Decrypting Payment Systems Using Reinforcement Learning Neural Networks

Leila Zaghari[1], Abbas Toloie Eshlaghy[2]

[1]Master student in Information Technology Management, Faculty of Management and Economics, Science and Research Branch, Islamic Azad University, Tehran, Iran

[2]Professor, Department of Information Technology Management, Faculty of Management and Economics, Science and Research Branch, Islamic Azad University, Tehran, Iran

DOI: https://dx.doi.org/10.47772/IJRISS.2024.806140

Received: 15 May 2024; Revised: 08 June 2024; Accepted: 14 June 2024; Published: 12 July 2024

ABSTRACT

The financial industry has experienced significant transformations with the advancement of information technology (IT); these developments include cost reduction and quality improvement of financial information services and creating new horizons and dynamic customer approach in utilizing banking services along with quality improvement of traditional services businesses. These changes would require specific attention to data transmission. Data encryption is one of the methods that can be utilized to protect data during its transmission. Neural network technology improves the speed, accuracy, and security of payment systems and increases users' trust in these systems. Additionally, reinforcement neural networks can design and implement fraud detection algorithms with high accuracy. This study examines neural network techniques, machine learning capabilities, payment system decryption, and security enhancement to prevent fraud and protect the financial information of customers. The main issue in this research is to improve the performance of payment systems by improving the speed, accuracy, efficiency, and security, maintaining the confidentiality of bank transaction data through encryption and decryption of transaction amounts as well as modeling using reinforcement neural networks. This research can be considered an important step towards improving electronic payment technologies and increasing the confidence of customers regarding the security of bank transactions to achieve these goals several million data sets of payment transactions underwent data preprocessing, normalization, and transformation for neural network compatibility, followed by encryption and decryption.

Keywords: Decryption; Encryption; Cryptanalysis; Machine Learning; Reinforcement Learning.

INTRODUCTION

In today's digital era, convenient online transactions require efficiency and security. The Internet has become an integral part of human life, and technology is advancing at an unprecedented speed. However, Reviewing the literature reveals online financial transactions, e-commerce, payment gateways, and communication methods are vulnerable to fraud and forgery. As a result, secure methods for information transmission are necessary to ensure privacy and prevent theft. Data security has always been essential, and it is even more critical now that the data is transmitted through unsecured channels. Encryption is a method used to secure digital data during communication, which transforms or maps data in such a way that it cannot be recognized by a third party(Verma et al., 2024).

Decryption is the process of converting an encrypted message back to its original form using the same set of rules that were used to create the encrypted message.Encryption algorithms use a secret key to turn plain text into an unintelligible form called ciphertext. A decryption algorithm, which is the reverse of the encryption algorithm, transforms the ciphertext into readable text with the help of the secret key. Encryption methods are used in computer and communication systems to maintain privacy and authenticate identity (Verma et al., 2024). Banking payment systems are critical components of financial and economic affairs. These systems include credit/debit card systems, interbank transaction systems, electronic transaction systems, and QR code-based payment systems (Lakew, 2023). Encryption and decryption of data are two main components of cryptography. Plain text refers to data that is easily understandable, while encryption is a process that transforms plain text into an unreadable form. Decryption is the process of reversing the encryption using a secret key to return the message to its original form (Amarnadh et al., 2022). Neural networks can be a useful tool when it comes to encryption. These networks are designed to simulate the functioning of the human brain, providing the capability to perform complex computations quickly and efficiently (Malyadri et al., 2022).

Reinforcement learning is a process where an agent and an environment interact with each other. The Markov decision process is the basis for this process (Sutton & Barto, 2018). The agent takes actions that change the environment's state, and the environment rewards the agent for its actions (Sutton & Barto, 2018). The agent learns to take better actions in specific situations in the environment as it interacts with the environment iteratively (Alavizadeh et al., 2022).Deep learning is a type of machine learning that uses neural networks to recognize patterns in data (Bengio et al., 2021). Deep reinforcement learning is a vital aspect of general artificial intelligence and has been successfully used in various fields, such as gaming and robotics (Khadivi et al., 2023).Reinforcement learning is a method of using neural networks to model problems and is used for unsupervised training of network parameters (Lin et al., 2023).The reinforcement learning component allows the agent to explore different strategies and learn from the resulting rewards and penalties(Fang et al., 2023).

This research aims to use neural network techniques and the power of machine learning to decrypt payment systems and enhance security, prevent fraud, and protect customer financial information. Deep reinforcement learning can analyze payment data and financial behaviors, recognize patterns of regular customers and respond to fraudulent changes. By combining historical payment data, reinforcement learning, and neural network training, effective and efficient decryption operations will be performed using deep reinforcement learning. Additionally, this research project aims to enhance the reliability and credibility of results by using real data, provide practical solutions for implementing deep reinforcement neural networks in banking payment systems, and improve decision-making processes and system performance in financial resource allocation, risk management, and payment decisions through data analysis and pattern identification. The project timeline is from March 20, 2023, to March 20, 2024.

This paper has the following contributions:

- For the first time, encryption and decryption of payment transaction amounts in the banking systems are presented in Iran.
- Our method can utilize reinforcement neural networks for decrypting the amount field in payment systems.
- Our method can increase security and reduce potential risks of data theft.
- Our method can increase the speed of processing completed payment transactions and reduce response times.
- A case study is given.

This section of the paper provides a review of the definitions and concepts of payment system decryption and deep reinforcement neural networks. The next sections of this paper are as follows: The literature review and Methodology are presented in sections 2 and 3 respectively.

The case study and Discussion are presented in Section 4. The conclusion, limitations, and recommendations are presented in section 5.

LITERATURE REVIEW

Cryptography and Decryption

To protect sensitive data, encryption and decryption operations rely on key management. It is essential to understand how these operations work. Both encryption and decryption keys must be securely stored to ensure the security of data. It should be difficult for any unauthorized parties to access these keys or the encryption/decryption process. The algorithms used in the encryption and decryption process should guarantee the safekeeping of sensitive data and establish its security(Vishwakarma et al., 2021). In public-key cryptography systems, the message's confidentiality, authentication, and non-repudiation are examined and ensured through a series of steps. Finally, the recipient verifies the message's authenticity using the sender's public key(Daniel et al., 2021).Malallah et al. (2023) proposed a method that uses cloud storage services to ensure the security of confidential banking documents. The method achieves data security by transforming the text into a QR code image and applying encryption before transferring it to cloud storage. Additionally, the method utilizes a security key to further enhance the level of protection.

Payment Systems

Banking payment systems are tools and processes that facilitate financial transactions through bank accounts, these systems include online transactions, fund transfers, electronic payments, transactions via bank cards, and other financial instruments.They use various tools and technologies to conduct secure and efficient financial transactions, playing a fundamental role in economic and commercial activities (Hassan et al., 2020). Some of the primary tools and technologies used in electronic payments are credit and debit cards, online payment gateways, electronic fund transfers, and payments based on digital currency(Kadjie et al., 2023).

