



# Explainable AI (XAI) in Deep Learning Models for Credit Card Fraud Detection

Thong Minh Lai

Student ID: U2259343

Department of Computer Science  
University of Huddersfield

Dr. Hyunkook Lee  
Professor and Supervisor

April 2025

# Contents

---

1. Introduction

2. Background

3. Methods

4. Results

5. Discussion

6. Conclusions

# Introduction

---

# Introduction

---

- Credit card fraud is a massive problem, with losses in the UK alone reaching £1.3 billion in 2021.<sup>1</sup>
- Deep Learning (DL) models are powerful for fraud detection, but their “black box” nature makes them hard to trust in high-stakes environments.
- My project tackles this by integrating Explainable AI (XAI) techniques into state-of-the-art DL models, focusing on local interpretability.
- All experiments are conducted on the Sparkov synthetic dataset, which is ideal for benchmarking fraud detection systems.<sup>2</sup>

## Project Aims

- Compare multiple deep learning architectures for fraud detection.
- Integrate and evaluate XAI methods (SHAP, LIME, Anchors) for local explanations.
- Develop robust evaluation metrics for both accuracy and interpretability.

---

<sup>1</sup>UK Finance, 2022.

<sup>2</sup>Grover et al., 2023.

# Background

---

# Background and Motivation

---

- Traditional fraud detection relied on hand-crafted rules, but fraudsters adapt quickly.<sup>3</sup>
- Deep Learning models (CNN, LSTM) can spot subtle, non-linear patterns in transaction data.
- However, financial institutions demand transparency for regulatory and trust reasons.<sup>4</sup>
- XAI methods help open up these black boxes, making model decisions understandable to humans.

## Why Local Explanations?

Local explanations help analysts understand *why* a specific transaction was flagged as fraud, which is crucial for real-world deployment.

---

<sup>3</sup>Sundararamaiah et al., 2024.

<sup>4</sup>Gilpin et al., 2018.

## Methods

---

# Dataset and Preprocessing

---

- **Dataset:** Sparkov synthetic data<sup>5</sup>, 1.2M transactions, 22 features, fraud rate  $\approx$  0.58%.
- **Preprocessing:**
  - Feature engineering (e.g., Haversine distance, age groups, temporal features).
  - Standardisation and encoding of categorical variables.
  - SMOTE<sup>6</sup> for balancing the highly imbalanced dataset.

---

<sup>5</sup>Harris, n.d.

<sup>6</sup>Bowyer et al., 2011.



# Model Architectures

---

## CNN Architecture

- Input: (15, 1) feature vector.
- Two Conv1D layers (64, 32 filters), batch norm, dropout.
- Dense layers with ReLU, final sigmoid for binary classification.
- **Parameters:** 60,065.



Figure: CNN Model Architecture

# Model Architectures

---

## CNN Architecture

- Input: (15, 1) feature vector.
- Two Conv1D layers (64, 32 filters), batch norm, dropout.
- Dense layers with ReLU, final sigmoid for binary classification.
- **Parameters:** 60,065.

## LSTM with Attention

- Input: (15, 1) feature vector.
- Lambda layer to expand dimensions.
- Two LSTM layers (50 units each) with dropout (0.3) and recurrent dropout (0.2).
- Custom attention mechanism.
- Dense output with sigmoid for binary classification.
- **Parameters:** 33,502.



Figure: LSTM Model Architecture<sup>7</sup>

---

<sup>7</sup>Ibtissam et al., 2021.

# Explainable AI (XAI) Techniques and Metrics

---

- **SHAP:** SHapley Additive exPlanations, provides both global and local feature attributions.
- **LIME:** Local Interpretable Model-agnostic Explanations, explains individual predictions with local surrogate models.
- **Anchors:** Rule-based, high-precision explanations for individual predictions.

## Evaluation Metrics

- **Faithfulness:** How well explanations reflect the model's true decision process.
- **Monotonicity:** Whether increasing a feature's value increases its importance.
- **Completeness:** How much of the model's behaviour is captured by the explanation.

# XAI Methods: SHAP

---

## How SHAP values are computed:

- Background dataset represents "average" feature values.
- Establishes baseline prediction for comparison.
- Measures feature impact by swapping actual values with background values.

