Feature Engineering Techniques in Fraud Detection

Fraud detection, particularly in domains like banking, e-commerce, and insurance, relies heavily on the quality of features fed into machine learning models. Since fraudulent activities often mimic legitimate ones, carefully crafted features can significantly enhance model performance by uncovering hidden patterns. This writeup explores key **feature engineering techniques** specifically tuned for fraud detection use cases.

🧩 Types of Features in Fraud Detection

1. Transactional Features

These are directly extracted from each transaction.

Feature	Description
`amount`	Monetary value of transaction
`merchant`	Merchant or vendor ID
`category`	Type of transaction (e.g., electronics, food)
`timestamp`	Time of the transaction

Transformations:

- Log-transform the `amount` to reduce skew.
- One-hot encode or target-encode `category`.

3. Historical Aggregates

Use rolling windows to detect deviation from normal behavior:

- Rolling average transaction amount over N days
- Standard deviation or z-score of past spending
- Frequency of visiting same merchant

```
df['rolling_avg_7d'] = df.groupby('user_id')['amount'].transform(lambd
```

5. Temporal Features

Fraud tends to follow abnormal timing patterns:

- Hour of day or day of week
- Night-time or weekend transactions
- Time gaps between successive transactions

```
df['hour'] = pd.to_datetime(df['timestamp']).dt.hour
df['is_night'] = df['hour'].apply(lambda x: x < 6 or x > 22)
```

7. Entity Linking Features

- Shared card number, device ID, or IP address across multiple users
- Transactions from different users using the same device
- Graph-based clustering for fraud rings

⍰ Best Practices

- Normalize or scale numeric features
- Handle class imbalance (SMOTE, undersampling)
- Avoid data leakage (e.g., future info)
- Use cross-validation across time splits
- Monitor concept drift in production

🔍 Tools and Libraries

- **Pandas** / **Polars** – Data manipulation
- **scikit-learn** – Preprocessing, pipelines
- **XGBoost/LightGBM** – Tree-based models
- **PyOD** – Outlier detection
- **GraphFrames** – Fraud ring detection in Spark

🎯 Conclusion

Feature engineering is central to fraud detection. Unlike some ML applications, off-the-shelf models perform poorly without customized domain features. A deep understanding of both domain context and user behavior is key to designing high-value features that surface subtle fraud signals.

In production systems, feature pipelines must be efficient, scalable, and real-time, especially when used with streaming platforms like Apache Kafka and Spark Streaming.