Decryption in the payment systems

Encryption and decryption play a crucial role in payment systems. They involve touse of algorithms and encryption techniques to transform financial information in a way that only authorized individuals can access. Keeping financial information secure and private when transferring it over the internet or other media is highly important. To accomplish this, secure methods are used that employ algorithms and private keys to encrypt sensitive information. This ensures that financial data is being securely exchanged in a way that is only accessible to those authorized to access it (Ogheneruemu & Taiye, 2023). Encryption in payment systems guarantees that financial information remains confidential and only accessible to authorized individuals(Razumov et al., 2023).Encryption is being used in payment systems to protect the financial and personal information of individuals during transmission and storage. According to Zhou et al. (2022), the primary purpose of encryption is to ensure the security of sensitive data. In their research on banking security applications, Pathak et al. (2020) proposed a method that uses an image as the authentication key to address this issue. The proposed method involves creating an image using visual cryptography and other image-processing techniques to encrypt the authentication parameters.

The Impact of Payment System Decryption in Banks

Banks need to securely maintain sensitive information like account numbers, passwords, and transaction details. This can be achieved through payment systems that use encryption and security technologies (Ogheneruemu & Taiye, 2023).Research by Mittal et al. (2021) found that encrypting bank data as text was the best way to preserve the privacy of the information. They explored how secure computations on encrypted texts could reduce the risk of data breaches. This approach has become increasingly useful in exploring the potential benefits of privacy-enhancing techniques, particularly in financial institutions where security is of utmost importance.

Deep and Reinforcement Neural Networks

Deep neural networks are capable of identifying anomalies in virtual communication networks by recognizing patterns in the data and reconstructing the network for better accuracy. However, these models require more training time for complex data, which is a significant challenge. To address this, there is a need for innovative approaches that can reduce the training time without compromising accuracy in detecting anomalies (Kathamuthu et al., 2022). Reinforcement neural networks learn and make decisions through reward and punishment mechanisms to help achieve specific goals. Reinforcement neural networks encompass concepts such as reinforcement learning algorithms like Q-learning, reward functions, value functions, the notion of interaction with the environment, action-value functions, temporal reward functions, and concepts related to machine learning and optimization methods(Khadivi et al., 2023).

Reinforcement of Neural Networks with Payment Systems

Reinforcement neural networks play a crucial role in payment systems. They can be used for various tasks such as pattern recognition, data-driven decision-making, customer behavior analysis, and forecasting financial transactions. For example, reinforcement neural networks are employed to detect fraud in online transactions and credit card usage. They can identify unusual patterns and reduce fraud-related risks, thus ensuring enhanced security. Besides, these networks are also used to optimize payment processing workflows and improve product recommendation systems (Matsunami et al., 2021).Reinforcement neural networks are a crucial tool that can be used to enhance the decision-making process in payment systems. These networks can learn from experience and improve continuously. Therefore, they can be effectively utilized to improve payment systems and the delivery of financial services (Meng & Khushi, 2019).

Reinforcement of Neural Networks in Decrypting Payment Systems

Reinforcement neural networks are important in the decryption of payment systems. These networks can analyze processed data within payment systems and make decisions based on that data. They act as decryption sensors and use mathematical and engineering rules to detect and decide on issues such as payment fraud(Khadivi et al., 2023). Pal and Datta (2022) have developed a new neural network-based encryption algorithm that is efficient, portable, dynamic, and simple. They used a multi-substitution balanced tree machine approach to generate dynamic single-digit series keys. Brown and Davis's (2020) article, focuses on the use of deep reinforcement learning techniques in cryptanalysis for decrypting encrypted messages.

In the study conducted by Yeow and Ngin 2023, different encryption methods were compared in neural networks, and it was proposed that data weighting using text and image encryption algorithms can improve the security of protected data.In 2023, Tsmots and colleagues used neural network technology to encrypt, protect, and transfer data. They explained that the use of neural networks for encryption and decryption can be applied to all types of data, not just image processing. Their findings suggest that this technology offers a secure and trustworthy method of transferring data.Amarnadh and his team published a research article in 2022 that discussed the training of neural networks using keys and plain text. They explained that for neural network training, weights are calculated between the keys and the network and used as encrypted data. The encrypted keys and text are then used to train the neural network.

METHODOLOGY

Research Method

In this section, we will discuss the various stages involved in conducting research. Firstly, we will explain the research method, overall design, and the main stages and steps of conducting research. Next, we will present details regarding the statistical population, determination of sample size, sampling method, operational tools, and modeling. The research method we will be using involves reinforcement neural networks for decrypting payment systems. This method primarily relies on a reinforcement learning approach, which is a computationally describable approach.In this approach, reinforcement neural networks learn from input data using reward and penalty mechanisms and strive to demonstrate improved performance. Through repetitive learning processes and interaction with the environment, these networks can enhance their performance and perform more complex tasks (Vimal et al., 2021).

Research Implementation Process

The primary objective of this research is to improve the encryption and decryption of payment systems by utilizing new encryption algorithms and reinforcement neural networks. This will enhance encryption performance, ensure payment information security, and increase the accuracy and precision of operations by using optimal algorithms. During the research process, we will explain all the fundamental steps taken to conduct the research transparently. We will also clarify the rules and principles used in the research. Our research is applied research with a descriptive methodology based on modeling.Here are the five steps used in this paper:

Step 1: Understanding Encryption and Decryption and Grasping the Issue

Data security has always been important, but it is now more crucial than ever, especially when data is being transmitted through insecure channels. To securely communicate digital data, one of the methods used is data encryption. In encryption, data is being transformed or mapped in a way that a third party cannot interpret the communication. This ensures data security. Information security is particularly critical in various areas, especially in banking systems (Gokcay & Tora, 2024).Asymmetric encryption or public-key cryptography is a commonly used method in payment systems. This type of encryption involves a shared key between the sender and the receiver, which is used for encryption and decryption. This shared key acts as a secret code between two parties and provides advantages such as simplicity, high speed, and efficiency. One of the most widely used algorithms in payment systems for asymmetric encryption is the AES algorithm. It is considered an effective encryption algorithm (Meraouche et al., 2021).

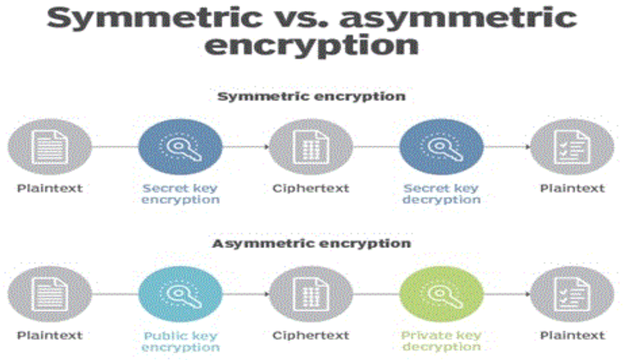


Fig 1: Symmetric and Asymmetric Encryption

Step 2: Data Collection

Accessing accurate and practical data from various sources is crucial to making informed decisions, especially given the diverse range of audiences, customers, and markets as well as the complexity of services. As the volume of data grows and the relationships between them become more complex, discovering hidden information within the data becomes increasingly challenging. This highlights the importance of data mining as a method for discovering more knowledge (Alfiah et al., 2023). In the first part of the data collection process, the number of bank payment transactions for various types of payments made from March 20, 2023, to March 20, 2024, has been made available as follows:

Table 1: Number of Payment Transactions for Each Payment Type in One Year

Row	Payment Type	Transaction count
1	Cash	25,592,273
2	Electronic (online and mobile) Banking	19,273,738
3	Card	2,157,352
4	Check	62,608

The bank has been requested to provide information on the number of payment transactions for various payment types between the period of March 20, 2023, to March 20, 2024.This data comprises about 48 million records.