---

## Algorithm SHAP for Credit Card Detection

---

- 1: **Input:** Trained model  $f$ , transaction  $x$
  - 2: **for** each feature  $i$  in  $x$  **do**
  - 3:     Initialize SHAP value  $\phi_i = 0$
  - 4:     **for** each subset  $S$  of features not containing  $i$  **do**
  - 5:         Create two samples:  $x_{S \cup \{i\}}$  and  $x_S$
  - 6:         Compute marginal contribution:  $f(x_{S \cup \{i\}}) - f(x_S)$
  - 7:         Weight the contribution based on subset size
  - 8:         Add weighted contribution to  $\phi_i$
  - 9:     **end for**
  - 10: **end for**
  - 11: **Output:** SHAP values  $\phi_1, \phi_2, \dots, \phi_n$  showing each feature's contribution to  $f(x)$
-

# XAI Methods: LIME

---

## How LIME Works

- Creates perturbed samples around the original transaction
- Weights samples by proximity to the original based on a kernel value
- Fits an interpretable model locally
- Shows feature contributions with confidence intervals

---

### Algorithm LIME for Credit Card Detection

---

- 1: **Input:** Trained model  $f$ , transaction  $x$
  - 2: Generate  $N$  perturbed samples around  $x$  by randomly changing feature values
  - 3: **for** each perturbed sample  $x'$  **do**
  - 4:     Predict  $f(x')$
  - 5:     Compute similarity between  $x$  and  $x'$
  - 6: **end for**
  - 7: Fit a simple interpretable model  $g$  (e.g., linear model) to predict  $f(x')$  using the perturbed samples, weighted by similarity
  - 8: **Output:** Coefficients of  $g$  as explanations for  $f(x)$ 's prediction
-

# XAI Methods: LIME

---

## What are Anchors?

- Highly precise "IF-THEN" rules explaining model decisions
- Focus on minimum conditions needed to maintain prediction
- Trade precision for coverage (fewer cases explained)
- Easily understood by non-technical stakeholders

---

### Algorithm Anchors for Credit Card Detection

---

- 1: **Input:** Trained model  $f$ , transaction  $x$
  - 2: Initialize anchors  $A = \emptyset$
  - 3: **while** precision of  $A$  (fraction of perturbed samples where  $f$  predicts same as  $f(x)$ )  $<$  threshold **do**
  - 4:     For each candidate feature not in  $A$ :
  - 5:         Add feature to  $A$  and estimate new precision
  - 6:         Add feature that increases precision the most to  $A$
  - 7: **end while**
  - 8: **Output:** Anchors  $A$ : set of feature-value rules that "guarantee"  $f(x)$ 's prediction with high precision
-

# XAI Evaluation: Faithfulness Metric

---

## Evaluate an explanation with Faithfulness metric:

- Measures the correlation between feature importance and changes in the model's predictions when features are altered
- Feature importance correlation analysis with sequential feature removal

---

### Algorithm Faithfulness Metric for XAI Explanations

---

- 1: **Input:** Model  $f$ , sample  $x$ , explanation scores  $E$  for features
  - 2: **for** each feature  $i$  in  $x$  **do**
  - 3:     Remove or mask feature  $i$  in  $x$  to get  $x_{-i}$
  - 4:     Compute prediction difference:  $\Delta_i = |f(x) - f(x_{-i})|$
  - 5: **end for**
  - 6: Compute correlation between  $\Delta_i$  and  $E_i$  across all features
  - 7: **Output:** Faithfulness score (e.g., correlation coefficient)
-

# XAI Evaluation: Monotonicity Metric

---

## Evaluate an explanation with Monotonicity metric:

- Evaluates whether removing features causes consistent changes in the model's predictions
- Sequential feature removal testing with prediction tracking

---

### Algorithm Monotonicity Metric for XAI Explanations

---

- 1: **Input:** Model  $f$ , sample  $x$ , explanation scores  $E$  for features
- 2: **for** each feature  $i$  in  $x$  **do**
- 3:     Change feature  $i$ 's value to get  $x_{+i}$
- 4:     Compute prediction change:  $\Delta_i = f(x_{+i}) - f(x)$
- 5:     **if**  $E_i > 0$  **then**
- 6:         Check if  $\Delta_i > 0$  (prediction increases)
- 7:     **else if**  $E_i < 0$  **then**
- 8:         Check if  $\Delta_i < 0$  (prediction decreases)
- 9:     **end if**
- 10: **end for**
- 11: Calculate fraction of features where explanation and prediction change agree
- 12: **Output:** Monotonicity score (agreement ratio)



# XAI Evaluation: Completeness Metric

---

## Evaluate an explanation with Completeness metric:

- Assesses how much of the model's prediction is captured by the explanation
- Coverage measurement comparing explained variance to total variance

