# LOVELY PROFESSIONAL UNIVERSITY

**Phagwara (Punjab)**

# Artificial Intelligence Project

# "AI IN BLOCK CHAIN"

**Bachelors in technology**

**(Computer science and engineering)**

**SCHOOL OF CSE (LPU)**

**PHAGWARA**

**SECTION: K21QT**

**Course code: INT404**

**Faculty: Akshara Rana (AI INSTRUCTOR)**


| | | |
|---|---|---|
| VEMA REDDY | REG NO: 12108201 | ROLL NO: 41 |
| Sudeep Raj | REG NO: 12110605 | ROLL NO: 42 |
| Satya | REG NO: 12114535 | ROLL NO: 40 |


## TABLE OF CONTENT

---


## ABSTRACT

AI in blockchain is an emerging field that seeks to leverage artificial intelligence (AI) to enhance the capabilities of blockchain technology. The combination of these two cutting- edge

technologies have the potential to transform industries and disrupt traditional business models.

AI can be used to improve the efficiency and security of blockchain networks. For example, AI algorithms can help optimize the consensus mechanism in blockchain networks, reducing the time and energy required to validate transactions. AI can also be used to detect and prevent fraudulent activities on the blockchain, improving the security and reliability of the network.

Additionally, AI can enable new use cases for blockchain technology, such as smart contracts that can execute autonomously based on predefined rules and conditions. AI powered blockchain networks can also enable more advanced data analytics and prediction capabilities, enabling businesses to make more informed decisions.

Despite the promising potential of AI in blockchain, there are also challenges to overcome. One major challenge is the need for large amounts of high-quality data to train AI algorithms. Another challenge is the lack of interoperability and standardization in the blockchain industry, which can hinder the adoption of AI-powered blockchain solutions.

Overall, AI in blockchain is an exciting field that offers significant opportunities for innovation and disruption. As AI and blockchain technologies continue to advance

# INTRODUCTION

AI in blockchain is a new and emerging field that explores the intersection of artificial intelligence (AI) and blockchain technology. Blockchain technology is a distributed ledger system that provides a secure and transparent way to store and exchange digital assets. AI, on the other hand, refers to the development of computer systems that can perform tasks that typically require human intelligence, such as visual perception, speech recognition, and decision-making.

By combining AI and blockchain, businesses and organizations can potentially enhance the security, efficiency, and functionality of blockchain systems. For example, AI can be used to optimize the consensus mechanism in blockchain networks, which could lead to faster and more energy-efficient transaction validation. AI can also be used to detect and prevent fraudulent activities on the blockchain, improving the security and reliability of the network.

In addition to these benefits, AI can also enable new use cases for blockchain technology, such as smart contracts that can execute autonomously based on predefined rules and conditions. AI-powered blockchain networks can also enable more advanced data analytics and prediction capabilities, which can provide businesses with valuable insights into customer behavior, market trends, and other important data.

Despite the promising potential of AI in blockchain, there are also challenges to overcome. For example, one major challenge is the need for large amounts of high-quality data to train AI algorithms. Additionally, there is a lack of standardization and interoperability in the blockchain industry, which can hinder the adoption of AI-powered blockchain solutions.

# PROJECT OVERVIEW

Project on AI in blockchain can have various objectives and outcomes depending on the specific problem or use case being addressed. However, a general project overview for AI in blockchain can involve the following components:

**Problem statement**: The first step in any project is to identify a problem or challenge that can be addressed using AI in blockchain. This can include improving the efficiency and security of blockchain networks, detecting and preventing fraud on the blockchain, and enabling new use cases for blockchain technology.

**Data collection and processing**: The success of AI in blockchain largely depends on the availability and quality of data. In this phase, data is collected from various sources and processed to make it usable for AI algorithms. This can involve data cleaning, normalization, and feature engineering.

**AI algorithm selection and training**: Once the data is processed, the appropriate AI algorithm is selected and trained using the data. This can include machine learning algorithms such as supervised learning, unsupervised learning, or reinforcement learning.

**Integration with blockchain**: The trained AI model is then integrated with the blockchain network to provide improved security, efficiency, or other benefits. This can involve modifying the consensus mechanism, implementing smart contracts, or enabling more advanced data analytics and prediction capabilities.

