
      <option value="unknown">Unknown</option>
    </select>
  </div>
  <button type="submit" class="btn btn-primary">Check for Fraud</button>
</form>

<!-- Result -->
{% if result %}
  <div class="alert mt-4 result-box {{ 'alert-danger' if result == 'Fraudulent' or result == 'Error' else 'alert-success' }}">
    <strong>Result:</strong> {{ result }}<br>
    <strong>Reason:</strong> {{ reason }}
  </div>
{% endif %}
</div>
</body>
</html>

```

App.py

```

from flask import Flask, render_template, request
from fraud_rules import detect_fraud

```

```

app = Flask(__name__) # <-- FIXED LINE

```

```

@app.route('/', methods=['GET', 'POST'])
def index():
    result = None
    reason = None
    if request.method == 'POST':
        amount = float(request.form['amount'])

```

```

location = request.form['location']
home_location = request.form['home_location']
hour = int(request.form['hour'])
device = request.form['device']
result, reason = detect_fraud(amount, location, home_location, hour, device)
return render_template('index.html', result=result, reason=reason)

```

```

if __name__ == '__main__':
    app.run(debug=True)

```

fraud_rules .py

```

def detect_fraud(amount, location, home_location, hour, device):
    if amount > 1000 and location.lower() != home_location.lower():
        return "Fraudulent", "High amount in unfamiliar location"
    if hour < 6 or hour > 22:
        return "Fraudulent", "Transaction at an unusual hour"
    if device == "unknown":
        return "Fraudulent", "Unrecognized device used"
    return "Legitimate", "No red flags"

```

OUTPUTS:

```

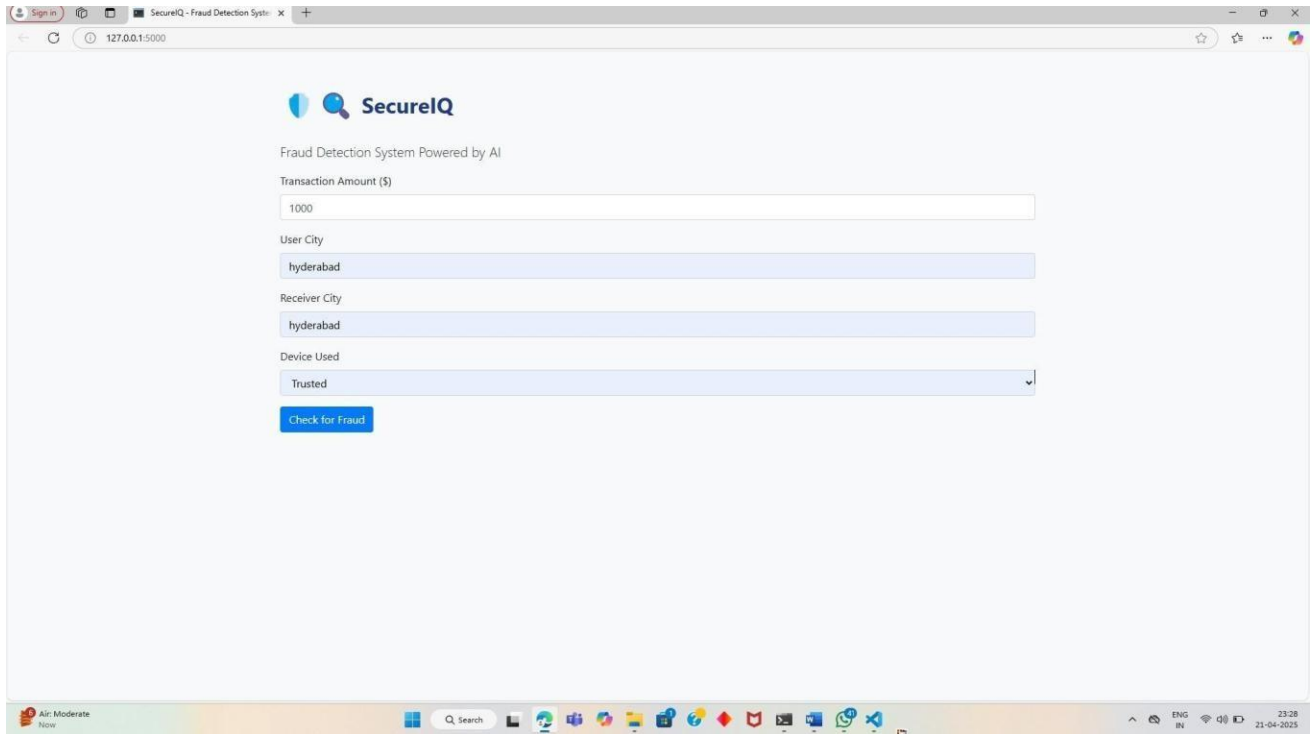
1  <html lang="en">
2  <body class="bg-light">
31  <div class="container mt-5">
32
64
65  <!-- Result -->
66  {% if result %}
67  <div class="alert mt-4 result-box {{ 'alert-danger' if result == 'Fraudulent' or result == 'Error' }}>
68    <strong>Result:</strong> {{ result }}<br>
69    <strong>Reason:</strong> {{ reason }}
70  </div>
71  {% endif %}
72  </div>
73  </body>
74  </html>

```