---

### Algorithm Completeness Metric for XAI Explanations

---

- 1: **Input:** Model  $f$ , sample  $x$ , explanation scores  $E$  for features
  - 2: Compute model prediction:  $y = f(x)$
  - 3: Compute baseline prediction:  $y_{base} = f(\text{baseline input})$
  - 4: Sum explanation scores:  $S = \sum_i E_i$
  - 5: Compute completeness error:  $|S - (y - y_{base})|$
  - 6: **Output:** Completeness score (lower error means higher completeness)
-

# Balancing with Stratified Sampling

---

- **Class Imbalance Challenge:**
  - Only 0.58% of transactions are fraudulent in training data
  - Risk of model bias towards predicting legitimate transactions
- **Confidence-based Stratified Sampling:** For comprehensive XAI evaluation

## Confidence Bins for Stratified Evaluation

- **Very Low** (0.0-0.2) — Strong contradiction to model classification
- **Low** (0.2-0.4) — Weak patterns contradicting classification
- **Borderline** (0.4-0.6) — Ambiguous cases with mixed signals
- **High** (0.6-0.8) — Strong but not conclusive patterns
- **Very High** (0.8-1.0) — Clear fraud/non-fraud patterns

# Balancing with Stratified Sampling

---

- **Class Imbalance Challenge:**
  - Only 0.58% of transactions are fraudulent in training data
  - Risk of model bias towards predicting legitimate transactions
- **Confidence-based Stratified Sampling:** For comprehensive XAI evaluation

## Key Benefits

- Ensures comprehensive XAI evaluation across confidence levels
- Includes edge cases with extreme feature values
- Enables fair comparison between SHAP, LIME and Anchors

# Edge Case Handling

---

- **Edge Case Identification:**

- Z-score calculation to identify feature value deviations
- Selection of transactions with extreme feature values
- Analysis of unusual transaction patterns

## Selection Process

- **Calculate z-scores:**  $z = \frac{|x - \mu|}{\sigma}$  for important features
- **Maximum deviation:** Find samples with largest z-scores across features
- **Combine with stratified samples:** Ensures both typical and extreme cases
- **Special handling:** Edge cases receive additional manual review

# System Architecture

1. **Data Collection & Model Design:** Obtain synthetic transaction data and design fraud detection models.
2. **Data Preprocessing:** Raw transaction data is cleaned, normalised, and balanced using the SMOTE method.
3. **Model Training:** Separate pipelines are implemented for training models.
4. **XAI Integration:** Use XAI methods to generate explanations for model predictions.
5. **Performance Evaluation:** XAI evaluation metrics (**Faithfulness, Monotonicity, Completeness**).



Figure: System architecture illustrative diagram

## Results

---

# Model Performance: CNN vs LSTM

- I compared two deep learning architectures: a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) model with attention.
- Both models were trained and evaluated on the Sparkov synthetic dataset, which is highly imbalanced and mimics real-world credit card transaction patterns.

Table: Performance Metrics for CNN and LSTM Models

Model	Accuracy	ROC AUC	Precision (Fraud)	Recall (Fraud)
CNN	98.66%	0.994	21.34%	91.84%
LSTM	97.58%	0.971	11.80%	81.49%

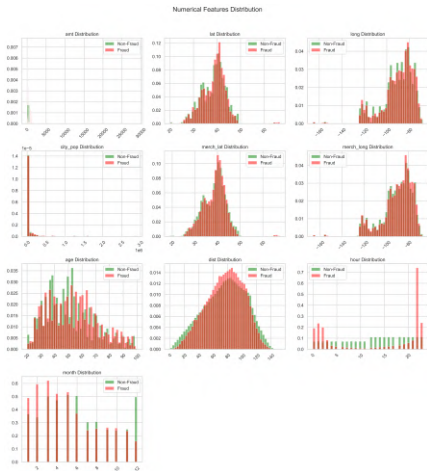
## CNN Model

- True Negatives: 546,311
- False Positives: 7,263
- False Negatives: 175
- True Positives: 1,970

## LSTM Model

- True Negatives: 540,512
- False Positives: 13,062
- False Negatives: 397
- True Positives: 1,748

# Feature Patterns using Statistical Analysis



(a) Numerical Features Distributions



(b) Categorical Features Distributions

Figure: Features Distributions in the Sparkov Train Dataset



# Feature Patterns using Statistical Analysis

---

## Key Statistical Patterns for Fraud Detection (without XAI methods)

- **Amount Anomalies:** Transactions exceeding \$500 showed 3.7x higher fraud probability
- **Temporal Patterns:** 23:00-04:00 transactions had 2.9x increased fraud risk
- **Geographical Anomalies:** Transactions >75km from cardholder location showed strong fraud indicators
- **Demographic Patterns:** Highest fraud concentration in 50-60 age group (1,443 cases)