During the second stage of data collection, we randomly gathered transaction amounts and dates, categorized them by payment type, and organized them into twelve sheets. Each sheet contained data for one month of the year 2023. The total number of entries in this data is approximately 800,000. Due to the confidential nature of customer payment information, we will only analyze and examine payment date details, payment transaction types, and payment transaction amounts.As a result, the dataset used in this paper includes payment transaction data consisting of payment date, payment amount, and payment type. The payment type is divided into 4 categories: cash, electronic, card, and check. After examining the data, for the sake of convenience, in the dataset used, the payment types have been assigned codes from 1 to 4 in the payment type column according to the type of payment.

Step 3: Data Preparation (Preprocessing) and Integration

The stages of data preprocessing include:

- **Data Cleaning**
  Data cleaning is a crucial step in data preprocessing that involves filling in missing data,managing noisy data, and removing outliers.

- **Handling Missing Values**
  Dealing with missing values is a common challenge in data analysis. There are two primary methods to address this challenge. The first method is to remove records that contain missing values. The second method is to impute missing values with appropriate values. There are various methods to address this challenge, such as manually filling in lost values or predicting lost values using regression or numerical methods, such as the mean, maximum, or minimum in the column feature.

- **Clustering**
  Clustering is a technique that refers to grouping or clustering data with similar values.

- **Handling Outliers**
  Clustering data places similar data into clusters. Data points that fall outside the data can be considered outliers or anomalies.There are various methods to remove or replace outliers, such as using the mean or median value.

- **Data Transformation**
  After data cleaning is completed, the quality of the data should improve significantly. Various methods are employed to enhance data quality, such as data normalization. This technique is widely used to transform data, where numerical features are rescaled to fit within a specific range.

- **Data reduction**
  Dimensionality reduction techniques aim to reduce the number of additional features considered in machine learning algorithms. This can be achieved using techniques like PCA[1], which helps to simplify the dataset by reducing its complexity.

Step 4: Applying Various Encryption and Decryption Algorithms to the Data

Principal Component Analysis[1]

When it comes to encrypting and decrypting, there are several algorithms available. One of the most significant encryption algorithms is the AES algorithm, which is a secure and robust encryption standard for sensitive data. It is used to encrypt information in various information systems, computer networks, and electronic payment transactions. The algorithm works by using blocks that are either 128 bits or 256 bits in size. The AES algorithm generally consists of four main stages as follows:

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

Byte substitution, row shifting, column mixing, and adding a round key are the four stages responsible for different aspects of data encryption (Muttaqin&Rahmadoni, 2020).The general schematic for the four stages can be outlined as follows:

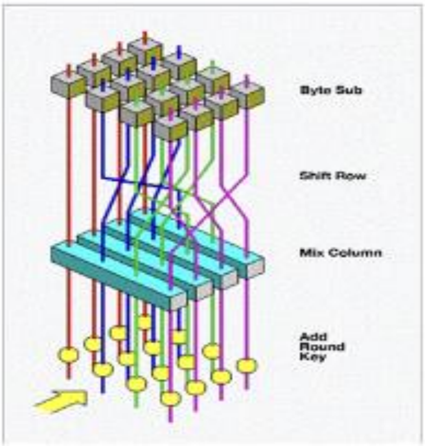


Fig 2. General Schematic of the AES Algorithm

Sub Bytes Method

In the AES encryption algorithm, a stage replaces each byte of the data block with a specific sub-byte from a substitution table called the S-box. This substitution table includes 256 8-bit inputs and 256 8-bit outputs. This stage aims to improve the security and integrity of the encrypted data by replacing each input byte with a sub-byte value from the S-box substitution table (Verma & Sharma, 2020).

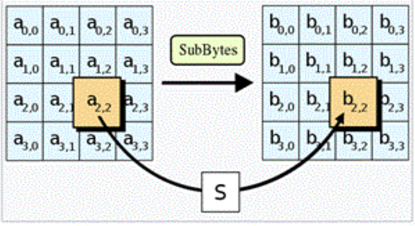

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S

Fig 3: In the SubBytes stage, each byte is replaced with its corresponding value in a fixed 8-bit lookup table

Shift Rows Method

During this stage of the AES encryption algorithm, a crucial step is accomplished by shifting the order of bytes in each row. To achieve this, each byte in a given row is moved to the left by a predetermined value. This process is referred to as "ShiftRows." By doing so, the bytes within each row interact in a complex manner, enhancing the encryption's security.

The term "ShiftRows" refers to the fact that the bytes in each row are shifted towards the next row. This operation is critical to the AES encryption algorithm's security and efficiency (Verma & Sharma, 2020).

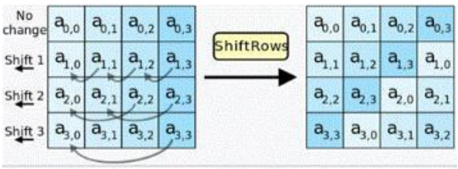

No change
Shift 1
Shift 2
Shift 3
ShiftRows
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,0}$
$a_{2,2}$ $a_{2,3}$ $a_{2,0}$ $a_{2,1}$
$a_{3,3}$ $a_{3,0}$ $a_{3,1}$ $a_{3,2}$

Fig 4: In the ShiftRows stage, the bytes in each row are cyclically shifted to the left. The number of positions .each byte is shifted varies step by step for each row

Mix Columns Method

The AES encryption algorithm involves a specific operation that transforms the byte columns in each input data block into new columns. In this process, a fixed matrix multiplies each set of four bytes in a column of the input data block to produce a new set of four bytes. This transformation results in reasonable and more complex changes in the data, enhancing the security of the encryption. The output of this operation is a specific combination of new bytes for each set of four input bytes in a column. This process leads to data diffusion and increased complexity in the patterns present in the bytes, improving the security and encryption process (Muttaqin & Rahmadoni, 2020).

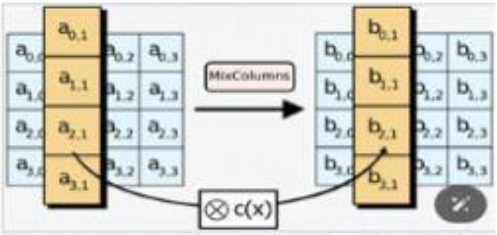

MixColumns
$a_{0,1}$ $a_{1,1}$ $a_{2,1}$ $a_{3,1}$
$b_{0,1}$ $b_{1,1}$ $b_{2,1}$ $b_{3,1}$
$\otimes c(x)$

Fig 5: In the MixColumns stage, each column is multiplied by a fixed polynomial C(x)

Key Addition Method

During the AES encryption algorithm, a random key is added to each input data block to increase the security of the encryption and modify the original data block. This is done by XORing each byte with its corresponding byte in the random key, which results in the addition of each byte to its corresponding byte in the random key. This process enhances the security and complexity of the encryption, ultimately making it more difficult to breach (Verma & Sharma, 2020).

