For example, platforms that enable trading and liquidity provision (like decentralized exchanges or automated market makers) may be considered to operate a trading platform and could be required to comply with MiCA's requirements for transparency, risk management, and consumer protection. Furthermore, MiCA mandates a crypto-asset white paper, where issuers must disclose detailed information about the asset, its risks, the underlying technology, and its environmental impact. These disclosures are crucial for consumer protection and ensuring that investors are fully informed about the assets they engage with. By enforcing these regulations, MiCA aims to increase accountability within the DeFi space, ensuring that even decentralized protocols that engage in significant economic activities are subject to oversight.

2.2.4 Badan Pengawas Perdagangan Berjangka Komoditi (BAPPEBTI) Indonesia

Indonesia's Commodity Futures Trading Supervisory Agency (BAPPEBTI) has introduced regulations to govern crypto asset trading, including DeFi-related transactions. One important step is BAPPEBTI's latest regulation Number 8 of 2024[23] which focuses on improving consumer protection, transaction supervision, and registration of Crypto Asset Physical Traders (PFAK). In this regulation, BAPPEBTI emphasizes the importance of stricter supervision of crypto asset transactions, including those taking place in the DeFi space. This regulation also includes rules related to crypto asset exchange governance and transaction monitoring systems to ensure that trading activities can run in an orderly and transparent manner, and comply with security and consumer protection standards.

Furthermore, BAPPEBTI also introduces an obligation for crypto exchanges to have a secure, accurate, and real-time accessible system for reporting crypto asset transactions. This is expected to minimize the risk of fraud and other problems that often arise in the DeFi ecosystem. Supervision by BAPPEBTI is expected to provide assurance for investors in transactions, while reducing the potential for abuse or risks associated with trading crypto assets.

2.2.5 UU ITE

In Indonesia, the use and transactions in DeFi (Decentralized Finance) are also governed by a number of regulations, including the Electronic Information and Transaction Law (UU ITE). Law No. 11/2008 ITE[24], as amended by Law No. 1/2024, regulates electronic transactions and the protection of personal data within the scope of the digital world. This regulation applies not only to transactions involving conventional products or services, but also covers activities that take place in blockchain and DeFi systems, especially in relation to

transactions involving crypto assets.In the context of DeFi, the ITE Law provides a legal basis for transactions using blockchain-based technology, including smart contracts and electronic system providers that can be involved in DeFi.

Several articles in this law mention the obligation to maintain data security and electronic transactions, as well as the protection of electronic system users who are potentially involved in illegal activities such as fraud or data misuse. On the other hand, although the ITE Law regulates electronic transactions and blockchain, in practice, the application of these regulations to DeFi is still under development. The Indonesian government also issued regulations related to crypto assets involving BAPPEBTI to provide further guidelines in regulating activities such as trading and storage of crypto assets. With this law, DeFi providers in Indonesia are expected to comply with applicable legal provisions, including rules related to digital transactions and personal data protection.

2.2.6 Otorisasi Jasa Keuangan

Indonesia's Financial Services Authority (Otoritas Jasa Keuangan, OJK) has established a comprehensive regulatory framework to oversee financial technology (fintech) services, electronic financial transactions, and fraud detection systems. Among the notable regulations is POJK No. 13/POJK.02/2018, which emphasizes the importance of cybersecurity, data protection, and transparency in digital financial services. Additionally, POJK No. 77/POJK.01/2016 governs information technology-based lending services, focusing on consumer protection, system reliability, and fair transaction practices. These regulations underscore the critical role of advanced fraud detection mechanisms, such as machine learning, in safeguarding the financial ecosystem.