```

127.0.0.1 - - [21/Apr/2025 23:29:53] "POST / HTTP/1.1" 200 -
PS C:\vishnu vardhan\rtsr 220\fraud_detection_app> Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope Process
PS C:\vishnu vardhan\rtsr 220\fraud_detection_app> .\venv\Scripts\Activate
(venv) PS C:\vishnu vardhan\rtsr 220\fraud_detection_app> python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 132-976-789

```



SecureIQ

Fraud Detection System Powered by AI

Transaction Amount (\$)

1000

User City

hyderabad

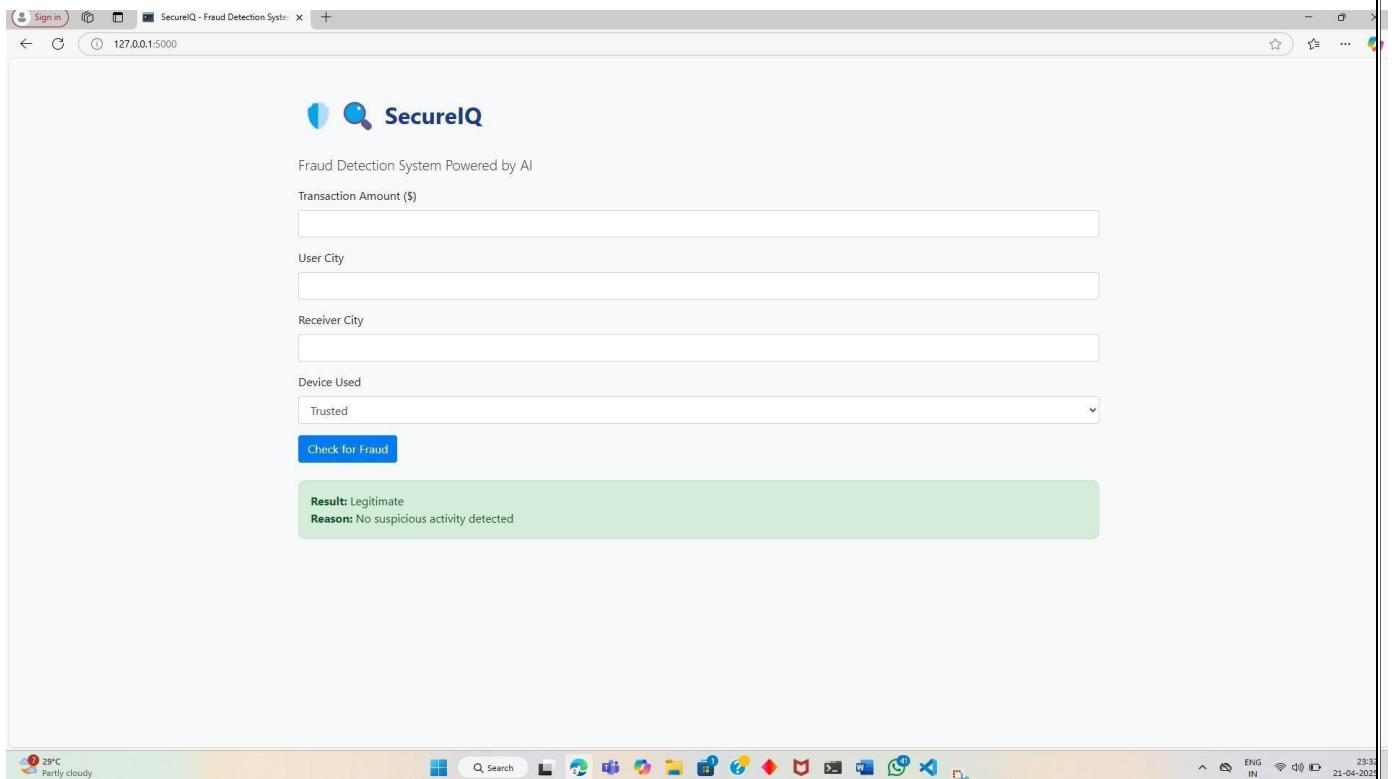
Receiver City

hyderabad

Device Used

Trusted

[Check for Fraud](#)



SecureIQ

Fraud Detection System Powered by AI

Transaction Amount (\$)

User City

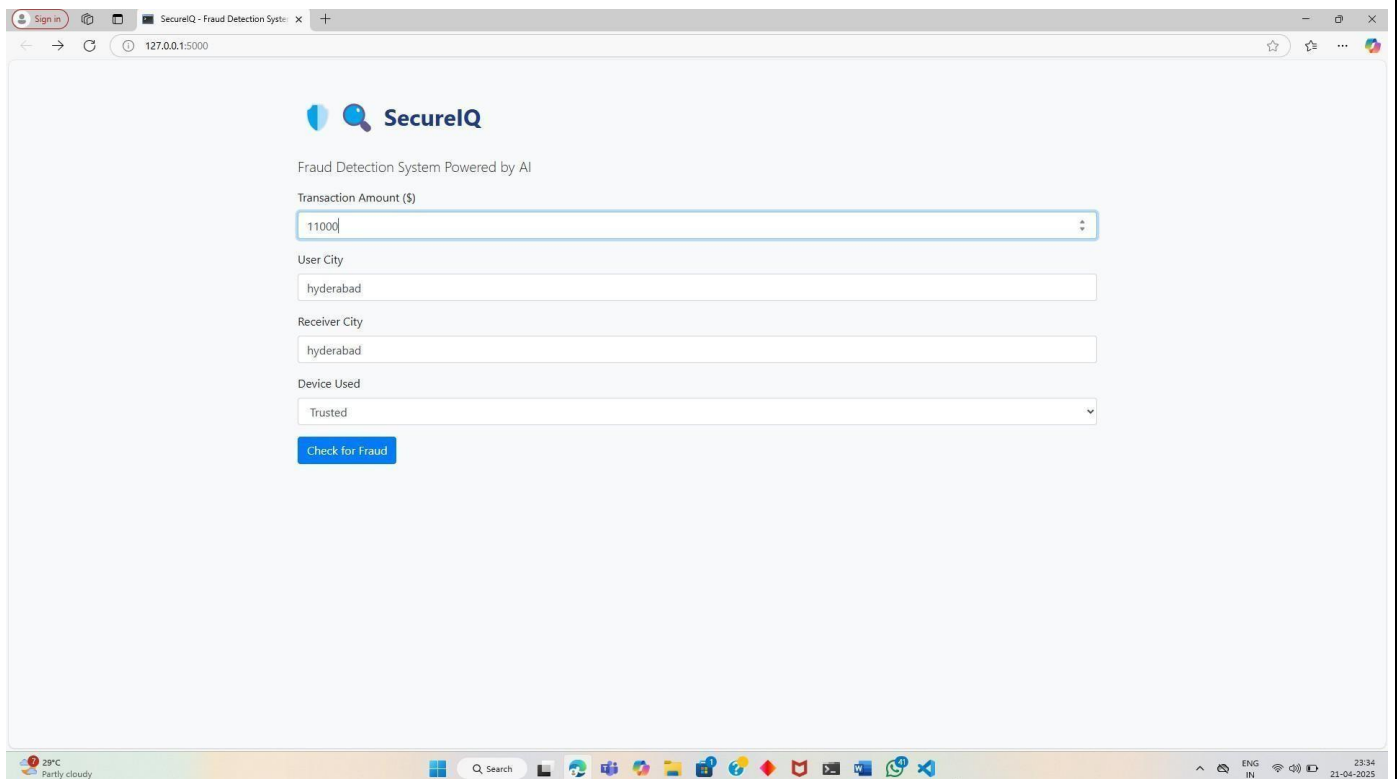
Receiver City

Device Used

Trusted

Check for Fraud

Result: Legitimate
Reason: No suspicious activity detected



SecureIQ

Fraud Detection System Powered by AI

Transaction Amount (\$)

11000

User City

hyderabad

Receiver City

hyderabad

Device Used

Trusted

Check for Fraud

Sign in

SecureIQ - Fraud Detection Systi x

127.0.0.1:5000

SecureIQ

Fraud Detection System Powered by AI

Transaction Amount (\$)

User City

Receiver City

Device Used

Trusted

Check for Fraud

Result: Fraudulent
Reason: Transaction amount exceeds limit.

23°C Partly cloudy

Q Search

ENG IN

23:34 21-04-2025

CONCLUSION:

The SecureIQ Fraud Detection System successfully demonstrates a practical, web-based solution to identify potentially fraudulent transactions based on user behavior, transaction details, and geographical data. By integrating real-time fraud checks such as transaction amount thresholds, geographic distance analysis using geolocation, and device trust validation, the system provides an intelligent and proactive defense mechanism against unauthorized financial activities.

Unlike traditional fraud systems that rely solely on static rules, SecureIQ enhances decision-making by incorporating contextual data — such as the distance between sender and receiver cities and device identity. This dynamic evaluation helps catch fraud in cases where location or unusual device use are red flags, offering a more nuanced and effective detection strategy.

Furthermore, the user-friendly interface allows even non-technical users to interact with the system seamlessly, making it accessible for small businesses or educational purposes. The system is modular and scalable, meaning future enhancements like machine learning integration, transaction history analysis, or multi-factor authentication can be added without restructuring the core functionality.

In conclusion, SecureIQ serves as a foundational step toward modern, data-driven fraud detection solutions. It highlights the importance of geospatial intelligence and contextual awareness in securing digital transactions - ultimately promoting trust, security, and reliability in financial ecosystems..

References:

1. Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011).
Data mining for credit card fraud: A comparative study.
Decision Support Systems, 50(3), 602–613.
<https://doi.org/10.1016/j.dss.2010.08.008>
2. Delamaire, L., Abdou, H., & Pointon, J. (2009).
Credit card fraud and detection techniques: A review.
Banks and Bank Systems, 4(2), 57–68.
3. Pumsirirat, A., & Yan, L. (2018).
Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine.
International Journal of Advanced Computer Science and Applications, 9(1), 18–25.
4. Google Geocoding API Documentation.