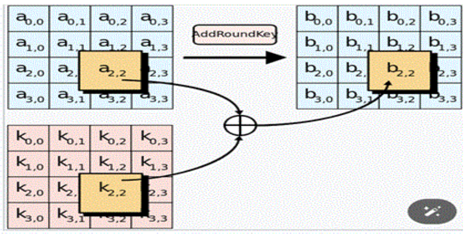


Fig 6: In the Key Addition stage, each byte is combined with a byte from the subkey using the XOR operation

The AES algorithm is a widely used method for securing sensitive information in various systems and payment transactions. It is commonly used to encrypt payment data and subsequently decrypt it.

Step 5: Modeling Using Reinforcement Learning Neural Network

Reinforcement learning modeling consists of three stages: model learning, model training, and model testing and evaluation.

Model Learning

Reinforcement learning is a process where an agent interacts with the environment to learn optimal behavior for achieving a specific objective or performing a task. The agent uses feedback from the environment to improve its performance by utilizing its experience. (Sanchez et al., 2023). The DQN model is the reinforcement learning neural network model used for decrypting payment systems. This model utilizes deep reinforcement learning methods. The DQN model works by first designing a deep neural network that estimates the action value function. The neural network takes environmental states as input and produces an estimated value of the action for each possible action in each state. The Q-learning algorithm is then used for training the DQN model. The algorithm updates the action value function based on the experiences gained from interacting with the environment to learn the best strategy for each state (Sanchez et al., 2023). When using reinforcement learning neural networks for decrypting payment systems, the current state may include information from data streams or parameters related to the payment process. The neural network can analyze this information and select appropriate decryption operations, such as selecting operations that can further facilitate access to the required information for payment or operations that can solve security issues and respond accordingly.

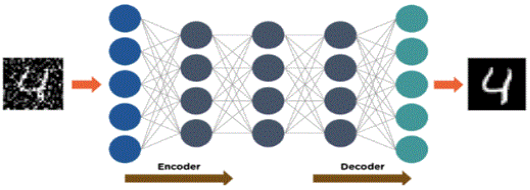


Fig 7: Encryption and Decryption Using Neural Network

The DQN algorithm employs reinforcement learning concepts and uses the Q-value function to learn which actions to take in each state to maximize the overall score. In this model, the states are fed as inputs to a neural network, which can analyze different features of these states (Alavizadeh et al., 2022).

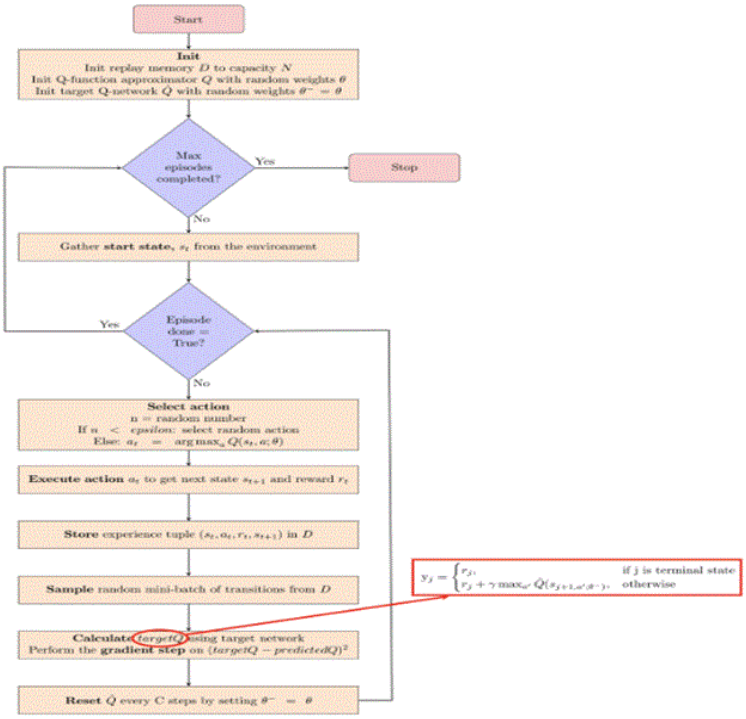

Start
Init
Init replay memory D to capacity N
Init Q-function approximator Q with random weights θ
Init target Q-network $\hat{Q}$ with random weights $\theta^- = \theta$
Max episodes completed?
Yes
Stop
No
Gather start state, s_t from the environment
Episode done = True?
Yes
No
Select action
n = random number
If $n < epsilon$: select random action
Else: $a_t = \arg\max_a Q(s_t, a; \theta)$
Execute action a_t to get next state s_{t+1} and reward r_t
Store experience tuple (s_t, a_t, r_t, s_{t+1}) in D
Sample random mini-batch of transitions from D
Calculate $targetQ$ using target network
Perform the gradient step on $(targetQ - predictedQ)^2$
Reset $\hat{Q}$ every C steps by setting $\theta^- = \theta$
$y_j = \begin{cases} r_j, & \text{if j is terminal state} \\ r_j + \gamma \max_{a'} \hat{Q}(s_{j+1}, a'; \theta^-), & \text{otherwise} \end{cases}$

Fig 8: Workflow Diagram of the DQN Algorithm

Model learning aims to create an accurate and powerful predictive model that the agent can use to choose the best actions based on the present state.

Model Training

Training a reinforcement neural network model aims to teach an agent the optimal behavior in a specific environment. This process is modeled after how living organisms learn in their everyday lives. The model learns how to receive varying responses from the environment by taking different actions to achieve the highest reward possible.

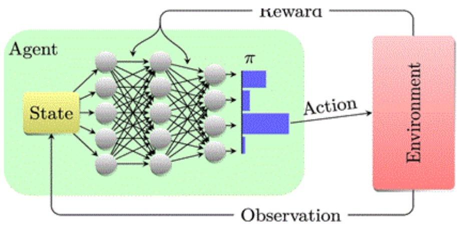


Fig 9: Deep Reinforcement Learning Model

The following are the steps involved in the model training process:

- **Environment Representation**
  First, the model must be introduced to the relevant environment by defining the states, actions, rewards, and all necessary information required for learning.

- **Action Selection**
  At each stage, the model must choose an action based on the strategies it has learned so far and the current state of the environment.

- **Action Execution**
  The environment responds to actions executed by the model.

- **Receiving Reward**
  After performing each action, the model receives a reward from the environment corresponding to the action taken.

- **Model Update**
  After receiving a reward, the model is updated using reinforcement learning algorithms such as DQN to improve future strategies and selections.

- **Iteration**
  The process will continue until a desired state is achieved or until termination conditions are met.