# SHAP Summary Plot and Global Importance

- **Feature Impact Analysis:**

- Summary plot shows feature importance across all test samples
- Color represents feature value (red = high, blue = low)
- Position shows impact on prediction (right = toward fraud)

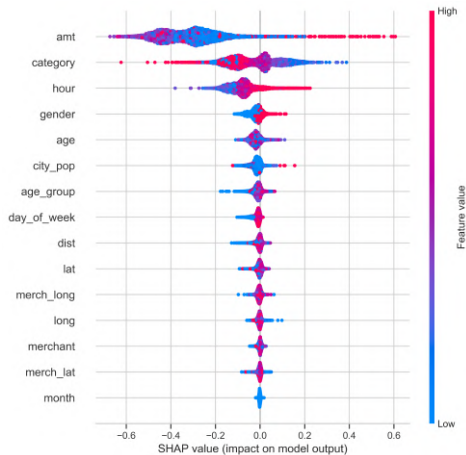


Figure: SHAP Summary Plot: Feature Importance for CNN Model

# Feature Importance: What Drives Fraud Predictions?

- Using SHAP, I identified the most influential features for the CNN model.
- **Top features:**
  - **Transaction Amount:** Higher values are a strong fraud indicator.
  - **Merchant Category**
  - **Hour of Transaction:** Transactions between 23:00 and 04:00 are riskier.
  - **Gender:** and **Age** Moderate influence.
  - **Distance to Merchant:** Unusual distances often signal fraud.
  - **Latitude/Longitude:** Minimal impact.

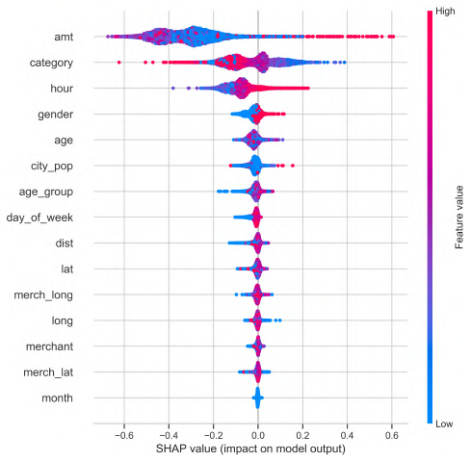


Figure: SHAP Summary Plot: Feature Importance for CNN Model

# SHAP Results Local Visualization

## SHAP Waterfall Plot

- Shows step-by-step impact
- Starting from base value (0.511)
- Category, hour, and age\_group decrease fraud probability
- Final prediction: 0.989 (98.9% fraud confidence)

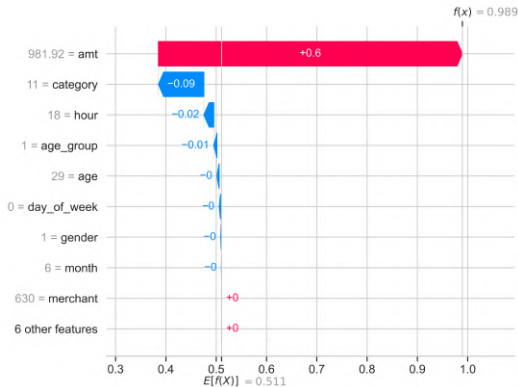


Figure: SHAP Waterfall Plot for the LSTM model's prediction (98,92%) at sample index 1044

# SHAP Results Local Visualization

## SHAP Force Plot

- Shows feature contributions pushing prediction from the base value
- Red = pushing toward fraud
- Blue = pushing toward legitimate
- Example: Transaction amount (+0.6) strongly indicates fraud

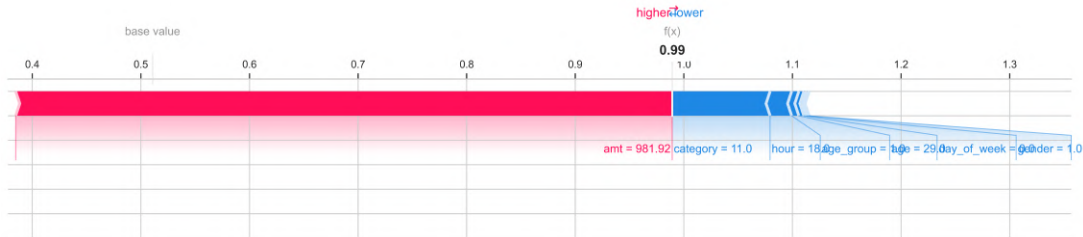


Figure: SHAP Force Plot for the LSTM model's prediction (98,92%) at sample index 1044

