# K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS), TRICHY.

## DEPARTMENT OF
## COMPUTER SCIENCE AND ENGINEERING

# 20CS5501 DESIGN PROJECT-1

**Batch No. : 01**

**Date : 05.12.2024**

# DETECTING MALICIOUS SMART CONTRACT INTERACTION USING MACHINE LEARNING

**Guided by**

Mr. R.VIGNESH KUMAR M.E.,

Assistant Professor, CSE

**Team**

ABINAYA J (811721104002)

ATCHAYADURGA K(811721104018)

JANE BEULA A (811721104160)

# OBJECTIVE OF THE PROJECT

- Automated Detection of Malicious Behavior
- Early Identification of Threats
- Improved Contract Auditing
- Real-time Detection and Monitoring
- Detection of Evolving Attacks

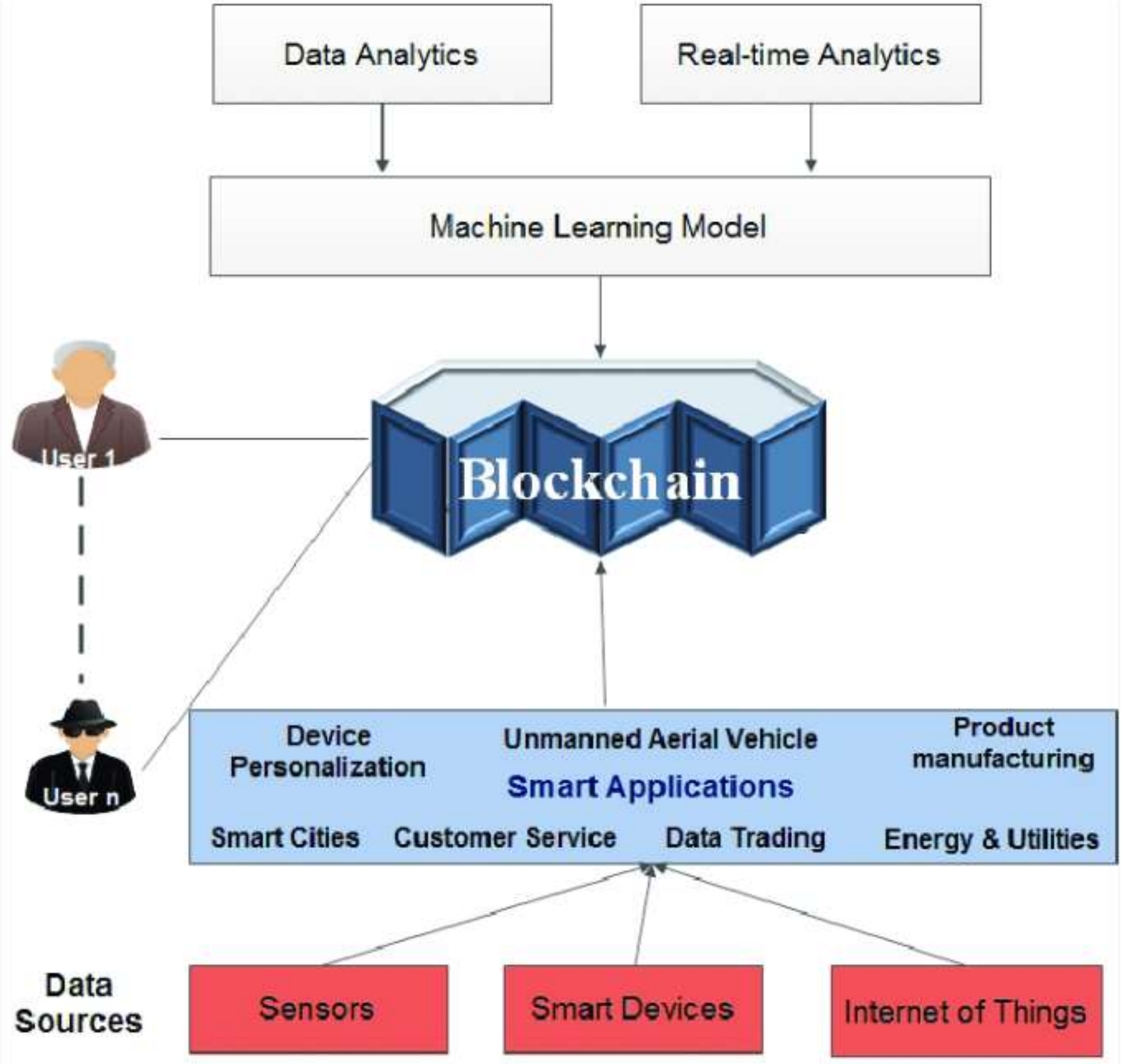
# ABSTRACT

With the increasing use of smart contracts in blockchain ecosystems, security risks such as exploits and vulnerabilities have become a major concern. Traditional auditing methods are often insufficient to detect sophisticated or novel attacks. This paper proposes a machine learning (ML)-based approach to detect malicious interactions with smart contracts by analyzing contract code, transaction data, and blockchain interactions. Using a combination of supervised and unsupervised learning techniques, we aim to identify harmful behaviors like reentrancy attacks and Ponzi schemes. Our approach enhances real-time detection, reduces false positives, and improves scalability to handle the growing number of deployed contracts. The results demonstrate that ML can effectively identify malicious contract interactions, providing a valuable tool for improving blockchain security and mitigating risks in decentralized applications.