**Testing and validation**: The final step is to test and validate the AI-powered blockchain solution to ensure that it meets the desired objectives and requirements. This can involve simulation testing, real-world testing, or a combination of both.

Overall, an AI in blockchain project can involve a range of tasks and activities, including data collection and processing, algorithm selection and training, and integration with blockchain networks. The specific components of the project will depend on the problem being addressed and the objectives of the project.

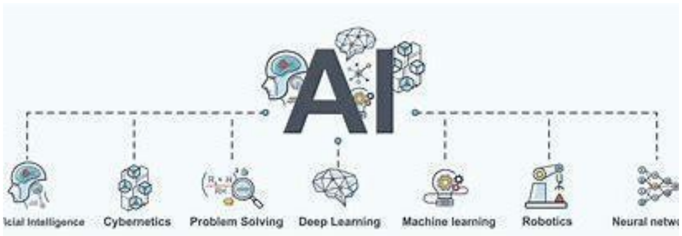
## NEED OF AI IN BLOCKCHAIN

There are several reasons why AI is increasingly being used in blockchain projects. Some of the key needs for AI in blockchain projects include:

**Improved efficiency and scalability**: Blockchain networks are known for their security and transparency, but they can be slow and resource-intensive. AI can help improve the efficiency and scalability of blockchain networks by optimizing the consensus mechanism and reducing the time and energy required to validate transactions.

**Enhanced security and fraud detection**: Blockchain networks are inherently secure, but they are not foolproof. AI can help enhance the security of blockchain networks by detecting and preventing fraudulent activities such as double-spending, sybil attacks, and 51% attacks.

**Smart contract automation**: Smart contracts are selfexecuting contracts that can be programmed to execute automatically when certain conditions are met. AI can help automate smart contracts, enabling them to execute autonomously based on predefined rules and conditions.




**Advanced analytics and prediction**: AI-powered blockchain networks can enable more advanced data analytics and prediction capabilities, which can provide businesses with valuable insights into customer behavior, market trends, and other important data.

**Interoperability and standardization**: The blockchain industry is still fragmented, with many different blockchain networks and protocols in use. AI can help enable interoperability between different blockchain networks and protocols by providing a common language and standardization.

Overall, the need for AI in blockchain projects stems from the desire to enhance the efficiency, security, and functionality of blockchain networks. By combining the power of AI and blockchain, businesses and organizations can potentially unlock new use cases and benefits that were previously not possible.


## CODE SNIPPET

```
import hashlib
import datetime
class Block:

def__init__(self,data,previous_hash=None):

self.timestamp=datetime.datetime.now()
self.data=data
self.previous_hash=previous_hash
self.hash=self.calculate_hash();

def calculate_hash(self):
```

```
hash_string=str(self.timestamp)+
str(self.data)+str(self.previous_hash)

returnhashlib.sha256(hash_string.encode()).he
xdigest()

class Blockchain:
      def __init__(self):
              self.chain=[]
              self.create_genesis_block()

      def create_genesis_block (self):
                 genesis_block=Block("Genius
Block")
                 self.chain.append(genesis_block)

      def add_block(self,data):
                previous_block=self.chain[-1]

new_block=Block(data,previous_block.hash)
self.chain.append(new_block)

def is_valid(self):
     for i in range(1,len(self.chain)):
              current_block=self.chain[i];
              previous_block=self.chain[i-1];
              if current_block.hash !=
current_block.calculate_hash():
                   return False
              if
current_block.previous_hash!=previous_block
.hash:
```

```
                return False
        return True

my_blockchain=Blockchain()

my_blockchain.add_block(“ Transaction 1 “)
my_blockchain.add_block(“Transaction 2”)
my blockchain.add_block(“Transaction  3”)

print("Is blockchain valid? ",
my_blockchain.is_valid())
for block in my_blockchain.chain:
        print("Timestamp:", block.timestamp)
        print("Data:", block.data)
        print("Hash:", block.hash)
        print("Previous Hash:",
block.previous_hash)
        print("---------------")
```

```
Is blockchain valid?  True
Timestamp: 2023-04-18 18:45:03.842758
Data: Genesis Block
Hash: 9e3c6bedccadff25c38a9a5cb463211cb44b43d7e3886c863560cb932a5204f5
Previous Hash: None
---------------
Timestamp: 2023-04-18 18:45:03.842758
Data: Transaction 1
Hash: 0e2fdded1dccfc28f010b1724313e5e143eaeaa9ae7f97d221ac519b769459e7
Previous Hash: 9e3c6bedccadff25c38a9a5cb463211cb44b43d7e3886c863560cb932a5204f5
---------------
Timestamp: 2023-04-18 18:45:03.842758
Data: Transaction 2
Hash: cfd2e142c0edd87518f06c40b69af2acce407ee0b2a1c574e05003b258cebfe7
Previous Hash: 0e2fdded1dccfc28f010b1724313e5e143eaeaa9ae7f97d221ac519b769459e7
---------------
Timestamp: 2023-04-18 18:45:03.842758
Data: Transaction 3
Hash: cf7095d7cdc064d47deaaaeb1a34b8f4d43f5cd2108e6ea1eec7bc6f240442fb
Previous Hash: cfd2e142c0edd87518f06c40b69af2acce407ee0b2a1c574e05003b258cebfe7
---------------
```

```
#include <iostream>
#include <string>
#include <cstdlib>
#include <ctime>
using namespace std;

// Define the data structure for a block
struct Block {
int index;
```