# LIME Explanation

## Example Interpretation

- Green bars support legitimate prediction
- Red bars indicate fraud signals
- Width shows contribution magnitude
- **Result:** 99.48% confidence in fraud

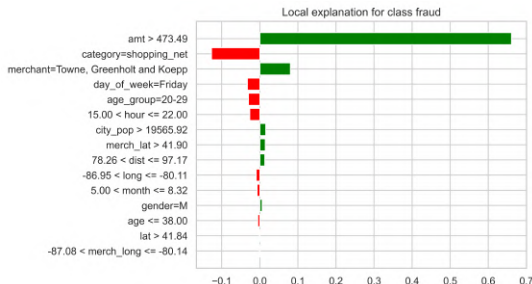


Figure: LIME explanation showing feature contributions for the CNN model's prediction (99.48%) at sample index 1044

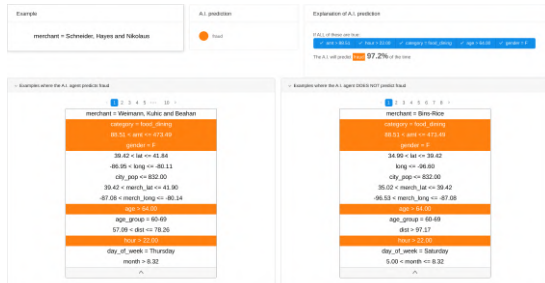
# Anchors Rule-Based Explanations

## Rule-Based Explanations:

1. `amt > 88.51`
2. `hour > 22.00`
3. `category = food_dining`
4. `age > 64.00`
5. `gender = F`

**Precision:** 97.2% of cases matching this rule are correctly predicted as fraud

**Coverage:** 8.4% of all fraud cases are covered by this rule



**Figure:** Anchors Explanation Interactive Observation in Notebook for the LSTM model's prediction (79,83%) at sample index 2025

# XAI Methods Performance Analysis

## Performance by Confidence Levels

- **SHAP:** Consistent across all confidence levels (0.594-0.629)
- **LIME:** U-shaped pattern, better at high/low confidence
- **Anchors:** Best for very high confidence predictions

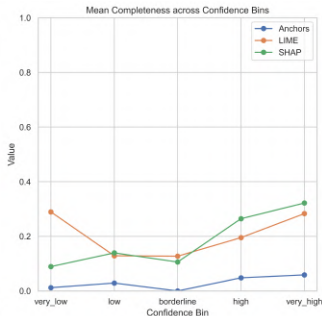
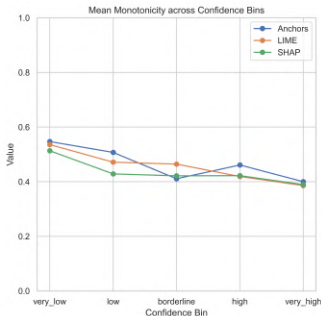
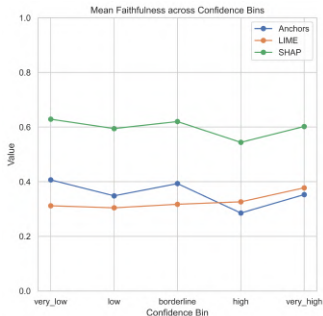


Figure: XAI Methods Performance Across Confidence Bins



# XAI Effectiveness Across Confidence Levels

- The effectiveness of the XAI method depends on the model's prediction confidence.
- **Very High Confidence ( $>0.8$ ):** SHAP explanations are most reliable.
- **High Confidence ( $0.6-0.8$ ):** SHAP and LIME together provide a balanced view.
- **Borderline ( $0.4-0.6$ ):** A multi-method approach (SHAP, LIME, Anchors) is best.
- **Low Confidence ( $<0.4$ ):** SHAP plus human review is recommended.

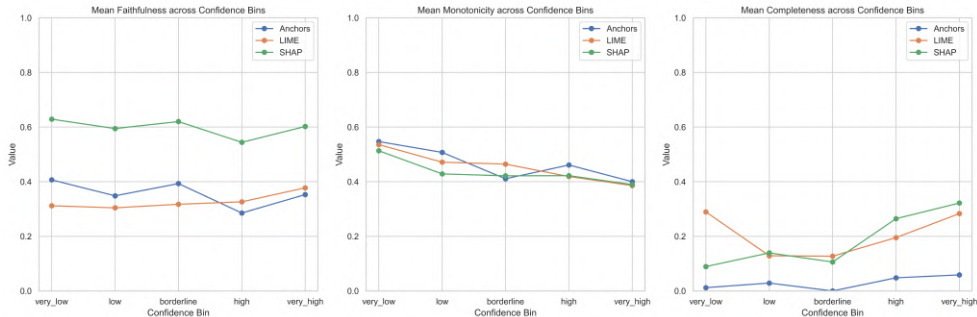


Figure: XAI Methods Performance Across Confidence Bins

# XAI Methods: Performance Comparison Between Models

## Key Findings:

- SHAP achieved the highest faithfulness for both models:
  - **LSTM:** 0.761 (significantly higher)
  - **CNN:** 0.443
- LSTM model consistently showed better faithfulness scores across all XAI methods
- Monotonicity and Completeness showed less variation between models