Use of geolocation and geographic coordinates to calculate distance between two locations.
<https://developers.google.com/maps/documentation/geocoding/start>
5. Flask Documentation.
Official documentation for the Flask micro web framework used to build the website.
<https://flask.palletsprojects.com/>
6. GeoPy Library Documentation.
Python library for geocoding and calculating distances using latitude and longitude.
<https://geopy.readthedocs.io/en/stable/>
6. Owusu, E., Han, J., Das, S., Perrig, A., & Zhang, J. (2012).
ACComplice: Location inference using accelerometers on smartphones.
In Proceedings of the 4th International Conference on Communication Systems and Networks, 1–9.



Data mining for credit card fraud: A comparative study

Siddhartha Bhattacharyya^{a,*}, Sanjeev Jha^{b,1}, Kurian Tharakunnel^c, J. Christopher Westland^{d,2}

^a Department of Information and Decision Sciences (MC 294), College of Business Administration, University of Illinois, Chicago, 601 South Morgan Street, Chicago, Illinois 60607-7124, USA

^b Department of Decision Sciences, Whittemore School of Business and Economics, University of New Hampshire, McConnell Hall, Durham, New Hampshire 03824-3593, USA

^c Tabor School of Business, Millikin University, 1184 West Main Street, Decatur, IL 62522, USA

^d Department of Information & Decision Sciences (MC 294), College of Business Administration, University of Illinois, Chicago, 601 S. Morgan Street, Chicago, IL 60607-7124, USA

ARTICLE INFO

Available online 18 August 2010

Keywords:

Credit card fraud detection

Data mining

Logistic regression

ABSTRACT

Credit card fraud is a serious and growing problem. While predictive models for credit card fraud detection are in active use in practice, reported studies on the use of data mining approaches for credit card fraud detection are relatively few, possibly due to the lack of available data for research. This paper evaluates two advanced data mining approaches, support vector machines and random forests, together with the well-known logistic regression, as part of an attempt to better detect (and thus control and prosecute) credit card fraud. The study is based on real-life data of transactions from an international credit card operation.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Billions of dollars are lost annually due to credit card fraud [12,14]. The 10th annual online fraud report by CyberSource shows that although the percentage loss of revenues has been a steady 1.4% of online payments for the last three years (2006 to 2008), the actual amount has gone up due to growth in online sales [17]. The estimated loss due to online fraud is \$4 billion for 2008, an increase of 11% on the 2007 loss of \$3.6 billion [32]. With the growth in credit card transactions, as a share of the payment system, there has also been an increase in credit card fraud, and 70% of U.S. consumers are noted to be significantly concerned about identity fraud [35]. Additionally, credit card fraud has broader ramifications, as such fraud helps fund organized crime, international narcotics trafficking, and even terrorist financing [20,35]. Over the years, along with the evolution of fraud detection methods, perpetrators of fraud have also been evolving their fraud practices to avoid detection [3]. Therefore, credit card fraud detection methods need constant innovation. In this study, we evaluate two advanced data mining approaches, support vector machines and random forests, together with the well-known logistic regression, as part of an attempt to better detect (and thus control and prosecute) credit card fraud. The study is based on real-life data of transactions from an international credit card operation.