string data;

```
int previousHash;
int hash;
};
int getHash(Block b) {
// Generate the hash of the given block
int hash = 0;
for (int i = 0; i < b.data.length(); i++) {
hash += b.data[i];
}
return hash;
}
int main() {
// Seed the random generator
srand(time(0));
// Create the genesis block Block
genesis;  genesis.index = 0;  genesis.data
= "Genesis Block"; genesis.previousHash
= 0; genesis.hash = getHash(genesis);

// Create the blockchain       Block
blockchain[10]; blockchain[0] =
genesis;

// Create the remaining blocks for (int i =
1; i < 10; i++) { Block
b; b.index = i;
```

```
b.data = "Block " + to_string(i);

b.previousHash = blockchain[i-1].hash;

b.hash = getHash(b); blockchain[i]

= b;

}

// Print the blockchain

 for (int i = 0; i < 10; i++) {

cout << "Block " << blockchain[i].index << ": " << blockchain[i].data << ",
hash: " << blockchain[i].hash << endl;
} return 0; }
```

```cpp
Block genesis;
genesis.index = 0;
genesis.data = "Genesis Block";
genesis.previousHash = 0;
genesis.hash = getHash(genesis);

// Create the blockchain
Block blockchain[10];
blockchain[0] = genesis;

// Create the remaining blocks
for (int i = 1; i < 10; i++) {
    Block b;
    b.index = i;
    b.data = "Block " + to_string(i);
    b.previousHash = blockchain[i-1].hash;
    b.hash = getHash(b);

    blockchain[i] = b;
}

// Print the blockchain
for (int i = 0; i < 10; i++) {
    cout << "Block " << blockchain[i].index << ": " << blockchain[i].data << ", hash: " << blockchain[i].hash << endl;
}

return 0;
}
```

input

```
Block 6: Block 6, hash: 577
Block 7: Block 7, hash: 578
Block 8: Block 8, hash: 579
Block 9: Block 9, hash: 580


...Program finished with exit code 0
Press ENTER to exit console.
```