Model & Method	Faith.	Mono.	Comp.
<b>CNN + SHAP</b>	0.443	0.464	0.208
CNN + LIME	0.254	0.488	<b>0.295</b>
CNN + Anchors	0.315	0.486	0.033
<b>LSTM + SHAP</b>	<b>0.761</b>	0.429	0.134
LSTM + LIME	0.396	0.445	0.139
LSTM + Anchors	0.412	<b>0.469</b>	0.022

Table: XAI Methods Performance Metrics

# Summary of Key Findings

---

- **SHAP is the most faithful XAI method**, especially for high-confidence predictions, providing explanations that align closely with the model's actual decision process.
- **LIME** offers balanced completeness and is particularly useful for local, case-by-case explanations.
- **Anchors** delivers the most human-interpretable rules, though with limited coverage.
- **Key global fraud indicators** across all models: transaction amount, category, and transaction hour.
- **XAI method effectiveness** varies with prediction confidence, SHAP is robust across all levels, while LIME achieves the best completeness. Anchors can give an easy-to-understand explanation.

## Takeaway

Integrating XAI with deep learning models not only boosts trust and transparency but also provides actionable insights for fraud analysts and operational teams.

## Discussion

---

# Practical Implications and Challenges

---

- **Real-world deployment:** Both models process single transactions in under 5 seconds, and generate explanations in less than 3 minutes, making them suitable for real-time fraud detection.
- **Computational overhead:** SHAP explanations are computationally intensive (up to 45 minutes for 100 explanations), requiring careful resource management.
- **Stakeholder needs:**
  - SHAP is best for data scientists needing technical depth.
  - LIME is ideal for fraud analysts seeking intuitive, local explanations.
  - Anchors are valuable for operational teams needing clear, actionable rules.
- **Access to real data:** As changing fraud patterns in the real world would be a problem
- **Model complexity trade-offs** while some approaches offer good explanations, they might not be optimised for the "ingenuity of fraudsters" and high transaction volumes that are the main challenges in fraud detection.

## Conclusions

---

# Conclusions

---

- Successfully integrated XAI methods (SHAP, LIME, Anchors) with deep learning models for credit card fraud detection.
- Developed a novel confidence-based evaluation framework for XAI effectiveness.
- **SHAP** is recommended for global model interpretation and high-stakes applications.
- **LIME** is best for local, case-by-case explanations.
- **Anchors** are ideal for generating clear, actionable rules for operational teams.

## Final Thought

XAI is not just a technical add-on, it is essential for building trust, meeting regulatory requirements, and empowering analysts in the fight against fraud.

# Future Work

---

- **Expand model diversity:** Explore more advanced Deep Learning models and their explainability for fraud detection.
- **User studies:** Conduct usability studies with fraud analysts to assess the practical value of XAI explanations.
- **Optimise XAI computation:** Investigate faster, scalable XAI methods for real-time deployment.
- **More advanced DL-based explainers:** DeepLIFT<sup>8</sup> (Deep Learning Important FeaTures), X-NeSyL<sup>9</sup> (eXplainable Neural Symbolic Learning).

## Invitation

I welcome any questions, feedback, or collaboration ideas. Let's make fraud detection smarter and more transparent together!

---

<sup>8</sup>Shrikumar et al., 2017.

<sup>9</sup>Díaz-Rodríguez et al., 2022.



# Poster Presentation



## Explainable AI (XAI) in Deep Learning Models for Credit Card Fraud Detection

Thong Minh Lai<sup>1</sup> Supervisor: Dr. Hyunook Lee<sup>2</sup>

<sup>1</sup>University of Huddersfield, Department of Computer Science

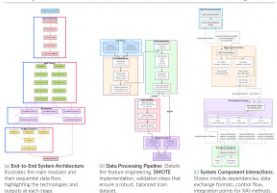
### Introduction & Motivation

Credit card fraud remains a persistent and costly issue, with UK losses reaching £1.3 billion in 2023 (UK Finance, 2023). While deep learning models have shown promise for fraud detection, their "black box" nature makes them difficult to trust and deploy in real-world financial systems. This project aims to bridge that gap by integrating Explainable AI (XAI) techniques, making model decisions transparent and actionable for analysts and stakeholders.