Statistical fraud detection methods have been divided into two broad categories: *supervised* and *unsupervised* [3]. In supervised fraud detection methods, models are estimated based on the samples of

fraudulent and legitimate transactions, to classify new transactions as fraudulent or legitimate. In unsupervised fraud detection, outliers or unusual transactions are identified as potential cases of fraudulent transactions. Both these fraud detection methods predict the probability of fraud in any given transaction.

Predictive models for credit card fraud detection are in active use in practice [21]. Considering the profusion of data mining techniques and applications in recent years, however, there have been relatively few reported studies of data mining for credit card fraud detection. Among these, most papers have examined neural networks [1,5,19,22], not surprising, given their popularity in the 1990s. A summary of these is given in [28], which reviews analytic techniques for general fraud detection, including credit card fraud. Other techniques reported for credit card fraud detection include case based reasoning [48] and more recently, hidden Markov models [45]. A recent paper [49] evaluates several techniques, including support vector machines and random forests for predicting credit card fraud. Their study focuses on the impact of aggregating transaction level data on fraud prediction performance. It examines aggregation over different time periods on two real-life datasets and finds that aggregation can be advantageous, with aggregation period length being an important factor. Aggregation was found to be especially effective with random forests. Random forests were noted to show better performance in relation to the other techniques, though logistic regression and support vector machines also performed well.

Support vector machines and random forests are sophisticated data mining techniques which have been noted in recent years to show superior performance across different applications [30,38,46,49]. The choice of these two techniques, together with logistic regression, for this study is based on their accessibility for practitioners, ease of use, and noted performance advantages in the literature. SVMs are statistical learning techniques, with strong

* Corresponding author. Tel.: +1 312 996 8794; fax: +1 312 413 0385.

E-mail addresses: sidb@uic.edu (S. Bhattacharyya), sanjeev.jha@unh.edu (S. Jha), ktharakunnel@millikin.edu (K. Tharakunnel), westland@uic.edu (J.C. Westland).

¹ Tel.: +1 603 862 0314; fax: +1 603 862 3383.

² Tel.: +1 312 996 2323; fax: +1 312 413 0385.

Linda Delamaire (UK), Hussein Abdou (UK), John Pointon (UK)

Credit card fraud and detection techniques: a review

Abstract

Fraud is one of the major ethical issues in the credit card industry. The main aims are, firstly, to identify the different types of credit card fraud, and, secondly, to review alternative techniques that have been used in fraud detection. The sub-aim is to present, compare and analyze recently published findings in credit card fraud detection. This article defines common terms in credit card fraud and highlights key statistics and figures in this field. Depending on the type of fraud faced by banks or credit card companies, various measures can be adopted and implemented. The proposals made in this paper are likely to have beneficial attributes in terms of cost savings and time efficiency. The significance of the application of the techniques reviewed here is in the minimization of credit card fraud. Yet there are still ethical issues when genuine credit card customers are misclassified as fraudulent.