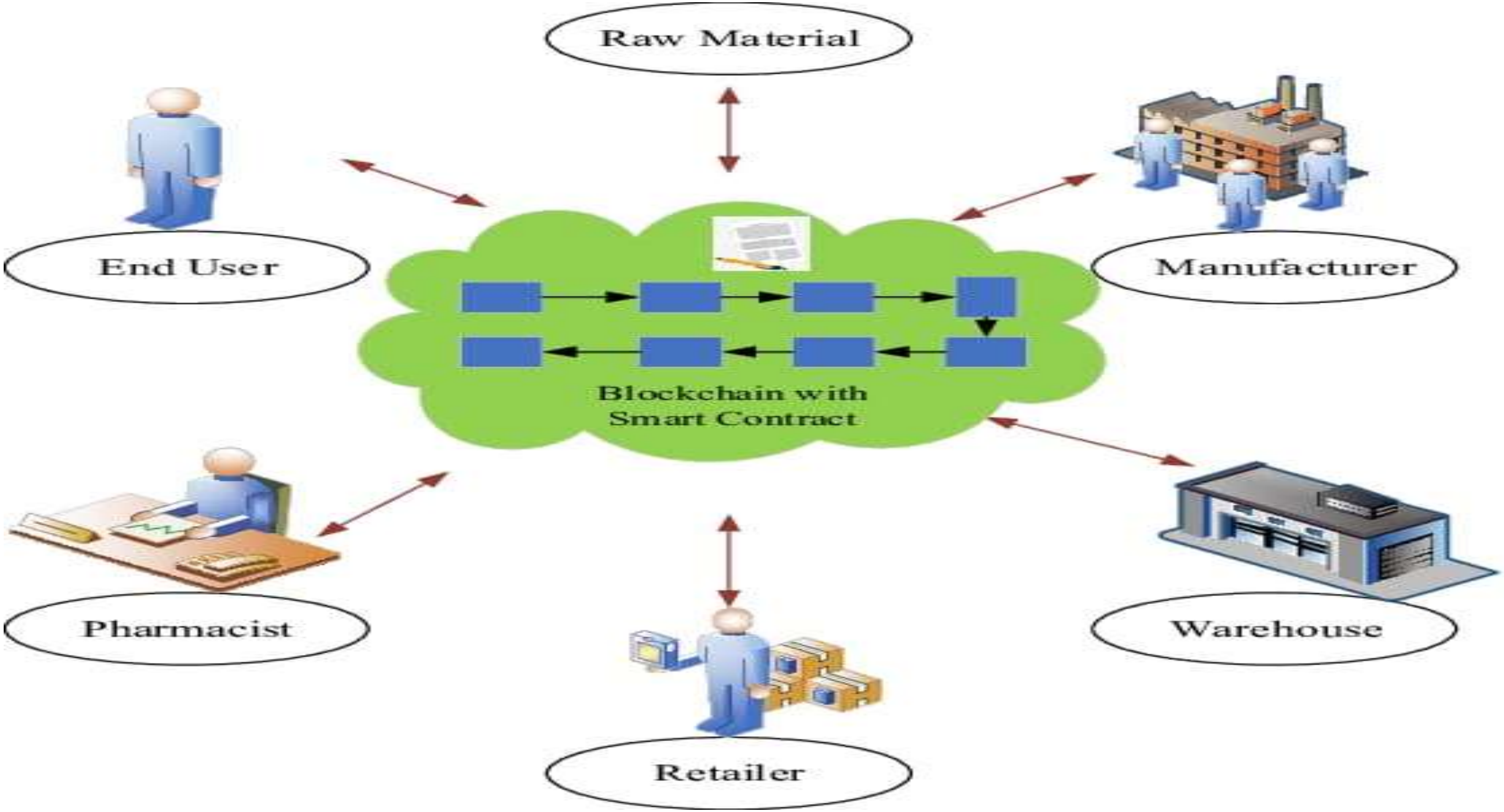
# LITERATURE SURVEY

| TITLE OF THE PAPER | AUTHOR (S) | PUBLISHER | PAPER GIST | TECHNOLOGY USED |
|---|---|---|---|---|
| Smart Contract Security and Vulnerabilities | 1. Loi luu<br>2. Vishwas patel | IEEE | improving security With best practices and automated tools. | Static Analysis Tool |
| best practices and automated tools. | 1. Saman Zahra<br>2. Joseph Bonneau | ACM | code audits, formal verification, and AI-driven detection to prevent vulnerabilities. | Cryptographic Techniques |
| Machine Learning for Anomaly and Malware Detection | 1. Vasant honavar<br>2. Alfredo cuzzocrea | ICML | detecting anomalies and malware, emphasizing their effectiveness | Supervised Learning Algorithm |
| Application of Machine Learning in Blockchain | 1. Qingli Tony<br>2. Hema gupta | ECCV | enhancing security, optimizing transaction processes | Deep Learning |
| Smart Contract Auditing Using Machine Learning | 1. Mohammad alrabaee<br>2. Hao chen | ICDM | automating and enhancing smart contract auditing | Blockchain |

# PROPOSED SYSTEM ARCHITECTURE



# EXISTING SYSTEM ARCHITECTURE



# SOFTWARE AND HARDWARE REQUIREMENTS

## HARDWARE

- CPU-Intel Core i7/i9 or AMD Ryzen 7/9)
- GPU-NVIDIA RTX 3060, 3080, 4090, A100, or V100
- RAM-Minimum 32 GB (recommended 64 GB or more)

## SOFTWARE

- Geth (Go Ethereum) or Parity Ethereum
- Web3.js or Web3.py
- Etherscan API
- Pip flask
- Pip pandas
- Pip numpy

# MODULES

1.Data Collection and Preprocessing Module

2.Feature Extreaction and Representation

3. Model Training and Detection Module

