Documentation: Anomaly Detection in Financial transaction

Overview:

Implement advanced AI-driven anomaly detection algorithms to identify a typical-patterns in financial transaction data. This includes employing machine learning models such as Isolation Forest or Autoencoders to detect potential fraudulent activities or errors, enhancing data integrity and security.

Introduction:

Anomaly detection refers to process of identifying data points, transactions or patterns that deviate from the standard norm. Early identification of anomalies in financial transactions is essential for preventing fraud, ensuring compliance, enhancing operational efficiency, and managing reputation.

Fraud is a major and increasing concern in the financial industry, with more and more advanced attack vectors leading to huge financial losses. This research emphasizes identity theft, especially in online and mobile payment systems, where fraudsters take advantage of weakness in payment systems to carry out unauthorized transactions.

The main aim is to detect and prevent online attacks by external parties. As these attackers tend to circumvent conventional account access controls, detection based only on login behavior is not adequate. Rather, payment transaction behavior analysis offers a better detection mechanism.

To detect such anomalies, machine learning algorithms are used, which try to identify suspicious patterns that may represent potential fraud. The performance of these algorithms relies heavily on both their computational power and the quality of the input data.

Role of AI in detecting financial anomalies:

Heart of AI- based fraud detection is machine learning. The two principle methods to train AI models to differentiate between genuine and fake transactions:

Supervised Learning:

Traditional AI can detect know patterns of fraud as it is trained on labeled sets of past fraudulent and legitimate transactions.

Unsupervised learning:

It is vital to detect new and develop fraud strategies because it can analyze transaction elements.

Advantages:

- Improved accuracy: AI can detect fine patterns of fraud that humans don’t notice.

- Real-time detection: AI reduces financial losses by detecting suspicious transactions immediately.
- Reduced false positives: AI enhances detection accuracy, reducing unnecessary transaction blocks and annoying customers
- Scalability: AI frameworks are ideal and large financial institutions since they have capacity to handle vast transactions
- Continuous Learning: AI models learn continuously about new schemes and stay updated against new threats.

Process or methodology:

Define Scope

Start by precisely defining what kinds of anomalies your detection system must detect, like fraudulent activity, data entry mistakes, or abnormal patterns. Also, identify which data sources, like transaction logs, accounting records, or customer databases, will be tapped to provide end-to-end coverage. Actually, using anomaly management software companies are able to recognize the various kinds of discrepancies and errors.

Collect Data

Acquire all the financial transaction data from the sources identified, making sure that the data is complete and latest. Once collected, sanitize and structure the data into a neat format to enable accurate and efficient analysis. If companies are using anomaly detection software, the same should easily integrate with sources of data

Data Preprocessing

Normalize the gathered data to make it consistent, like making transaction amounts and timestamps standard. Resolve any missing values by imputing or deleting them to increase the accuracy and dependability of the anomaly detection system.

Feature Selection

Find out the most essential features most directly applicable for anomalous detection, including transaction sizes, timestamps, transaction types, and customer IDs. These should be emphasized for the model in order to distinctly differentiate normal transactions from abnormal transactions.

Model selection

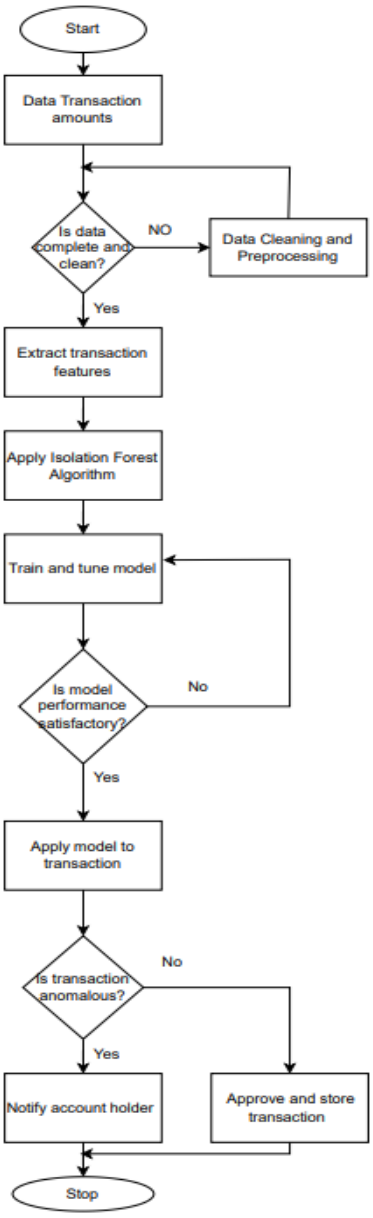
Select a suitable model depending on the complexity and nature of the data, whether it is a statistical algorithm, machine learning method, or deep learning approach. The model chosen must be able to represent patterns well and identify anomalies in the financial data.