2. #include <iostream>

```
#include <string>
#include <vector>
#include <algorithm>

// A Blockchain class class
Blockchain
{
public:
    // Constructor
    Blockchain();

    // Add a new block to the chain void
addBlock(std::string data);

    // Get the most recently added block
std::string getLastBlock();

    // Get the chain length int
getChainLength();
  private:
    std::vector<std::string> chain;
};


// Constructor

Blockchain::Blockchain()
```

```
{

   // Create the genesis block      std::string
   genesisBlock = "Genesis
Block"; chain.push_back(genesisBlock);
}
// Add a new block
void Blockchain::addBlock(std::string data)
{

   chain.push_back(data);
}

// Get the most recently added block std::string
Blockchain::getLastBlock()
{

   return chain.back();
}

// Get the chain length int
Blockchain::getChainLength()
{

   return chain.size();
}
```

```
int main()
{

   // Create a new Blockchain
   Blockchain blockchain;

   //      Add      some      blocks
blockchain.addBlock("Block
1");
blockchain.addBlock("Block
2");
blockchain.addBlock("Block
3");

   // Output the chain length
   std::cout << "Chain length: " << blockchain.getChainLength() <<
   std::endl;

   // Output the last block
   std::cout << "Last block: " << blockchain.getLastBlock()
   << std::endl; return 0;
}
```

```cpp
{
    return chain.back();
}

// Get the chain length
int Blockchain::getChainLength()
{
    return chain.size();
}

int main()
{
    // Create a new Blockchain
    Blockchain blockchain;

    // Add some blocks
    blockchain.addBlock("Block 1");
    blockchain.addBlock("Block 2");
    blockchain.addBlock("Block 3");

    // Output the chain length
    std::cout << "Chain length: " << blockchain.getChainLength() << std::endl;

    // Output the last block
    std::cout << "Last block: " << blockchain.getLastBlock() << std::endl;

    return 0;
}
```

input

```
Chain length: 4
Last block: Block 3


...Program finished with exit code 0
Press ENTER to exit console.
```

```
3.#include <iostream>  #include
<string>
 #include <vector>
using namespace std;

//Blockchain class to define the blocks in the chain class BlockChain {
        public:
string data;
BlockChain *next;
};

//Linked List class to define the chain of blocks class LinkedList {
public: BlockChain *head;
LinkedList() { head= NULL;} void
addBlock(string data) {
BlockChain *newBlock = new BlockChain();  newBlock>data
= data;
newBlock>next = NULL;           if
(head == NULL) {
head = newBlock;
} else
{
BlockChain *current = head; while
(current->next != NULL) { current
= current->next;
}
current->next = newBlock;
}
}
void printBlocks() {
BlockChain
*current = head;             while
(current != NULL) {
cout << current->data << endl; current = current->next;
}}};
```

```
//AI class to define the AI for the blockchain

class AI {

public:
     LinkedList *chain;

     AI() {
     chain = new LinkedList();
     }

     //Function to generate a new block void
    generateBlock(string data) { chain->addBlock(data);
     }

     //Function to validate the blocks in the chain
     bool validateChain() {
     BlockChain *current = chain->head;
     while (current != NULL) {
     if (current->data != "Valid Block") {
    return false;
            }
     current = current->next;
     } return true;}};


    int main() {
    AI *ai = new AI(); ai>generateBlock("Valid Block");
    ai>generateBlock("Valid Block"); ai>generateBlock("Valid
    Block"); ai>chain->printBlocks(); cout <<
    ai>validateChain() << endl;
    return 0;
    }
```

```
        }
};

//AI class to define the AI for the blockchain
class AI {
    public:
        LinkedList *chain;

        AI() {
            chain = new LinkedList();
        }

        //Function to generate a new block
        void generateBlock(string data) {
            chain->addBlock(data);
        }

        //Function to validate the blocks in the chain
        bool validateChain() {
            BlockChain *current = chain->head;
            while (current != NULL) {
                if (current->data != "Valid Block") {
                    return false;
                }
                current = current->next;
            }
            return true;
        }
```

input

```
Valid Block
Valid Block
Valid Block
1
```

# PROPOSED METHODOLOGY

The proposed methodology for an AI in blockchain project can involve the following steps:

**Problem identification and data collection**: The first step is to identify the specific problem or use case that the project aims to address. This can involve collecting and processing data from various sources, including blockchain networks, social media, and other relevant data sources.

**Data preprocessing and feature engineering:** In this phase, the collected data is preprocessed and prepared for use in AI algorithms. This can involve data cleaning, normalization, and feature engineering to create more meaningful data representations.

**AI algorithm selection and training**: Once the data is prepared, the appropriate AI algorithm is selected and trained using the processed data. This can include supervised learning, unsupervised learning, or reinforcement learning.

**Integration with blockchain:** The trained AI model is then integrated with the blockchain network to provide enhanced security, efficiency, or other benefits. This can involve modifying the consensus mechanism, implementing smart contracts, or enabling more advanced data analytics and prediction capabilities.