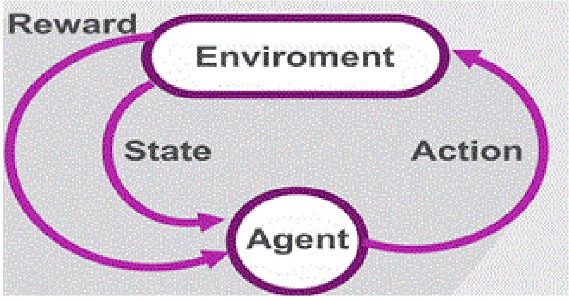


Fig 10: Relationship Between Components of the Reinforcement Learning Neural Network

Training a model in reinforcement neural networks involves training it independently of labeled data and allowing it to learn interactively from its environment. This process enables the model to learn the optimal behavior for solving the desired problem. By following these steps, we can enhance payment systems using reinforcement neural networks, which can lead to improved performance, security, and efficiency. The primary objective of model training is to enhance the model's accuracy and performance in predicting future states and rewards.

- **The model testing and evaluation**
  Indicators for comparing algorithms in the study include accuracy, precision, recall, and F1 score with their respective equations.

  - **Accuracy**
    Accuracy is a widely used metric to evaluate a model's performance. This metric measures the total number of correct predictions made by the model divided by all predictions made by the model. Accuracy considers the true positive rate, true negative rate, false positive rate, and false negative rate(Alavizadeh et al., 2022). The following equation shows how the accuracy metric is calculated:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

  - **Precision**
    Precision represents the proportion of positive samples that were correctly predicted as positive out of all the samples that were predicted as positive. The denominator in this case is the sum of the true positive rate and the false positive rate, which indicates the model's ability to predict positive samples out of the entire dataset(Alavizadeh et al., 2022). It shows how accurately the model is making positive predictions, and it is calculated using the following equation:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

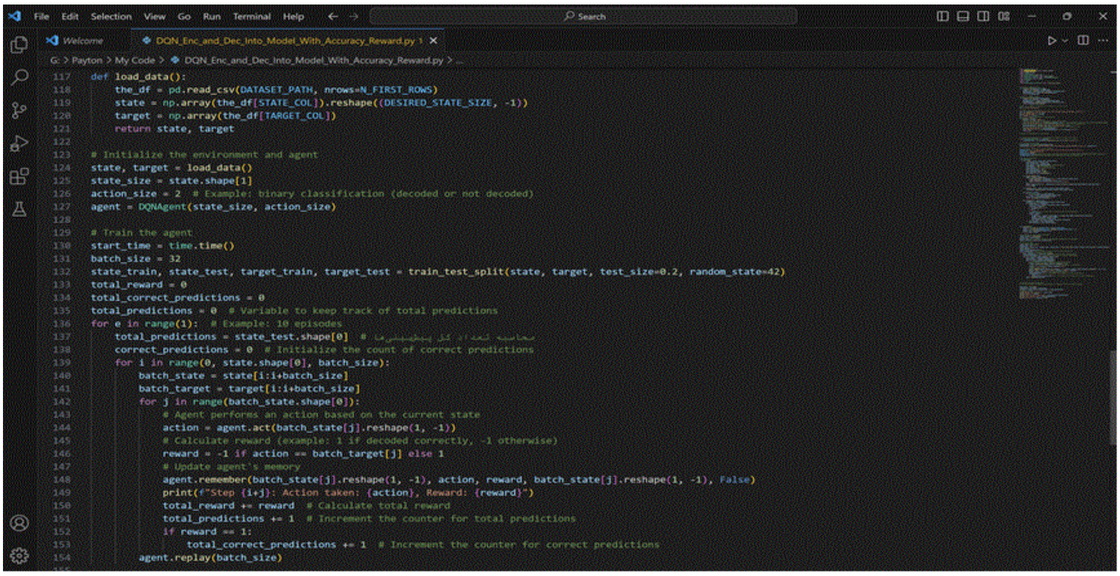
  - **Recall**
    Precision is a metric that measures the proportion of correctly predicted positive samples out of all the samples that were predicted as positive. The denominator of this metric is the sum of the true positive rate and the false positive rate, which indicates the model's ability to predict positive samples out of the entire dataset. It shows how accurately the model is making positive predictions(Alavizadeh et al., 2022). Precision can be calculated using the following equation:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

  - **F1 Score**
    The F1 score is a metric used to evaluate the performance of a classification model. It is calculated as the harmonic mean of precision and recall values. The F1 score takes into account both precision and recall values, making it a more reliable indicator of a model's performance than either precision or recall alone. A higher F1 score indicates better results. The formula for calculating the F1 score is shown below(Alavizadeh et al., 2022). Ifeither precision or recall values decrease, the final F1 score will also decrease significantly.

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

In this study, we will display the output of various algorithms using real data and their respective indicators. The code for relevant reinforcement learning using the DQN algorithm is shown below.


File
Edit
Selection
View
Go
Run
Terminal
Help
Search
Welcome
DQN_Enc_and_Dec_Into_Model_With_Accuracy_Reward.py 1
G: > Payton > My Code > DQN_Enc_and_Dec_Into_Model_With_Accuracy_Reward.py > ...

```python
def load_data():
    the_df = pd.read_csv(DATASET_PATH, nrows=N_FIRST_ROWS)
    state = np.array(the_df[STATE_COL]).reshape((DESIRED_STATE_SIZE, -1))
    target = np.array(the_df[TARGET_COL])
    return state, target

# Initialize the environment and agent
state, target = load_data()
state_size = state.shape[1]
action_size = 2  # Example: binary classification (decoded or not decoded)
agent = DQNAgent(state_size, action_size)

# Train the agent
start_time = time.time()
batch_size = 32
state_train, state_test, target_train, target_test = train_test_split(state, target, test_size=0.2, random_state=42)
total_reward = 0
total_correct_predictions = 0
total_predictions = 0  # Variable to keep track of total predictions
for e in range(1):  # Example: 10 episodes
    total_predictions = state_test.shape[0]  # محاسبه تعداد کل پیش‌بینی‌ها
    correct_predictions = 0  # Initialize the count of correct predictions
    for i in range(0, state.shape[0], batch_size):
        batch_state = state[i:i+batch_size]
        batch_target = target[i:i+batch_size]
        for j in range(batch_state.shape[0]):
            # Agent performs an action based on the current state
            action = agent.act(batch_state[j].reshape(1, -1))
            # Calculate reward (example: 1 if decoded correctly, -1 otherwise)
            reward = -1 if action == batch_target[j] else 1
            # Update agent's memory
            agent.remember(batch_state[j].reshape(1, -1), action, reward, batch_state[j].reshape(1, -1), False)
            print(f"Step {i+j}: Action taken: {action}, Reward: {reward}")
            total_reward += reward  # Calculate total reward
            total_predictions += 1  # Increment the counter for total predictions
            if reward == 1:
                total_correct_predictions += 1  # Increment the counter for correct predictions
        agent.replay(batch_size)
```

Fig 11: Sample Python code for a reinforcement neural network with the DQN model in the Visual Studio software environment

After executing the code, the chart displaying the Episode and Epsilon rows and columns is presented as shown below.

