Fraud Detection System using Bayesian Networks

Bachelor in Technology
In
Computer Science and Engineering

***S*ubmitted By:**

Deep Senchowa (CSE-09/20)
Anushmit Tamuli (CSE- 03/20)
Joy Anupol Neog (CSE-16/20)

Submitted to
Dr. Abhijit Boruah
Assistant Professor Department of Computer Science and Engineering
DUIET, Dibrugarh University

Department of Computer Science and Engineering
Dibrugarh University Institute of Engineering and Technology
Dibrugarh University
Dibrugarh-786004
May, 2024

Contents

1 Introduction

Transactional fraud detection is a critical task for businesses and financial institutions to safeguard their assets and maintain trust with customers. Traditional rule-based systems may not be sufficient to detect sophisticated fraudulent activities. Bayesian Networks offer a probabilistic approach to analyzing complex data and inferring the likelihood of fraud. In this project, we develop a fraud detection system using a Bayesian Network implemented in C programming language.

2 Problem Statement

The goal of this project is to design and implement a fraud detection system that analyzes financial transaction data to identify potentially fraudulent activities. The system utilizes probabilistic reasoning techniques to infer the likelihood of fraud based on various features such as transaction amount, location, account age, frequency, and time.

3 System Architecture

The system architecture consists of the following components:

- Bayesian Network: A graphical model that represents the probabilistic relationships between different variables.
- CPT Initialization: Conditional Probability Tables (CPTs) are initialized to encode the probabilities of different states for each node in the Bayesian Network.
- Inference Engine: Performs probabilistic inference using the evidence provided to calculate the probability of fraud.
- User Interface: Allows users to input transaction data and view the probability of fraud.

4 Implementation

The implementation is done in C programming language and consists of the following modules:

- Main Module: Initializes the CPTs, reads input data from the user, performs inference, and displays the probability of fraud.
- CPT Initialization: Initializes the CPTs with predefined probabilities for different states of each node.

- Inference Function: Performs probabilistic inference using the evidence provided by the user.
- Input Module: Reads input data (transaction amount, location, account age, frequency, time) from the user.

5 Bayesian Network Structure

The Bayesian Network structure consists of five nodes representing different features of financial transactions:

1. Transaction Amount
2. Transaction Location
3. Account Age
4. Transaction Frequency
5. Transaction Time

6 Conditional Probability Tables (CPTs)

CPTs encode the conditional probabilities of different states for each node given the states of its parent nodes. The probabilities are pre-defined based on domain knowledge or historical data.

Table 1: Conditional Probability Table for Transactional Fraud Detection

Feature	**Values**	**Normal**	**Fraud**
Transaction Amount	[0, 1000]	0.8	0.1
	[1000, 5000]	0.1	0.2
	[5000, 10000]	0.05	0.3
	[10000, 50000]	0.05	0.4
Transaction Location	Local	0.7	0.1
	International	0.3	0.9
Account Age	[0, 1 year]	0.5	0.3
	[1, 5 years]	0.3	0.4
	[5, 10 years]	0.1	0.2
	[10, 20 years]	0.1	0.1
Transaction Frequency	[1, 5 transactions]	0.8	0.2
	[5, 10 transactions]	0.1	0.3
	[10, 20 transactions]	0.05	0.3
	[20, 50 transactions]	0.05	0.2
Transaction Time	[9:00 AM, 5:00 PM]	0.8	0.1
	[5:00 PM, 9:00 PM]	0.1	0.2
	[9:00 PM, 12:00 AM]	0.05	0.3
	[12:00 AM, 9:00 AM]	0.05	0.4

7 Mathematical Explanation

7.1 Initialization

prob_fraud[0] and prob_fraud[1] are initialized to 1.0. This represents a uniform prior belief about the states of the target node (fraud or not fraud).

7.2 Evidence Incorporation

The code iterates over each node. For nodes with observed evidence (evidence[node] $\neq -1$), it updates prob_fraud by multiplying with the corresponding conditional probabilities from the CPT.

Mathematically, if node i has observed evidence e_i:

$$\text{prob\_fraud}[s] \leftarrow \text{prob\_fraud}[s] \times \text{cpt}[i][e_i][s]$$

for each state s (0 and 1).

7.3 Marginalization

For nodes without observed evidence (evidence[node] $== -1$), the code calculates the marginal probability by summing over all possible evidence values weighted by the current prob_fraud.

Mathematically, for node i without observed evidence:

$$\text{marginal}[s] = \sum_{e=0}^{3} \text{cpt}[i][e][s] \times \text{prob\_fraud}[s]$$

After computing the marginal for both states:

$$\text{prob\_fraud}[s] \leftarrow \text{marginal}[s]$$

7.4 Normalization

After processing all nodes, the final probabilities in prob_fraud are normalized to ensure they sum to 1.

Mathematically:

$$\text{sum} = \text{prob\_fraud}[0] + \text{prob\_fraud}[1]$$

$$\text{prob\_fraud}[0] \leftarrow \frac{\text{prob\_fraud}[0]}{\text{sum}}$$

$$\text{prob\_fraud}[1] \leftarrow \frac{\text{prob\_fraud}[1]}{\text{sum}}$$

7.5 Result

The function returns prob_fraud[1], the normalized probability of the target event being in state 1 (fraud).

8 User Interface

The user interface allows users to input transaction data through a series of prompts. The system then calculates the probability of fraud based on the provided data and displays the result to the user.

9 Conclusion

In conclusion, the developed fraud detection system utilizes Bayesian Networks and probabilistic reasoning techniques to analyze financial transaction data and infer the likelihood of fraud. The system provides a robust and efficient approach to identifying potentially fraudulent activities, thereby helping businesses and financial institutions mitigate risks and protect their assets. Future enhancements may include incorporating real-time data processing and integrating machine learning algorithms for improved accuracy and performance.