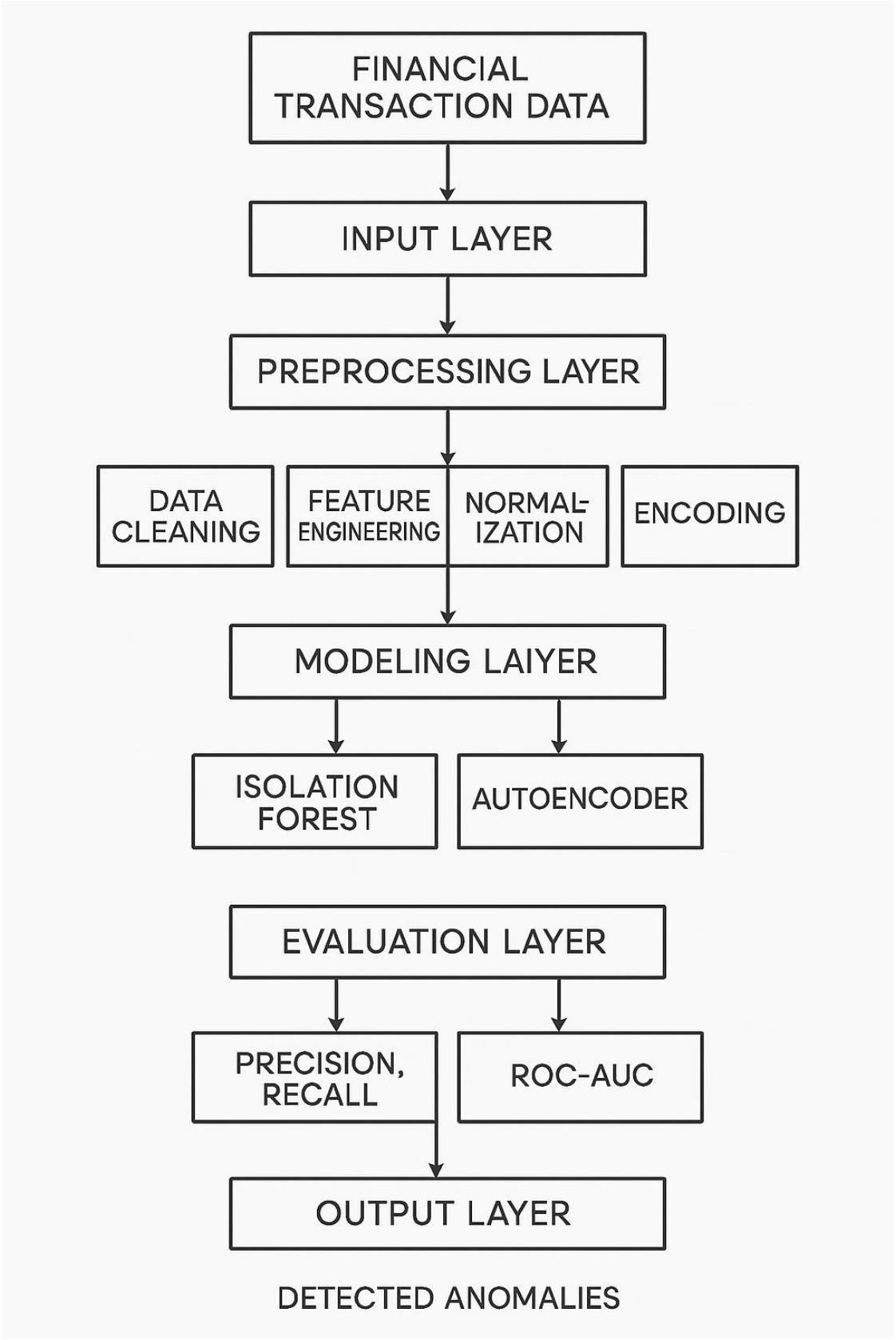
Design


FINANCIAL TRANSACTION DATA
INPUT LAYER
PREPROCESSING LAYER
DATA CLEANING
FEATURE ENGINEERING
NORMAL-IZATION
ENCODING
MODELING LAIYER
ISOLATION FOREST
AUTOENCODER
EVALUATION LAYER
PRECISION, RECALL
ROC-AUC
OUTPUT LAYER
DETECTED ANOMALIES

1.1 System Architecture

The system is composed of five key components, organized in a sequential pipeline:

1. **Input Layer**: Accepts raw financial transaction data from structured sources such as CSV files or databases.
2. **Preprocessing Layer**: Performs data cleaning, feature engineering, normalization, and encoding to prepare inputs for modelling.
3. **Modelling Layer**: Includes two unsupervised learning techniques—**Isolation Forest** and **Autoencoder neural networks**—to detect deviations from typical transaction behaviour.
4. **Evaluation Layer**: Assesses model performance using metrics such as **Precision**, **Recall**, and **ROC-AUC**, with thresholds defined based on statistical analysis or percentiles.
5. **Output Layer**: Flags anomalous transactions and optionally presents them through a visualization interface for further review or alerting.

1.2 Data Design and Preprocessing

The input data typically includes attributes such as transaction amount, timestamp, transaction type, and account identifiers. Preprocessing steps are crucial to enhance model performance and include:

- Handling missing values and outliers
- Encoding categorical features
- Normalizing continuous variables
- Generating derived features (e.g., transaction frequency, rolling statistics)

1.3 Model Design

- **Isolation Forest**: A tree-based ensemble method that isolates anomalies by randomly partitioning the feature space. Its low computational complexity makes it well-suited for large datasets.
- **Autoencoder**: A neural network trained to reconstruct normal transaction patterns. Transactions with high reconstruction errors are considered anomalous. The

architecture consists of multiple encoding and decoding layers with a Mean Squared Error (MSE) loss function.

Both models are trained on normal data distributions without requiring Labeled anomalies. Anomaly thresholds are selected using validation heuristics or statistical cutoffs.

8.4 Tools and Environment

The system is implemented using:

- **Python** as the core programming language
- **Pandas** and **NumPy** for data manipulation
- **Scikit-learn** for implementing Isolation Forest
- **TensorFlow/Keras** for Autoencoder model design
- **Matplotlib** and **Seaborn** for visual analytics
- **Jupyter Notebook** for interactive development and experimentation