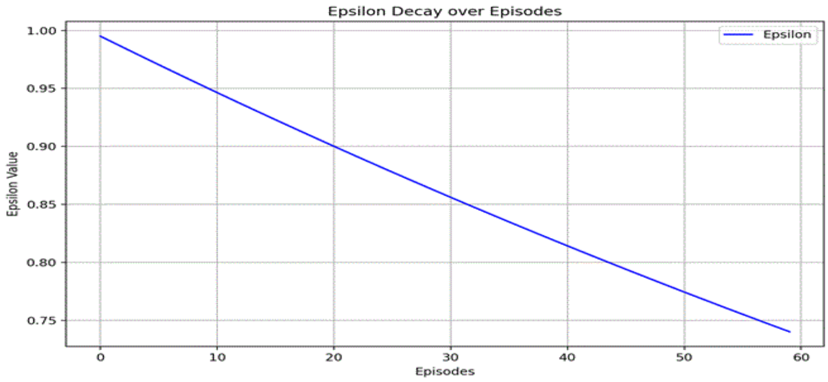


Fig 12: Exploration Chart in the DQN Algorithm

The following code generates a chart that displays the variations in the value of epsilon throughout the episodes. Epsilon is a crucial parameter in reinforcement learning algorithms, such as DQN, which dictates the balance between exploration and exploitation. The chart illustrates that the value of epsilon gradually decreases during training, resulting in a reduction of exploration and an increase in exploitation.

When the value of epsilon is low, the model tends to exploit more and is likely to perform better in the training environment. On the other hand, when the value of epsilon is high, the model explores more and is likely to achieve more potential in the training environment. The epsilon chart provides insights into how the model balances between exploration and exploitation over time. As the value of epsilon decreases, it indicates an improvement in the model's performance. In this section, we performed the process of data collection Then, we presented the modeling of encryption and decryption on payment data using reinforcement neural networks with encryption and decryption algorithms and with the help of programming in Python.

Ethical Guidelines

The use of technologies with high capabilities for accessing sensitive and financial information must be executed with care and ethical considerations. Establishing mechanisms to protect security and privacy can help increase trust and effectively use these technologies. The primary goal of using reinforcement learning neural networks for payment system decryption is to protect sensitive information, create a secure environment for financial transactions, and increase efficiency and productivity in these systems. Therefore, ethical considerations are critical and must be observed in the design and use of these technologies, especially in the financial industry. When using sensitive financial data to train a neural network, a high level of privacy protection is necessary. Customer data and their transactions must be used optimally to uphold individual privacy. Additionally, the processes and decisions made by these networks should be understandable and justifiable to users and customers. The security of financial and payment information is of utmost importance. Therefore, observing data protection security standards and procedures is necessary for the design and use of neural networks for decryption. Under no circumstances should customer financial information be used for advertising purposes. In summary, to use deep learning and reinforcement-based technologies in the field of payment system decryption, careful observation of ethical issues and customer rights is necessary. Due to the confidentiality of banking customer information, the bank's name cannot be disclosed, and the data used has been completely normalized.

CASE STUDY AND DISCUSSION

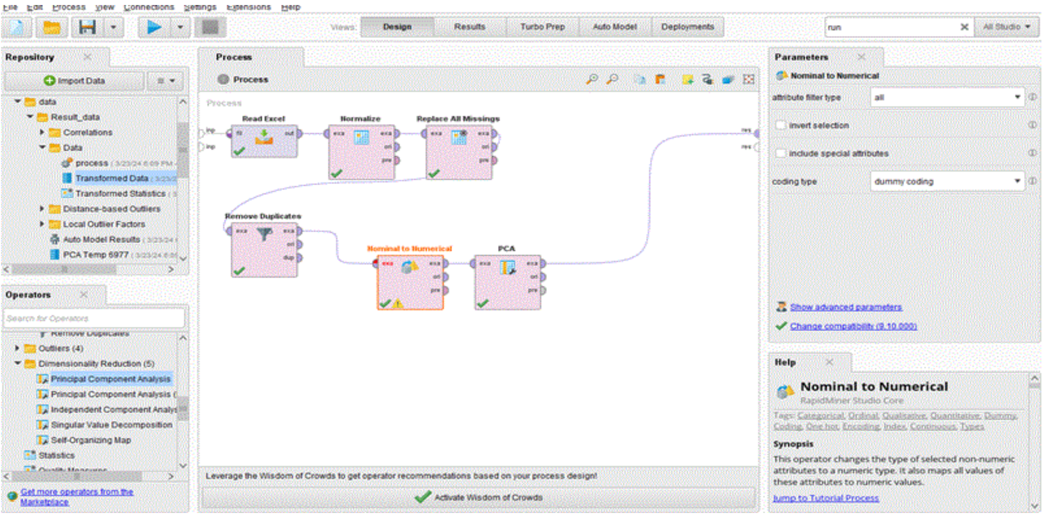
Data Collection, Preparation (Preprocessing), and Data Integration

The data used in this paper included the initial information related to transactions between March 20, 2023, and March 20, 2024. This information included transaction date, transaction type, amount, customer personal and account information, card information, check information, SATNA information, and electronic payment information. However, due to the confidentiality of the information, only the transaction date, amount, and payment type were selected for encryption and decryption operations using the bank data.Additionally, data refinement and cleansing operations were necessary to identify and remove fields with corrupted data due to the extensive volume of the extracted information. The study used RapidMiner software for data refinement, cleaning, normalization, and integrationas mentioned below.

Data Cleaning

Using RapidMiner software, the bank's payment data was cleaned through

- data normalization
- missing value replacement
- duplicate data removal
- conversion of categorical data to numerical data.



attribute filter type: all

□ invert selection

□ include special attributes

coding type: dummy coding

Nominal to Numerical

RapidMiner Studio Core

Tags: Categorical, Ordinal, Qualitative, Quantitative, Dummy, Coding, One hot, Encoding, Index, Continuous, Types

Synopsis

This operator changes the type of selected non-numeric attributes to a numeric type. It also maps all values of these attributes to numeric values.

Jump to Tutorial Process

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Fig 13: A representation of normalization, integration, dimensionality reduction, and data subset selection

During the data normalization stage, transaction amounts are transformed to a specific scale. This allows for more accurate comparison and analysis of different amounts. In some records, values may be missing. In such cases, the missing values are replaced with either maximum, minimum, or mean values. This helps to increase the accuracy of subsequent analyses.

Duplicate data can be found in some records. Removing this duplicate data can simplify and clean the data, thus preventing an increase in data redundancy in the model. Converting payment type to a numerical format is another important step in the data cleansing process.