4.Evatuation and Testing

5. Real-time Monitoring and Mitigation Module

# SUMMARY OF MODULE-1

1. **Data Collection:** Gather transaction logs, smart contract code, and known malicious contract datasets.

2. **Preprocessing:** Extract features, label data as malicious or non-malicious, handle class imbalances, normalize/encode data, and perform static code analysis.

**Output:** A cleaned and labeled dataset ready for machine learning model training.

# SUMMARY OF MODULE-2

**1.Objective:**To extract meaningful features from smart contracts and represent them in a format suitable for machine learning models.

**2.Process:**Extract opcode sequences from the bytecode.Identify specific patterns or function calls indicative of vulnerabilities (e.g., reentrancy, overflow).Gather contract metadata (e.g., contract size, number of functions).

**Output:**Static Features: Opcode frequency, contract size, function counts.

Dynamic Features: Gas consumption patterns, state transition counts.

Graph Features: Adjacency matrix of function calls.

# SUMMARY OF MODULE-3

1. **Objective**: Develop and train machine learning models to detect malicious smart contract interactions.

2. **Process:** Select and train models (e.g., Random Forest, SVM, neural networks), validate performance, and use advanced techniques like unsupervised learning if needed.

**Output:** A trained model capable of identifying malicious contract behaviors.

# SUMMARY OF MODULE-4

**1.Objective:**Evaluate the model's predictive performance, including its ability to correctly classify malicious and non-malicious interactions.