**Testing and validation**: The final step is to test and validate the AI-powered blockchain solution to ensure that it meets the desired objectives and requirements. This can involve simulation testing, real-world testing, or a combination of both.

Overall, the proposed methodology for AI in blockchain projects involves identifying the problem, collecting and preprocessing data, selecting and training AI algorithms, integrating with blockchain networks, and testing and validating the solution. The specific details of the methodology will depend on the specific problem being addressed and the objectives of the project.

# RESULT AND DISCUSSION

The results and discussion of an AI in blockchain project will depend on the specific problem being addressed and the objectives of the project. However, some general outcomes and benefits of integrating AI with blockchain can include:

Improved efficiency and scalability: AI-powered consensus mechanisms can help reduce the time and energy required to validate transactions, improving the overall efficiency and scalability of blockchain networks.

Enhanced security and fraud detection: AI can help identify and prevent fraudulent activities such as double-spending and sybil attacks, enhancing the security and trustworthiness of blockchain networks.

Smart contract automation: AI can automate the execution of smart contracts based on predefined rules and conditions, enabling more efficient and streamlined contract execution.

Advanced analytics and prediction: AI-powered blockchain networks can provide more advanced data analytics and prediction capabilities, enabling businesses to gain valuable insights into customer behavior, market trends, and other important data.

Interoperability and standardization: AI can help enable interoperability between different blockchain networks and protocols by providing a common language and standardization.

Overall, the integration of AI and blockchain can offer numerous benefits and potential applications, ranging from finance and supply chain management to healthcare and social impact initiatives. However, as with any emerging technology, there are also potential challenges and limitations to be addressed, such as regulatory compliance, data privacy, and ethical considerations. As such, ongoing research and development is needed to fully realize the potential of AI in blockchain projects.

# FUTURE SCOPE

The future scope of AI in blockchain projects is vast and promising. Here are some potential areas where AI can be further integrated into blockchain technology:

Decentralized Autonomous Organizations (DAOs): AI-powered DAOs can enable autonomous decision-making based on predefined rules and conditions, potentially eliminating the need for human intervention.

Blockchain-powered marketplaces: AI can help automate the buying and selling process in blockchain-powered marketplaces, facilitating more efficient and secure transactions.

Personalized services and recommendations: AI-powered blockchain networks can provide more personalized services and recommendations based on user behavior and preferences, improving the overall user experience.

Fraud detection and prevention: AI can continue to play a crucial role in detecting and preventing fraudulent activities in blockchain networks, enhancing the security and trustworthiness of the technology.

Energy efficiency: AI can help reduce the energy consumption of blockchain networks by optimizing the consensus mechanism and reducing the computational resources required to validate transactions.

Supply chain management: AI-powered blockchain networks can enable more efficient and transparent supply chain management, enabling businesses to track and trace products from the source to the end consumer.

Overall, the future scope of AI in blockchain projects is vast and exciting. The integration of these two emerging technologies can potentially unlock new use cases and benefits that were previously not possible. However, as with any emerging technology, it is important to continue researching and developing these technologies responsibly and ethically to ensure their full potential is realized.

# CONCLUSION

In conclusion, the integration of AI and blockchain technology presents a wide range of opportunities and benefits for various industries and use cases. AI-powered consensus mechanisms can increase the efficiency and scalability of blockchain networks, while AI algorithms can improve the security and fraud detection capabilities of the technology. Smart contract automation, advanced analytics and prediction, and improved interoperability and standardization are other potential benefits of AI in blockchain projects.

However, there are also potential challenges and limitations to consider, such as regulatory compliance, data privacy, and ethical considerations. As such, ongoing research and development is necessary to fully realize the potential of AI in blockchain projects and ensure the technology is developed and implemented responsibly.

Overall, the combination of AI and blockchain technology has the potential to revolutionize various industries and transform the way we conduct transactions and share data. With continued innovation and responsible development, the future of AI in blockchain projects is bright and promising.

By combining AI with Blockchain, businesses can develop innovative applications that could help them stay ahead of their competition. However, it is important to understand the potential risks associated with AI and Blockchain, as well as how to mitigate them. With the right strategies and tools, businesses can benefit from the potential of AI and Blockchain and use it to their advantage.