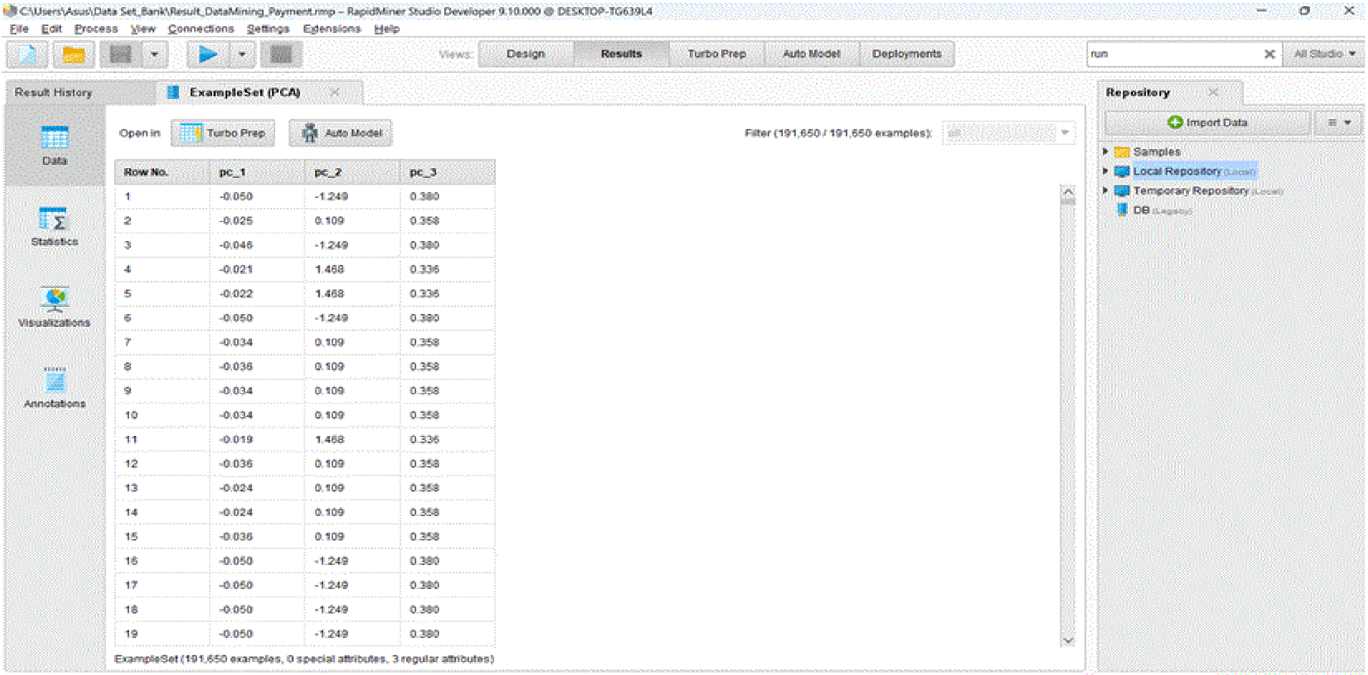
This conversion enables neural network models to be better prepared for training, leading to more accurate results. Categorical data is converted to numerical data as most models benefit from numerical inputs. These steps help to prepare payment data for training reinforcement neural network models. By following these steps, more accurate and effective analysis can be performed.

Preprocessing

To prepare and process data from a valid and accurate source, it is imperative to maintain the confidentiality of the data and ensure that no issues arise during the execution of the model. The data must be integrated, and its dimensionality must be reduced before being processed to create an effective model.

Dimensionality reduction is a technique used to decrease the number of dimensions in a dataset. The main objective of this technique is to preserve the existing information volume in the data while reducing the complexity of the model. In the case of payment data, dimensionality reduction can be particularly beneficial as this data often contains a high number of dimensions, such as a large number of transactions within a specific period.

These additional dimensions can reduce the performance of the model. By reducing these dimensions, the focus is on the important dimensions and useful information needed for modeling, which can improve the performance of reinforcement neural network models and reduce training and prediction time.


C:\Users\Asus\Data Set_Bank\Result_DataMining_Payment.rmp – RapidMiner Studio Developer 9.10.000 @ DESKTOP-TG639L4
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
run
All Studio
Result History
ExampleSet (PCA)
Data
Statistics
Visualizations
Annotations
Open in Turbo Prep Auto Model
Repository
Import Data
Samples
Local Repository (Local)
Temporary Repository (Local)
DB (Legacy)

Filter (191,650 / 191,650 examples): all

Row No.	pc_1	pc_2	pc_3
1	-0.050	-1.249	0.380
2	-0.025	0.109	0.358
3	-0.046	-1.249	0.380
4	-0.021	1.468	0.336
5	-0.022	1.468	0.336
6	-0.050	-1.249	0.380
7	-0.034	0.109	0.358
8	-0.036	0.109	0.358
9	-0.034	0.109	0.358
10	-0.034	0.109	0.358
11	-0.019	1.468	0.336
12	-0.036	0.109	0.358
13	-0.024	0.109	0.358
14	-0.024	0.109	0.358
15	-0.036	0.109	0.358
16	-0.050	-1.249	0.380
17	-0.050	-1.249	0.380
18	-0.050	-1.249	0.380
19	-0.050	-1.249	0.380

ExampleSet (191,650 examples, 0 special attributes, 3 regular attributes)

Fig14: An example of the output of data after the PCA algorithm

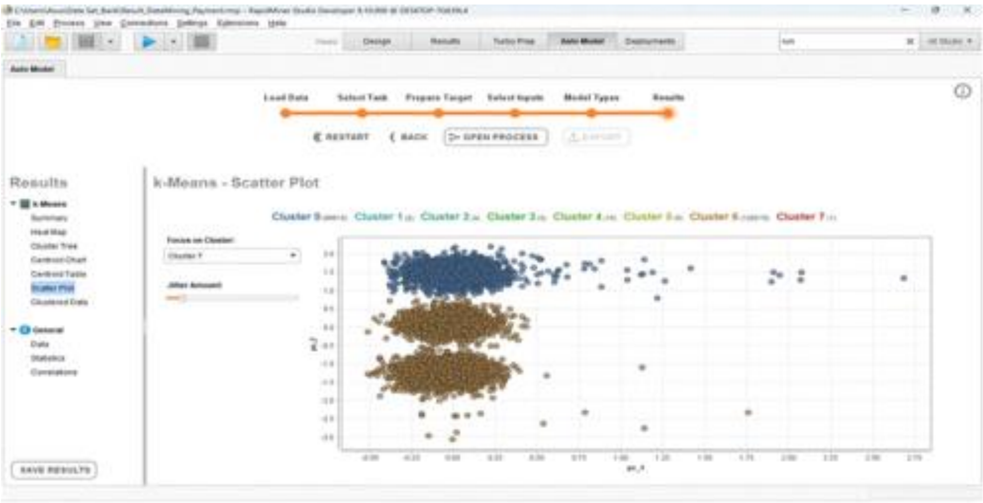
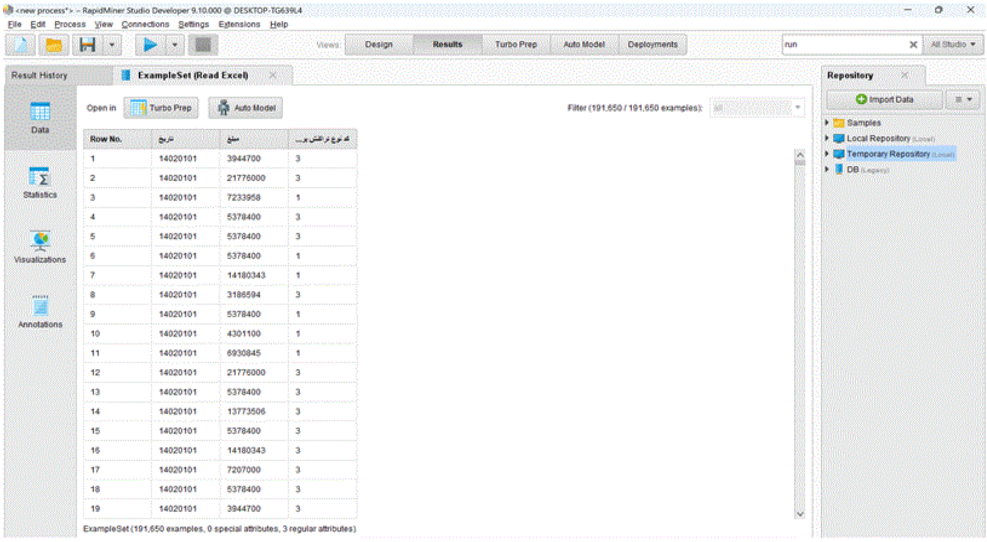