Training & validation

Train the chosen model with past financial records so that it can learn and recognize patterns of normal and abnormal transactions. Test the performance of the model on an independent dataset to verify that it generalizes well and correctly identifies anomalies in real-world applications.

Deployment

Once trained and tested, deploy the model in a production environment where it will process real-time transaction data. Monitor the performance of the model continuously, making updates and changes as required to ensure its accuracy and efficiency in identifying anomalies.



Flowchart for Anomaly detection in financial transaction

Isolation Forest

Isolation forest algorithm is used to detect abnormalities, or outliers within a dataset. It is good at finding these anomalies in large amounts of data. It has become renowned in many different industries such as cybersecurity, finance and medicine as a fast and reliable anomaly detection tool.

Isolation for Finding Anomalies

Similar to Random Forests, Isolation Forests (IF) are built with decision trees. Moreover, this model is unsupervised since it does not have pre-defined labels.

Randomly subsampled data is handled in a tree framework with randomly selected features in an isolation forest. Because more cuts were needed to isolate the samples that went further into the tree, they are less probable to be abnormalities.

Isolation Forest working:

1. Random Partitioning

Random Partitioning is a building block of Isolation Forest. This is how it operates:

- Random Feature Selection: The process begins with selecting a feature from the dataset at random.
- Random Splitting: After selecting a feature, a random value from within the range of values of that feature is used as a splitting threshold. This splits the data into two subsets: one containing data points with values less than or equal to the threshold, and the other containing data points with values higher than the threshold.
- Recursive Partitioning: This recursive feature selection by chance and partition splitting is carried out repeatedly until there are isolated single data points into separate partitions or a predetermined max depth has been achieved.

2. Isolation Path

Isolation Path gauges the separation or "anomaliness" of a data point within the tree

Path Length: The isolation path of a data point is defined by how many splits or partitions it takes to isolate that point in a tree. Anomalies, being less representative of the general data distribution, tend to take fewer splits to be isolated than normal data points. This is due to the fact that anomalies are quite different from normal instances and tend to fall into smaller, isolated partitions.

3. Ensemble of Trees

Isolation Forest does not use a single tree but constructs an ensemble of isolation trees:

- Multiple Trees: The algorithm creates an independent specified number of isolation trees. Each tree splits the data randomly, so each data point has different isolation paths in the ensemble of trees.
- Group Grading: Anomalies are detected by considering the isolation path through all the trees in the ensemble. Data points with a shorter isolation path through several trees are anomalies as they need to be isolated in fewer partitions.

4. Scoring of the Anomaly

Mean Separation Distance: For every data point, the algorithm calculates an anomaly score by using the average path length or separation distance over all the trees in the ensemble. This is a measure of how far from normal a data point is. Lower anomaly score means that a data point is more isolated (and hence more likely to be an outlier).

5. Classification

Thresholding: In order to separate normal from anomalous points, a threshold is applied on the anomaly scores. Points that have anomaly scores greater than the threshold are categorized as anomalies and those less than the threshold as normal.

Tools required:

Jupiter Notebook

Packages required:

Pandas, NumPy

Seaborn, Scikit-learn