**2.Process:**Use the labeled dataset, ensuring it is balanced and representative of both malicious and benign contracts. Divide it into training and testing sets, keeping the testing set unseen during training for unbiased evaluation.

**Output:**

Accuracy: Overall percentage of correctly classified smart contracts (malicious and benign).

Precision: Proportion of smart contracts predicted as malicious that are truly malicious.

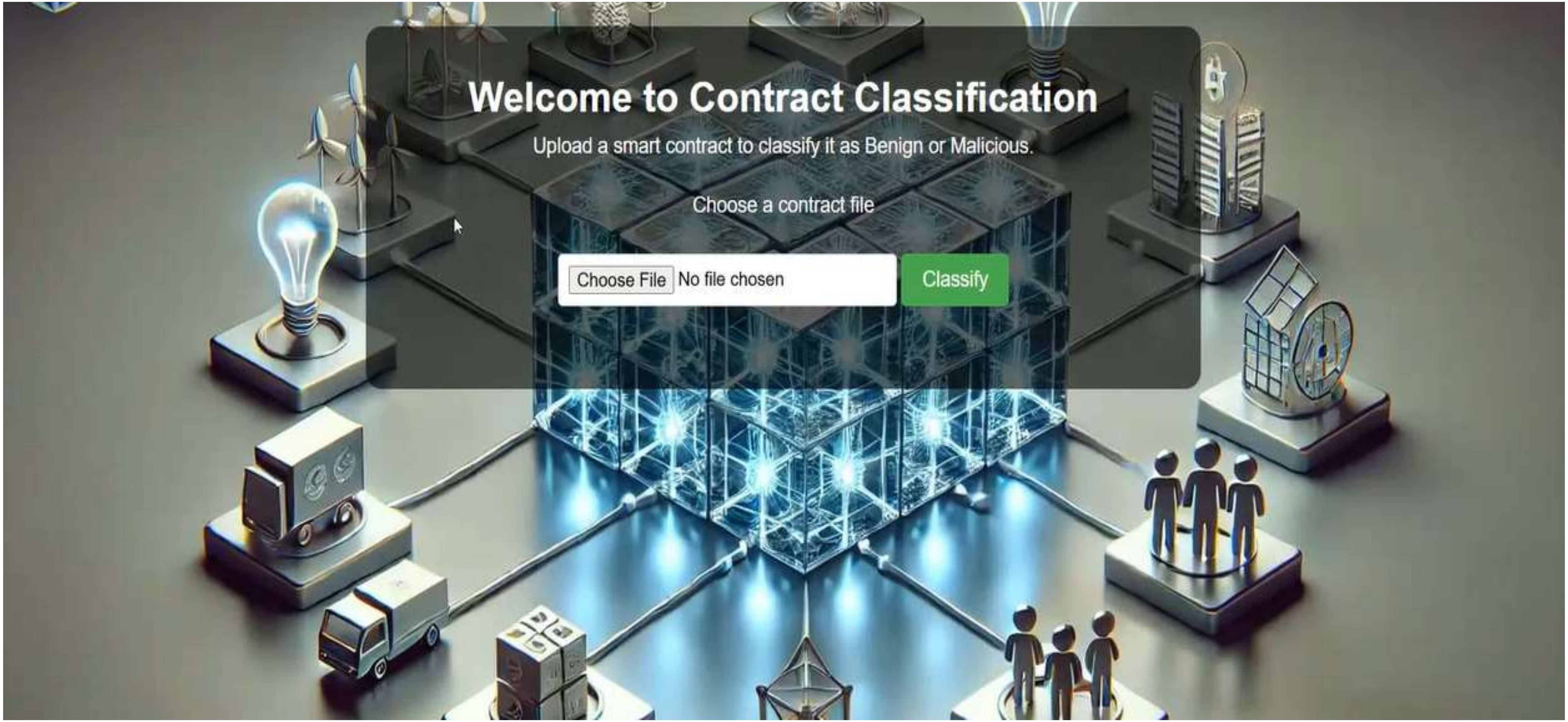
# SUMMARY OF MODULE-5

1. **Objective:** Deploy the trained model to monitor and manage smart contract interactions in real-time.

2**. Process:** Integrate the model for real-time detection, trigger alerts for suspicious activity, and implement automated responses or manual reviews.