Keywords: credit card fraud, detection techniques, credit bureaux, data mining techniques.

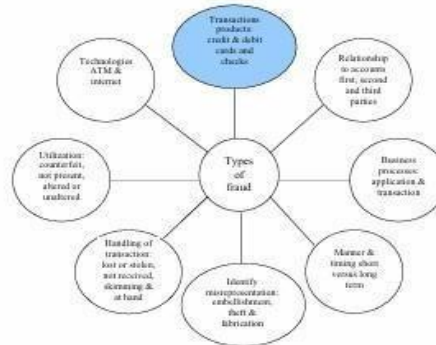
JEL Classification: C49, G21, G24, K42.

Introduction

For some time, there has been a strong interest in the ethics of banking (Molyneaux, 2007; George, 1992), as well as the moral complexity of fraudulent behavior (Clarke, 1994). Fraud means obtaining services/goods and/or money by unethical means, and is a growing problem all over the world nowadays. Fraud deals with cases involving criminal purposes that, mostly, are difficult to identify. Credit cards are one of the most famous targets of fraud but not the only one; fraud can occur with any type of credit products, such as

personal loans, home loans, and retail. Furthermore, the face of fraud has changed dramatically during the last few decades as technologies have changed and developed. A critical task to help businesses, and financial institutions including banks is to take steps to prevent fraud and to deal with it efficiently and effectively, when it does happen (Anderson, 2007).

Anderson (2007) has identified and explained the different types of fraud, which are as many and varied as the financial institution's products and technologies, as shown in Figure 1.



Source: own figure, following Anderson's classification (2007).

Fig. 1. Types of fraud

The main aims are, firstly, to identify the different types of credit card fraud, and, secondly, to review alternative techniques that have been used in fraud

detection. The focus here is in Europe, and so ethical issues arising from other cultures are not taken into account; but for a discussion of these the reader is referred to Chepaitis (1997) and Gichure (2000). Indeed, transaction products, including credit cards, are the most vulnerable to fraud. On the other hand, other products such as personal loans and retail are also at risk, and have serious ethical

© Linda Delamaire, Hussein Abdou, John Pointon, 2009.
Acknowledgement: The authors wish to thank Peter Sinclair for helpful comments. And very special thanks for Pago e-Transaction Services GmbH, for providing us with all the Figures used in the current paper.

A Geolocation Databases Study

Yuval Shavitt, *Senior Member, IEEE*, and Noa Zilberman

Abstract—The geographical location of Internet IP addresses is important for academic research, commercial and homeland security applications. Thus, both commercial and academic databases and tools are available for mapping IP addresses to geographic locations. Evaluating the accuracy of these mapping services is complex since obtaining diverse large scale ground truth is very hard. In this work we evaluate mapping services using an algorithm that groups IP addresses to PoPs, based on structure and delay. This way we are able to group close to 100,000 IP addresses world wide into groups that are known to share a geo-location with high confidence. We provide insight into the strength and weaknesses of IP geolocation databases, and discuss their accuracy and encountered anomalies.