Furthermore, POJK No. 27/POJK.03/2024 has been introduced to specifically address the integration of innovative technologies in financial services, including blockchain-based systems. This regulation provides guidelines on utilizing technology for transaction monitoring, fraud detection, and risk management, particularly in electronic payment systems and decentralized finance (DeFi) platforms. By requiring financial institutions to adopt real-time monitoring systems and robust security measures, OJK aims to minimize fraud risks and enhance transparency. Collectively, these regulations reflect OJK's commitment to fostering a secure, stable, and trustworthy fintech environment while ensuring compliance with global standards and prioritizing consumer protection.

CHAPTER 3 SYSTEM MODEL DESIGN

3.1 Research Design

The system design for detecting blockchain transaction fraud utilizes a RoBERTa-based framework with a robust validation methodology to ensure efficient and reliable model training and evaluation. The process begins with loading the raw transaction dataset. To prevent data leakage and ensure unbiased final evaluation, the first crucial step is to perform an initial data split, which partitions the entire dataset into a larger training/validation set and a smaller test set, stored separately. After this split, the pre-processing stage is applied independently to both sets. This process includes feature engineering, where structured data is converted into descriptive sentences, which are then tokenized and encoded to be suitable for analysis by the language model.

The core of this system validation process operates on the processed training/validation set, using a K-Fold Cross-Validation loop. Within this loop, the data is divided again into 'K' parts (folds), and the entire training-evaluation cycle is repeated 'K' times. In each iteration, a new pre-trained RoBERTa model is loaded to leverage its transformer architecture in capturing complex patterns. This model is then trained on K-1 folds. The training phase begins with a forward pass to generate predictions, followed by loss calculation. To improve computational efficiency on limited hardware, the system implements gradient accumulation, which accumulates gradients over several backward passes before performing a single model parameter update, effectively simulating training with a larger batch size.

After the model is trained in one iteration of the loop, its performance is immediately measured on the remaining validation fold. Performance metrics such as accuracy, precision, recall, and F1-score are calculated and stored for that iteration. After the K-Fold loop is complete, all stored performance metrics from each fold are aggregated, typically by averaging, to produce a stable and reliable estimate of model performance. Based on this validation, a final model is retrained using the entire data in the training-validation set to create the most optimal version of the model. This final model then undergoes a final evaluation on the test set that has been stored from the beginning. If the results of this unbiased final evaluation meet the specified threshold, the model is saved for deployment purposes.

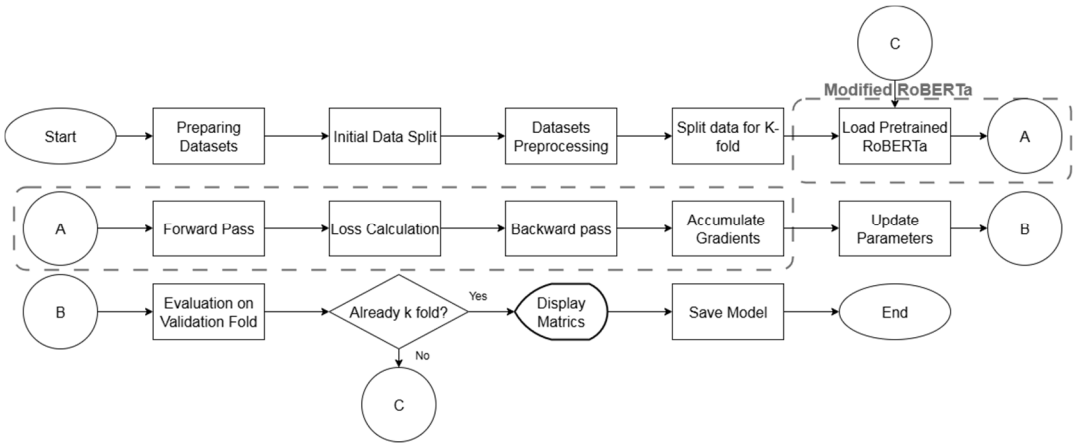

Figure 3. 1 Reseach flow

3.2 Simulation Scenario

To conduct a simulation for the fraud detection system using RoBERTa on blockchain transaction data, the scenario simulation can take the form of either real-world simulations or software-based simulations. This is the outline of how this simulation would work in both cases:

1. Real-World Simulation Scenario:

This scenario involves testing the fraud detection system within an actual blockchain environment to ensure its practicality.