**Output:** A system that continuously monitors transactions, detects malicious behavior, and takes preventive actions**.**

# RESULTS AND DISCUSSION

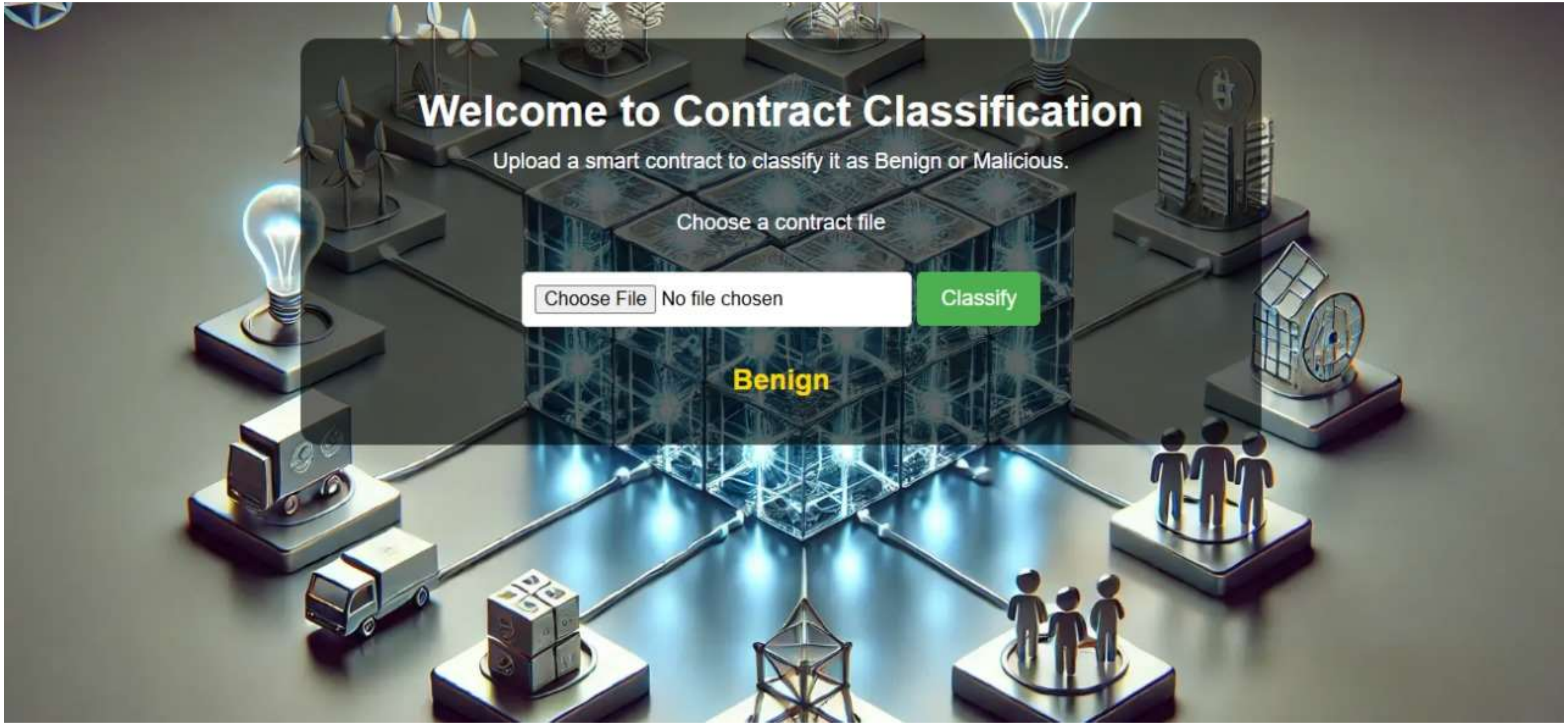


Welcome to Contract Classification

Upload a smart contract to classify it as Benign or Malicious.