Index Terms—Geographic Information Systems, Web and internet services, Internet topology.

I. INTRODUCTION

IN THE RECENT years, geolocation services have become a necessity in many fields and for many applications. While the end user is usually not aware of it, many websites visited every day use geolocation information for targeted localized advertising, localized content (such as local news and weather), and compliance with local law.

The last decade presented a new threat to the world: cyber terrorism. Cyber terrorism and warfare targets communication networks as well as important infrastructure facilities, and thus threatens to cause havoc through online attacks. Finding and blocking such cyber attacks is in a high priority for national security forces, and IP geolocation can help by providing geographic information about the attacker hosts. The DHS cyber security center [19] classified geolocation research to be in the field of situational understanding and attack attribution, with the intent to identify attackers. The DHS also comments that geolocation improves visualization, thus simplifies large-scale data analysis. A patent filed by the NSA [22] notes that geolocation can be used to monitor remote access and prevent login using stolen passwords or login ID. It can only be speculated that military and government based agencies use geolocation techniques to detect the source of activity on terrorist related websites as well as trying to track down enemy communication centers.

Perhaps the most highlighted purpose of geolocation information is for fraud prevention and various means of security. Banking, trading, and almost any other type of business that handles online money transactions are exposed to phishing attempts as well as other schemes. Criminals try to break into user accounts to transfer money, manipulate stocks, make purchases and other illegal activities. Geolocation information provides means to reduce the risk, for example by blocking

users from certain high-risk countries and cross-referencing user expected and actual locations.

The IETF has also commenced in defining standards for geolocation and emergency calling through IETF GEOPRIV working group [23], which discusses internet geolocation standards and privacy protection for geolocation. Some examples are DHCP location, as in RFC3825 and RFC4776, and defining protocols for discovering the local location information server [42]. Even common emergency services, such as dispatching emergency responders to the location of emergency use it.

Geolocation information is also important in many research fields. It improves internet mapping and characterization, as it ties the internet graph to actual node positions, and allows exploring new aspects of the network that are otherwise uncovered, such as the effect of ISP location on its services and types of relationships with other service providers.

Many previous papers from various fields have discussed the usage of geolocation information in day-to-day applications ([41], [12], [26] and more). However, not many works have focused on the accuracy of geolocation databases. In 2008, Siwipersad *et al.* [38] examined the accuracy of Maxmind [30] and IP2Location [18]. They assessed their resolution and confidence area and concluded that their resolution is too coarse and that active measurements provide a more accurate alternative. Gueye *et al.* [16] investigated the imprecision of relying on the location of blocks of IP addresses to locate Internet hosts and concluded that geolocation information coming from exhaustive tabulation may contain an implicit imprecision. Muir and Oorschot [32] conducted a survey of geolocation techniques used by geolocation databases and examined means for evasion/circumvention from a security standpoint.

