Transactly uses a **layered fraud detection architecture**, starting with a fast, interpretable **XGBoost model** for initial risk scoring, followed by a richer **Wide & Deep Neural Network** that combines categorical and continuous features to predict fraudulent transactions with greater accuracy.

**1. XGBoost Fraud Detection Model**

**Purpose**

Provides a fast, explainable baseline fraud risk score based on structured transaction features. It is the first filter in the multi-model fraud detection pipeline.

**🧠 Model Type**

* Gradient Boosted Decision Trees (XGBoost)

**🧾 Input Features**

The model uses a mix of:

* **Real-time transaction features**
* **Historical user behavior metrics**

REAL\_TIME\_FEATURES = [

'amount', 'ip\_risk\_score', 'country\_code', 'device\_age\_days', 'hour\_of\_day',

'is\_vpn', 'distance\_from\_home\_km', 'category\_id', 'cvv\_attempts',

'session\_duration\_sec', 'is\_new\_device', 'payment\_method', 'shipping\_billing\_match'

]

**📦 Model Training**

* Framework: xgboost==3.0.2
* Evaluation Metric: AUC-ROC
* Hyperparameters:

{

"max\_depth": 5,

"eta": 0.1,

"objective": "binary:logistic",

"eval\_metric": "auc",

"subsample": 0.8,

"colsample\_bytree": 0.8

}

**Output**

* Single float score: fraud\_risk\_score ∈ [0, 1]
* Interpretable via SHAP or built-in feature importance.

Used for:

* Early-stage filtering
* Input into the Wide & Deep model as a feature

2. Wide & Deep Neural Network (FraudScoreNet)

### ****Purpose****

#### Combines memorization (wide features) and generalization (deep features) to detect fraud patterns in both short-term and l **Wide Features** (payload["wide"])

* Passed as a Pandas DataFrame
* Real-time transaction features, categorical + continuous

#### **Deep Features** (payload["deep"])

* Passed as a 2D NumPy array
* Standardized using StandardScaler
* Includes engineered features from past activity, device usage, embedding vectors, and the fraud\_risk\_score from XGBoost

### 📦 ****Model Training****

* Framework: PyTorch
* Loss: Binary Cross Entropy
* Optimizer: Adam
* Evaluation Metric: ROC-AUC
* Batch size: 64
* Epochs: 20

### ****Model Artifacts****

Stored in model.tar.gz with:

* wd\_model.pth – Trained PyTorch model weights
* metadata.pkl – Dictionary with:
  + real\_time\_features: list of wide feature names
  + user\_embeddings: list of deep features
  + scaler: fitted StandardScaler object for deep features

Features:-

# Feature lists (match those used in preprocess.py)

USER\_EMBEDDINGS = [

'recency\_days\_since\_last\_txn', 'recency\_days\_since\_last\_login',

'recency\_days\_since\_last\_cart\_abandonment', 'recency\_days\_since\_last\_support\_contact',

'recency\_days\_since\_last\_refund', 'frequency\_txns\_per\_week',

'frequency\_sessions\_per\_day', 'frequency\_logins\_per\_month',

'frequency\_failed\_logins\_per\_month', 'frequency\_searches\_per\_session',

'monetary\_avg\_order\_value\_30d', 'monetary\_avg\_order\_value\_90d',

'monetary\_total\_spend\_30d', 'monetary\_total\_spend\_90d',

'monetary\_spend\_volatility', 'monetary\_median\_order\_value',

'monetary\_high\_ticket\_orders', 'monetary\_micro\_txns',

'monetary\_coupon\_usage\_rate', 'monetary\_gift\_card\_redemption\_rate',

'cat\_electronics', 'cat\_grocery', 'cat\_fashion', 'cat\_home\_kitchen',

'cat\_books\_media', 'cat\_health\_beauty', 'cat\_sports\_outdoors',

'cat\_toys\_games', 'cat\_automotive', 'cat\_prime\_video', 'cat\_prime\_music',

'cat\_kindle', 'cat\_aws\_api', 'cat\_aws\_storage', 'cat\_amazon\_fresh',

'cat\_audible', 'cat\_pharmacy', 'cat\_amazon\_pay', 'cat\_amazon\_photos',

'temp\_hour\_00', 'temp\_hour\_01', 'temp\_hour\_02',

'temp\_hour\_03', 'temp\_hour\_04', 'temp\_hour\_05', 'temp\_hour\_06',

'temp\_hour\_07', 'temp\_hour\_08', 'temp\_hour\_09', 'temp\_hour\_10',

'temp\_hour\_11', 'temp\_hour\_12', 'temp\_hour\_13', 'temp\_hour\_14',

'temp\_hour\_15', 'temp\_hour\_16', 'temp\_hour\_17', 'temp\_hour\_18',

'temp\_hour\_19', 'temp\_hour\_20', 'temp\_hour\_21', 'temp\_hour\_22',

'temp\_hour\_23', 'temp\_day\_mon', 'temp\_day\_tue', 'temp\_day\_wed',

'temp\_day\_thu', 'temp\_day\_fri', 'temp\_day\_sat', 'temp\_day\_sun',

'device\_pct\_mobile', 'device\_pct\_desktop', 'device\_pct\_tablet',

'device\_distinct\_count', 'device\_entropy', 'device\_pct\_android',

'device\_pct\_ios', 'device\_pct\_windows', 'device\_pct\_macos',

'device\_pct\_headless', 'network\_pct\_vpn', 'network\_avg\_asn\_score',

'network\_pct\_tor', 'network\_geo\_mismatch', 'network\_distinct\_ips',

'network\_pct\_new\_ips', 'geo\_city\_count', 'geo\_country\_count',

'geo\_avg\_distance', 'geo\_max\_distance\_24h', 'geo\_home\_mismatch',

'geo\_pct\_top\_cities', 'geo\_pct\_overseas', 'geo\_cross\_region\_hops',

'returns\_pct\_returned', 'returns\_avg\_return\_time',

'returns\_pct\_refunds\_approved', 'returns\_chargebacks',

'returns\_pct\_partial\_refunds', 'support\_tickets',

'support\_resolution\_time', 'support\_pct\_escalated',

'support\_negative\_feedback', 'support\_pct\_reopened',

'engagement\_avg\_pages', 'engagement\_ctr', 'engagement\_pct\_video',

'engagement\_pct\_audio', 'engagement\_pct\_search',

'engagement\_search\_abandon', 'engagement\_wishlist\_adds',

'engagement\_cart\_adds', 'engagement\_pct\_buy\_again',

'engagement\_pct\_social\_share', 'behavior\_std\_amount',

'behavior\_std\_time', 'behavior\_std\_sessions', 'behavior\_gini\_spend',

'behavior\_temporal\_churn', 'behavior\_burstiness',

'behavior\_device\_stability', 'behavior\_address\_stability'

] # Same as before, truncated for brevity

CATEGORICAL = ['country\_code', 'payment\_method', 'category\_id']

REAL\_TIME\_FEATURES = [

'amount', 'ip\_risk\_score', 'country\_code', 'device\_age\_days', 'hour\_of\_day',

'is\_vpn', 'distance\_from\_home\_km', 'category\_id', 'cvv\_attempts',

'session\_duration\_sec', 'is\_new\_device', 'payment\_method', 'shipping\_billing\_match'

]