Choose a contract file

# DELECTING BENIGN FILE



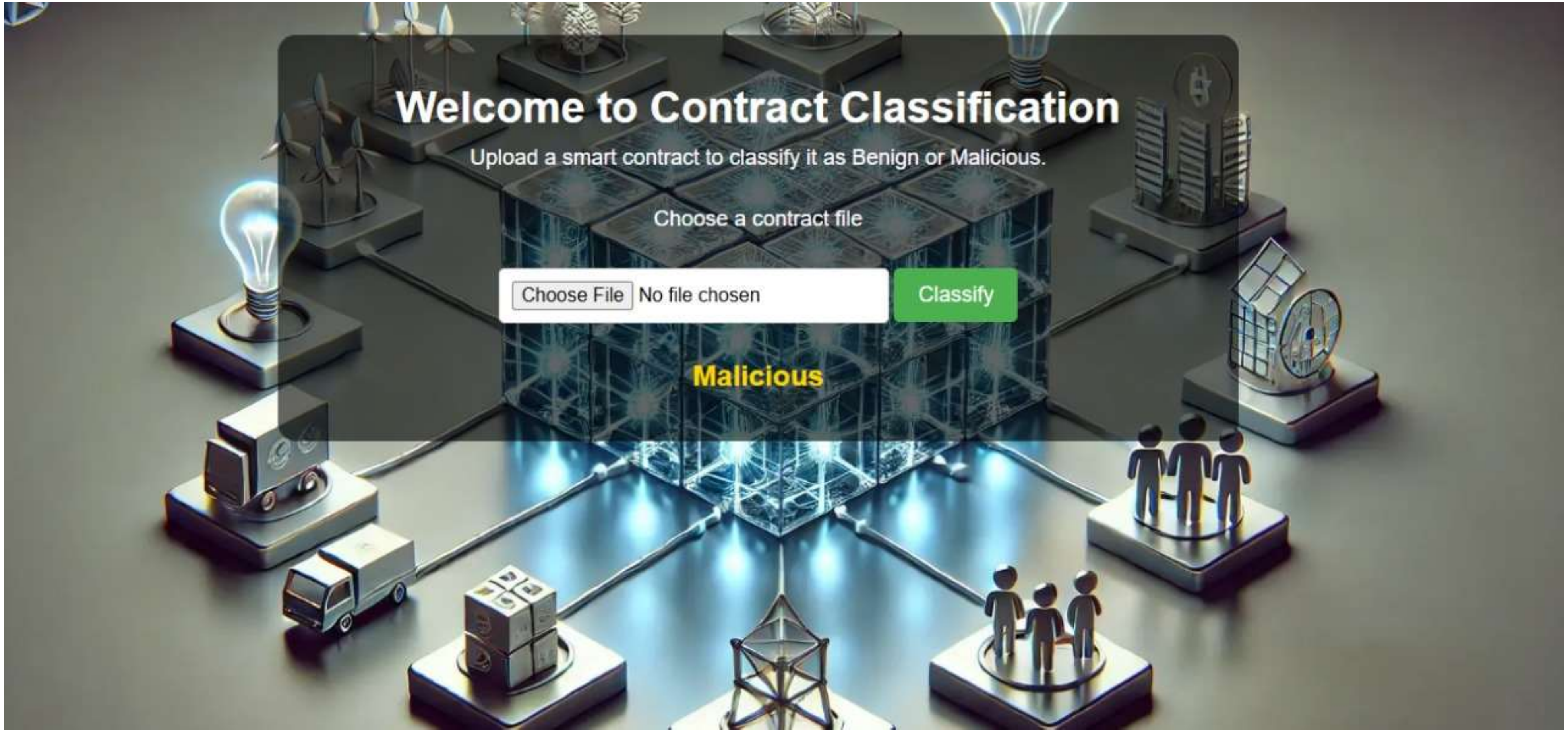
Welcome to Contract Classification

Upload a smart contract to classify it as Benign or Malicious.

Choose a contract file

Benign

# DETECTING MALICIOUS FILE



Welcome to Contract Classification

Upload a smart contract to classify it as Benign or Malicious.

Choose a contract file

Malicious

# CONCLUSION

- **Efficient Detection**: Machine learning effectively identifies malicious behaviors in smart contracts by analyzing patterns and anomalies.
- **Scalability**: It can handle large datasets in blockchain environments, automating threat detection and saving resources
- **Adaptability**: Machine learning models evolve over time, staying up-to-date with new malicious tactics.
- **Improved Accuracy**: Feature engineering enhances detection accuracy by focusing on relevant smart contract characteristics.
- **Challenges**: Data quality and model interpretability remain challenges, requiring high-quality data and explainable AI.

# THANK YOU