Improving location accuracy by measurements has been addressed by several works in the recent years. IP2Geo [33] was one of the first to suggest a measurement-based approach to approximate the geographical distance of network hosts. A more mature approach is constraint based geolocation [17], which uses several delay constraints to infer the location of a network host by a triangulation-like method. Later works, such as Octant [43] use a geometric approach to localize a node within a 22 mile radius. Katz-Bassett *et al.* [25] suggested topology based geolocation using link delay to improve the location of nodes. Yoshida *et al.* [44] used end-to-end communication delay measurements to infer PoP level topology between thirteen cities in Japan. Laki *et al.* [27] increased geolocation accuracy by decomposing the overall path-wise packet delay to link-wise components and were thus able to approximate the overall propagation delay along the measurement path. Eriksson *et al.* [6] apply a learning based approach to improve geolocation. They reduce IP geolocation

Manuscript received 15 November 2010; revised 1 May 2011.
Y. Shavitt and N. Zilberman are with the School of Electrical Engineering,
Tel Aviv University, Israel (e-mail: shavitt.noa@eng.tau.ac.il).
Digital Object Identifier 10.1109/JSAC.2011.111214.

Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine

Apapan Pumsirirat, Liu Yan
School of Software Engineering, Tongji University
Shanghai, China

Abstract—Frauds have no constant patterns. They always change their behavior; so, we need to use an unsupervised learning. Fraudsters learn about new technology that allows them to execute frauds through online transactions. Fraudsters assume the regular behavior of consumers, and fraud patterns change fast. So, fraud detection systems need to detect online transactions by using unsupervised learning, because some fraudsters commit frauds once through online mediums and then switch to other techniques. This paper aims to 1) focus on fraud cases that cannot be detected based on previous history or supervised learning, 2) create a model of deep Auto-encoder and restricted Boltzmann machine (RBM) that can reconstruct normal transactions to find anomalies from normal patterns. The proposed deep learning based on auto-encoder (AE) is an unsupervised learning algorithm that applies backpropagation by setting the inputs equal to the outputs. The RBM has two layers, the input layer (visible) and hidden layer. In this research, we use the Tensorflow library from Google to implement AE, RBM, and H2O by using deep learning. The results show the mean squared error, root mean squared error, and area under curve.

Keywords—Credit card; fraud detection; deep learning; unsupervised learning; auto-encoder; restricted Boltzmann machine; Tensorflow

I. INTRODUCTION

Fraud detection in online shopping systems is the hottest topic nowadays. Fraud investigators, banking systems, and electronic payment systems such as PayPal must have an efficient and complex fraud detection system to prevent fraud activities that change rapidly. According to a CyberSource report from 2017, the present fraud loss by order channel, that is, the percentage of fraud loss in their web store was 74 percent and 49 percent in their mobile channels [1]. Based on this information, the lesson is to determine anomalies across patterns of fraud behavior that have undergone change relative to the past.