1. Data Collection:
   - Access Ethereum blockchain data.
   - Collect both historical and live transaction data, including metadata such as sender/receiver addresses, transaction amounts, timestamps, and smart contract interactions.
   - Ensure a mix of normal, anomalous, and fraudulent transaction patterns in the dataset.
2. Data Preprocessing:
   - Tokenize transaction metadata into embeddings for input to RoBERTa.
   - Remove irrelevant information, address null values, and normalize numeric data (e.g., transaction amounts).
3. Model Integration and Fine-Tuning:
   - Use RoBERTa pre-trained on a general NLP corpus and fine-tune it on the preprocessed transaction data.

- Employ gradient accumulation to train the model effectively on large datasets within memory constraints.

4. Fraud Classification:
   - Use the trained model to classify transactions into legitimate or fraudulent.
   - Employ thresholds on model outputs (e.g., probabilities) for fraud detection.
5. Action Trigger and Response:
   - Flag potentially fraudulent transactions for review or block them automatically.
   - Send alerts to system administrators or end-users.
6. Performance Monitoring:
   - Collect metrics such as accuracy, precision, recall, F1-score, false positive/negative rates, and latency.
   - Monitor the system's ability to detect evolving fraud patterns in real-time.

2. Software-Based Simulation Scenario:

Alternatively, a software simulation can be performed using data generated from a synthetic or historical dataset of blockchain transactions. The steps for the simulation would include:

- Synthetic Data Generation: In the case of using software-based simulation, the first step would be to generate or collect a dataset of blockchain transactions. This could be either real blockchain data (from a public ledger) or a synthetic dataset mimicking real transactions. The synthetic data would be designed to include a variety of transaction behaviors, such as normal transactions, those that match known fraud patterns, and outlier transactions.
- Data Preprocessing and Tokenization: Like the real-world scenario, this data would then be preprocessed. Tokenization would convert addresses and transaction features into a form that the RoBERTa model can process. This could involve converting addresses into embeddings or one-hot encoding.
- Model Evaluation: The preprocessed data would then be passed through the RoBERTa model. The simulated environment allows for testing different thresholds for fraud classification, as well as various configurations of the model (e.g., training with more or fewer historical transactions, adjusting learning rates, or trying out different tokenization strategies).
- Simulation of Alerts and Actions: Based on the classification output from the model, the software simulation would generate alerts and action triggers in real time. The simulation

could be configured to flag transactions with a high fraud probability and simulate actions such as halting a transaction or notifying a fraud detection team.

- Performance Monitoring: After running the simulation, performance metrics would be gathered to evaluate the model's performance. These metrics would include accuracy, precision, recall, and f1-score for fraud detection, helping assess the model's robustness and reliability.

In this scenario, tools like Python, TensorFlow, or PyTorch for model implementation and simulation software (such as Simulink or AnyLogic) for monitoring performance would be used to run the simulation. Additionally, the simulated blockchain environment could use tools like Ganache or Truffle to simulate transactions in a controlled manner.

Both scenarios provide an effective way of testing the fraud detection system, with the real-world simulation offering a closer approximation of real-world conditions, while the software-based simulation offers a more controlled environment for evaluating different configurations and optimizing the system.

3.3 Scenario Data Collection

For the Scenario Data Collection in this fraud detection system using blockchain data, the process involves obtaining real-time transaction data from the blockchain and processing it to extract key parameters that are used for fraud detection. The collection can be carried out through a real-time blockchain simulation or by leveraging API-based tools like Infura or Alchemy that pull live data from blockchain networks such as Ethereum.