#### Project Objectives:

- Develop and compare models explainability (CNN, LSTM with attention) for fraud detection.
- Integrate XAI methods (SHAP, LIME, Anchors) for local interpretability.
- Evaluate the explainability using XAI metrics (Faithfulness, Monotonicity, Completeness).
- Provide insights for financial institutions implementing transparent AI.

### System Architecture: End-to-End Workflow and Modular Design



- Data Collection & Model Design:** Gather *Sparkov* synthetic transaction data (10k items, n.d.), design model architectures (CNN, LSTM). Data is loaded and versioned for reproducibility.
- Data Preprocessing:** Applies feature engineering (temporal, spatial, demographic), handles class imbalance using SMOTE (Bossey et al., 2013), and standardizes/removes features to ensure high-quality model input.
- Model Training (CNN & LSTM):** Trains and optimizes for fraud detection. Each model is tested and validated independently.
- XAI Integration:** Implements SHAP, LIME, and Anchors for local interpretability, providing instance-level explanations for model predictions.
- Explainability Performance Evaluation:** Comprehensive XAI evaluation metrics (Faithfulness, Monotonicity, Completeness), with visual dashboards for comparative analysis.

lai@thonglai@gmail.com | t1234567@journal.hudd.ac.uk

### Predictive Model Architectures: CNN and LSTM



#### CNN:

- Conv2D layers (64, 32 filters)
- Batch Normalisation, Dropout, Dense layers
- 59,937 trainable parameters

#### Figure 3: LSTM Model Layers with Attention Mechanism



#### LSTM with Attention:

- Two LSTM layers (50 units each)
- Attention mechanism, Dense output layer
- 33,502 trainable parameters

### Explanations from XAI Methods: SHAP, LIME, and Anchors

SHAP (Shapley Additive exPlanations) visualisations, including the waterfall and force plots, illustrate how individual features influence the model's prediction. In the waterfall plot, each bar represents a feature's contribution, starting from a base value (e.g., 0.5111). Red bars indicate features that increase the probability of fraud, while blue bars show features that decrease it. The force plot provides a complementary view, displaying the cumulative effect of all features, with the width of each bar indicating the magnitude of its impact. Features are arranged by their influence, making it easy to identify which factors most strongly push the prediction towards fraud or legitimacy.



Figure 4: SHAP Waterfall Plot: Contributions for a CNN prediction.



Figure 5: SHAP Force Plot: Cumulative impact from base value.

LIME (Local Interpretable Model-agnostic Explanations) visualisations use a bar chart to display feature importance and directionality for a specific prediction. Green bars represent features that support legitimate transactions, while red bars indicate features suggesting fraud. The length of each bar corresponds to the magnitude of the feature's impact, and features are ordered by their absolute importance, allowing for quick identification of the most influential factors.

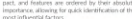


Figure 6: LIME Explanation Chart: Bar visualisation of feature impact.



Figure 7: Anchors Rule-based: Conditions for a prediction.

### XAI Evaluation Metrics and Insights

#### XAI Metrics:

- Faithfulness:** How well explanations reflect model behavior. Higher scores indicate better alignment with model predictions.
- Monotonicity:** Consistency of feature importance across different inputs.
- Completeness:** Coverage of the model's decision process by the explanation.

Table 1: Evaluation Metrics Comparison for XAI Methods on CNN, LSTM Models. Higher values are better for all metrics.

Metric	SHAP (CNN)	LIME (CNN)	Anchors (CNN)	SHAP (LSTM)	LIME (LSTM)	Anchors (LSTM)
Faithfulness	0.781	0.734	0.612	0.540	0.574	0.511
Monotonicity	0.429	0.445	0.469	0.481	0.488	0.485
Completeness	0.134	0.135	0.027	0.268	0.295	0.033

#### Key Takeaways:

- SHAP** is the most robust, especially with LSTM, making it a go-to for high-stakes explanations.
- LIME** offers the best completeness with CNN; is more intuitive for fraud analysts.
- Anchors** are great for clear, actionable rules, but their coverage is limited.
- The choice of XAI method should be guided by the confidence of the prediction and the end-user's needs.

#### XAI Effectiveness by Confidence Level

- Very High (>0.9):** SHAP explanations most reliable.
- High (0.8-0.9):** SHAP and LIME provide a balanced view.
- Baseline (0.4-0.6):** Multi-method approach recommended.
- Low (<0.4):** SHAP human review recommended.