A good fraud detection system should be able to identify the fraud transaction accurately and should make the detection possible in real-time transactions. Fraud detection can be divided into two groups: anomaly detection and misuse detection [2]. Anomaly detection systems bring normal transaction to be trained and use techniques to determine novel frauds. Conversely, a misuse fraud detection system uses the labeled transaction as normal or fraud transaction to be trained in the database history. So, this misuse detection system entails

a system of supervised learning and anomaly detection system a system of unsupervised learning. What is the difference between supervised learning and unsupervised learning? The answer is that supervised learning studies labeled datasets. They use labeled datasets to train and to render it accurate by changing the parameters of the learning rate. After that, they apply parameters of learning rate to the dataset, the techniques that implement supervised learning such as multilayer-perceptron (MLP) to build the model based on the history of the database. This supervised learning has a disadvantage, since if new fraud transactions happen that do not match with the records of the database, then this transaction will be considered genuine. While, unsupervised learning acquires information from new transactions and finds anomalous patterns from new transaction. This unsupervised learning is more difficult than supervised learning, because we have to use appropriate techniques to detect anomalous behavior.

Neural networks were introduced to detect credit card frauds in the past. Now, we focus on deep learning that is a subfield of machine learning (ML). Based on deep learning in the first period, they use deep learning to know about an image's processing. For example, Facebook uses deep learning in the function to tag people and to know who the person is for subsequent reference. Further, deep learning in neural networks have many algorithms for use in fraud detection, but in this paper, we selected the AE and RBM to detect whether normal transaction of datasets qualified as novel frauds. We believe that some normal transaction in datasets that were labeled as fraud also show suspicious transaction behavior. So, in this paper we focus on unsupervised learning.

In this paper, we use three datasets in these experiments; these datasets are the German, Australian, and European datasets [4], [3], [18]. The first dataset is German, provided by Professor Dr. Hans Hofman [4]. There are twenty attributes that describe the capability, such as credit history, purpose to use credit card, credit amount, job, among others. The German dataset were 1000 instances. The second dataset is from Australia. [3] The attributes' names and values in this dataset have been changed to meaningless symbols to protect the confidentiality of the data. There were 690 instances. The last dataset was from a European cardholder from September 2013. This dataset shows the transaction that occurred in two days with 284, 807 transactions. There were 31 features in this dataset. The 28 features, such as V1, V28 is a numerical input variable result of PCA transformation. Other 3 feature that do

S