Data Collection Scenario:

1. Data Source: The data collection begins by accessing blockchain transaction data from Ethereum (or another blockchain). This is typically done through API providers. These platforms allow users to query real-time transaction data directly from the blockchain without the need for hosting a full node. The APIs provide endpoints to fetch various data such as block number, transaction hashes, block timestamps, sender and receiver addresses, and transaction amounts.
2. Parameters Observed: The key parameters for this fraud detection system, which need to be observed and collected from each transaction, include:
    - Sender Address: The address initiating the transaction.
    - Receiver Address: The address receiving the funds.
    - Transaction Amount: The value of the transaction in cryptocurrency.
    - Transaction Hash: A unique identifier for each transaction.

- Gas Fees: The fees associated with processing the transaction.
- Block Timestamp: The time when the transaction was included in a block.
- Transaction Status: Whether the transaction was successful, pending, or failed.

These parameters are critical for detecting fraudulent activities, as abnormal patterns in transaction size, timing, and sender/receiver behavior could indicate fraud.

3. Collection Method: Data is collected from the blockchain via the Infura/Alchemy API, which allows querying the blockchain for specific blocks or transaction data in real-time.
4. Preprocessing the Collected Data: Once the data is collected, it is preprocessed to extract the relevant features. The addresses might be tokenized or converted into embeddings for input into the RoBERTa model. Additionally, any missing data should be handled by either imputing values or removing incomplete records. Data normalization or scaling could be applied to numerical values (such as the transaction amount) to bring them to a standard range for model processing.
5. Data Verification: To ensure the data is accurate and reflects the state of the blockchain in real-time, continuous monitoring is conducted. This could include checking for any anomalies in the blockchain node response times or discrepancies in data returned from the API calls.
6. Frequency of Data Collection: The frequency of data collection would depend on the system's configuration and how quickly new transactions are processed on the blockchain. For example, data could be collected at intervals (e.g., every minute) or triggered by each new block added to the blockchain.

3.4 Scenario Analysis

The Scenario Analysis for the fraud detection system using the RoBERTa model involves evaluating the results of the classification made by the model and understanding the effectiveness of the system. The goal of the analysis is to determine whether the detected anomalies are true frauds, whether any patterns can be identified in fraudulent transactions, and how well the model generalizes to new, unseen data. The following steps outline how the analysis would be conducted:

1. Analyzing model outputs: Once the RoBERTa model classifies a transaction as either fraudulent or legitimate, the analysis begins by examining the confidence scores output by the model. These scores indicate the model’s certainty in its classification. For example, a high confidence score (close to 1) in a fraudulent transaction suggests strong certainty that the transaction is indeed fraudulent. This data can be used to assess how

well the model differentiates between normal and abnormal patterns in the transaction data. The analysis will involve reviewing the predicted labels (fraud or legitimate) against actual transaction outcomes (whether the fraud was detected later by human verification or other systems). The analysis will compute accuracy, precision, recall, and F1 score, which are essential metrics for understanding model performance:

- Accuracy indicates how well the model performs in classifying all transactions correctly.
- Precision measures the proportion of fraudulent transactions detected out of all the flagged ones.
- Recall shows how well the model captures all the actual fraudulent transactions.
- F1 Score is a balanced measure between precision and recall, giving insight into the model's ability to detect fraud without overwhelming false positives.