Figure 8: Bar Plot: Comparing Faithfulness, Monotonicity, and Completeness across models and XAI methods.

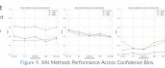


Figure 10: Radar Chart: Comparison of All Average Metrics Values Across Methods and Models.

### Practical Implications

- Real-time Detection:** A transaction's explanation can be produced in under 3 minutes, suitable for operational deployment.
- Stakeholder Suitability:**
  - SHAP: Data scientists (technical depth)
  - LIME: Fraud analysts (intuitive, visual)
  - Anchors: Compliance (clear, rule-based)
- Computational Overhead:** SHAP is resource-intensive, requiring careful optimisation of resource management.

### Limitations & Future Work

- Data:** Reliance on synthetic data (*Sparkov*) may limit real-world generalisability.
- User Experience:** XAI outputs can be complex for non-technical users.
- Next Steps:**
  - Optimize XAI computation for real-time use
  - Further research on other advanced explainers (DeepLIFT, X-Model, etc.)
  - Conduct usability studies with fraud analysts
  - Incorporate non-domain, real-world datasets

### References

UK Finance. (2023). *UK Fraud Statistics 2023*. Retrieved from <https://www.ukfinance.co.uk/pdfs/fraud-statistics-2023.pdf>

<https://github.com/ThongLai/Credit-Card-Transaction-Fraud-Detection-Using-Explainable-AI>

Full poster available at: <https://docs.google.com/viewer?url=githu.com/ThongLai/Credit-Card-Transaction-Fraud-Detection-Using-Explainable-AI/blob/main/poster.pdf?raw=true>

# Acknowledgements

---

- I would like to thank my supervisor, Dr. Hyunkook Lee, for his guidance, expertise, and encouragement throughout this project.
- A grateful attitude to the pioneering researchers behind the previous brilliant works on XAI concepts and Deep Learning architecture designs, Sparkov synthetic data in credit card fraud detection, which made this project possible.

# References

---

-  Bowyer, K. W., Chawla, N. V., Hall, L. O., & Kegelmeyer, W. P. (2011). SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813. <http://arxiv.org/abs/1106.1813>
-  Díaz-Rodríguez, N., Lamas, A., Sanchez, J., Franchi, G., Donadello, I., Tabik, S., Filliat, D., Cruz, P., Montes, R., & Herrera, F. (2022). Explainable neural-symbolic learning (x-nesyl) methodology to fuse deep learning representations with expert knowledge graphs: The monumai cultural heritage use case. *Information Fusion*, 79, 58–83. <https://doi.org/https://doi.org/10.1016/j.inffus.2021.09.022>
-  Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, 80–89. <https://doi.org/10.1109/DSAA.2018.00018>
-  Grover, P., Xu, J., Tittelfitz, J., Cheng, A., Li, Z., Zablocki, J., Liu, J., & Zhou, H. (2023). Fraud dataset benchmark and applications. <https://arxiv.org/abs/2208.14417>
-  Harris, B. (n.d.). *Generate fake credit card transaction data, including fraudulent transactions*. GitHub. Retrieved April 23, 2025, from [https://github.com/namebrandon/Sparkov\\_Data\\_Generation](https://github.com/namebrandon/Sparkov_Data_Generation)
-  Ibtissam, B., Samira, D., Bouabid, E. O., & Jaafar, J. (2021). Enhanced credit card fraud detection based on attention mechanism and lstm deep model. *Journal of Big Data*, 8(151). <https://doi.org/10.1186/s40537-021-00541-8>
-  Shrikumar, A., Greenside, P., & Kundaje, A. (2017, August). Learning important features through propagating activation differences. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning* (pp. 3145–3153, Vol. 70). PMLR. <https://proceedings.mlr.press/v70/shrikumar17a.html>
-  Sundararamaiah, M., Nagarajan, S. K. S., Mudunuru, K. R., & Remala, R. (2024). Unifying ai and rule-based models for financial fraud detection. *International Journal of Computer Trends and Technology*, 72, 61–68. <https://doi.org/10.14445/22312803/IJCTT-V72I12P107>
-  UK Finance. (2022). *Annual fraud report 2022*. UK Finance. Retrieved April 23, 2025, from <https://www.ukfinance.org.uk/policy-and-guidance/reports-and-publications/annual-fraud-report-2022>

# Thank you!

U2259343@unimail.hud.ac.uk

Github: [ThongLai/Credit-Card-Transaction-Fraud-Detection-Using-Explainable-AI](#)

*Questions and discussion welcome!*