2. Error analysis: After evaluating the overall performance of the model, the next step is conducting error analysis. This involves closely examining the false positives (transactions that were flagged as fraudulent but were legitimate) and false negatives (fraudulent transactions that were not flagged by the model). By inspecting these cases, it is possible to identify patterns in the data that the model might be missing or confusing. For example, certain legitimate transactions might be flagged as fraudulent due to unusual but non-fraudulent behaviors, such as sudden large transactions or cross-border transactions, which may need more nuanced handling. Identifying these patterns could lead to improvements in the model, such as fine-tuning certain thresholds or adding new features to the model to help distinguish between legitimate and fraudulent activities more accurately.
3. Behavioral pattern identification: The next part of the analysis is to identify any behavioral patterns in the fraudulent transactions. For example, a certain set of sender or receiver addresses might be involved in multiple fraudulent activities. The transaction amounts or timings may also reveal trends in fraudulent behavior. These patterns can be uncovered by visualizing data, for example through clustering techniques (like k-means clustering) or network analysis to observe relationships between addresses and transaction behaviors. Once patterns are identified, the system could be improved by integrating these patterns into the model, thus making it more responsive to emerging fraud schemes.
4. Continuous feedback loop: In this scenario, continuous monitoring and analysis are crucial for ensuring the system remains effective over time. As new fraudulent strategies emerge, the analysis can include a feedback loop where the detected fraudulent

transactions (false positives or missed frauds) are used to retrain the model. The system can also be configured to provide real-time insights to fraud analysts, who can manually inspect flagged transactions if necessary. Over time, these adjustments can help the model become more accurate, reduce false positives, and ensure more real-time fraud detection. Data from new transaction batches will be incorporated into the ongoing training cycle, allowing the model to stay up to date with the latest fraud patterns in the blockchain environment.

In summary, the Scenario Analysis after data collection focuses on understanding the performance of the fraud detection model, identifying its strengths and weaknesses, and continuously improving its accuracy. Through careful analysis of the output, error patterns, and ongoing feedback from new data, the system can evolve to detect fraud more effectively in real-time blockchain transactions.

3.5 Evaluation Performance

3.5.1 Confusion Matrix

To evaluate the performance of the classification model, a confusion matrix is used as the basis for all quantitative metric calculations. The confusion matrix presents a comparison between the actual class and the class predicted by the model, which is divided into four categories: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The structure of the confusion matrix can be seen in Table 3.1.

Tabel 3. 1. Confusion Matrix Structure

	Prediction: Fraud	Prediction: Normal
Actual: Fraud	True Positive (TP)	False Negative (FN)
Actual: Normal	False Positive (FP)	True Negative (TN)

Where each category in the matrix has the following definitions:

- True Positive (TP): The number of fraud cases that were correctly predicted as fraud.
- True Negative (TN): The number of normal cases correctly predicted as normal.
- False Positive (FP): The number of normal cases incorrectly predicted as fraud (Type I Error).
- False Negative (FN): The number of fraud cases incorrectly predicted as normal (Type II Error).

3.5.2 Accuracy

$$Acc. = \frac{TP + TN}{TP + TN + FP + FN} \quad \textbf{(3.1)}$$

Accuracy measures the proportion of correct predictions (both fraud and legitimate) out of the total number of transactions. This metric provides an overall understanding of the model's effectiveness.

3.5.3 Precision

$$Precision = \frac{TP}{TP + FP} \quad \textbf{(3.2)}$$

Precision focuses on the accuracy of positive predictions by measuring how many predicted fraudulent transactions are truly fraudulent. This metric is crucial in fraud detection systems to minimize false alarms, which can inconvenience users or lead to unnecessary manual reviews. A high precision indicates that the model is reliable in labeling transactions as fraudulent, ensuring trust in its decisions.

3.5.4 Recall

$$Recall = \frac{TP}{TP + FN} \quad \textbf{(3.3)}$$

Recall measures the model's ability to correctly identify actual fraud cases from all existing fraud cases. This metric is vital in scenarios where missing fraudulent transactions (false negatives) can result in financial losses or security breaches. A high recall ensures the system captures most fraud cases, prioritizing thorough detection over the risk of false positives.

3.5.5 F1-Score

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \textbf{(3.4)}$$

F1 Score is the harmonic mean of precision and recall. This metric provides a balance between the model's ability to avoid false positives and false negatives. This metric is essential when both aspects are equally important, as it ensures the model performs well overall without compromising one metric for the other.