Fig15: An example of the output of data in the form of a graph and cluster scatter plot


<new process*> – RapidMiner Studio Developer 9.10.000 @ DESKTOP-TG639L4
File Edit Process View Connections Settings Extensions Help
Views: Design Results Turbo Prep Auto Model Deployments
run
All Studio
Result History
ExampleSet (Read Excel)
Data
Statistics
Visualizations
Annotations
Open in Turbo Prep Auto Model
Repository
Import Data
Samples
Local Repository (Local)
Temporary Repository (Local)
DB (Legacy)

Filter (191,650 / 191,650 examples): all

Row No.	تاریخ	مبلغ	کد نوع تراکنش پ...
1	14020101	3944700	3
2	14020101	21776000	3
3	14020101	7233958	1
4	14020101	5378400	3
5	14020101	5378400	3
6	14020101	5378400	1
7	14020101	14180343	1
8	14020101	3186594	3
9	14020101	5378400	1
10	14020101	4301100	1
11	14020101	6930845	1
12	14020101	21776000	3
13	14020101	5378400	3
14	14020101	13773506	3
15	14020101	5378400	3
16	14020101	14180343	3
17	14020101	7207000	3
18	14020101	5378400	3
19	14020101	3944700	3

ExampleSet (191,650 examples, 0 special attributes, 3 regular attributes)

Fig16: A view of payment data in the software output

Encryption and Decryption of Amount Using AES Algorithm and Creating the Match Column

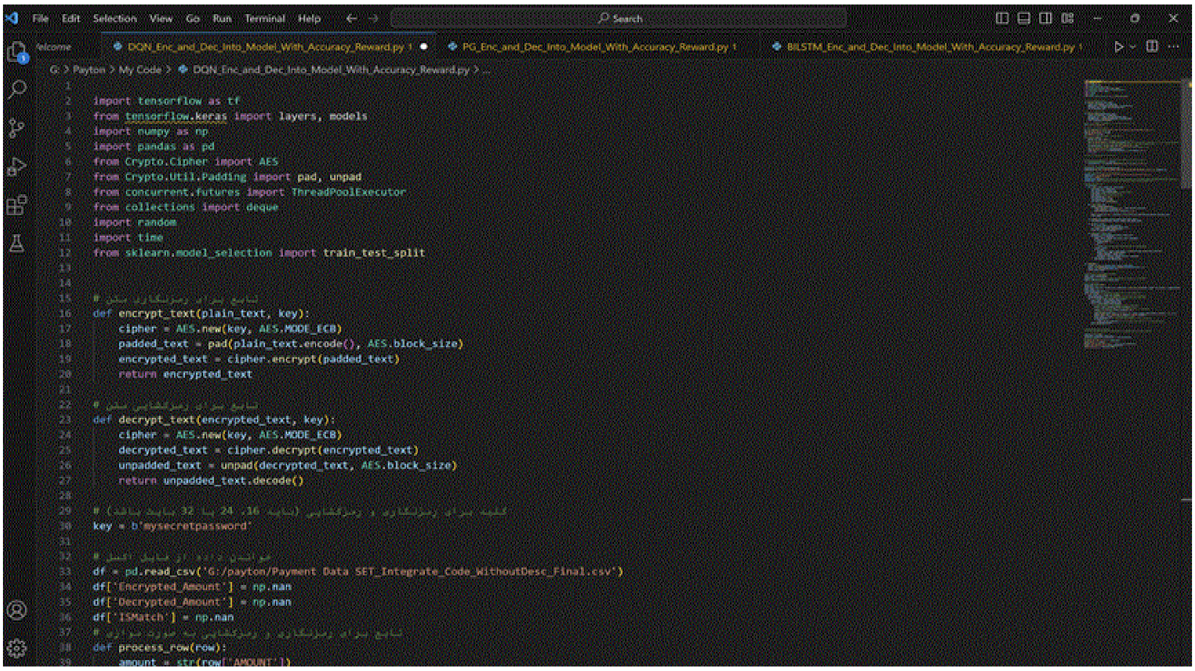
The following is an example of how to use the AES algorithm to encrypt and decrypt the amount column in a payment file. Once the amount column has been encrypted, it can then be decrypted using a specific key. If the decrypted amount matches the original amount column, the Match column will be set to zero. The Match column's equivalent will be considered a reward in the model.

Modeling Encryption and Decryption on Payment Data with Reinforcement Neural Networks

The research method involves modeling the encryption and decryption of payment data using the DQN reinforcement neural network model and the AES encryption and decryption algorithm. The AES encryption algorithm is used to encrypt sensitive information, such as payment amounts, which is initially available in unencrypted form. The data is then encrypted using a specific key and stored in an encrypted text format. Next, the DQN reinforcement neural network model is used to learn the decryption process based on the encrypted data. The DQN model receives the encrypted data as input and tries to learn the correct decryption operations by performing various actions and receiving rewards related to correct decryption. The reward is based on the level of match between the decrypted amount and the original amount. If the decrypted amount matches the original amount, the model receives a positive reward; otherwise, a negative reward is given. The DQN model uses reinforcement learning algorithms such as Q-learning or Deep Q-learning to gradually learn the decryption operations. By following these steps and training the DQN model, a secure and accurate encryption and decryption system can be implemented for payments using strong algorithms such as AES and reinforcement neural networks.

Modeling Using Neural Networks and Rewarding

A reinforcement learning model called DQN has been implemented in this paper to train an optimizing agent for a virtual environment. The virtual environment is created using a specific dataset from a file loaded using Python libraries. It uses Tensor Flow and Tensor Flow Keras libraries to construct a deep neural network model with three layers for solving the reinforcement learning problem. The model is defined with the relu and linear activation functions. The DQNAgent class is defined to manage all the model learning operations. The creation of a model with weight derivatives and the mean squared error (MSE) function is performed. It also includes functions for memory storage, performing an action in the environment, and experiencing to improve the model performance. The dataset is loaded from the file, and the inputs and outputs are sliced to match the model architecture. Then, the agent is trained based on the invoked data and consecutive actions performed. The agent is trained to solve a reinforcement learning problem by executing several episodes and consecutive steps. A part of the code and output are shown below.


File
Edit
Selection
View
Go
Run
Terminal
Help
Search
Welcome
DQN_Enc_and_Dec_Into_Model_With_Accuracy_Reward.py 1
PG_Enc_and_Dec_Into_Model_With_Accuracy_Reward.py 1
BILSTM_Enc_and_Dec_Into_Model_With_Accuracy_Reward.py 1
G: > Payton > My Code > DQN_Enc_and_Dec_Into_Model_With_Accuracy_Reward.py > ...

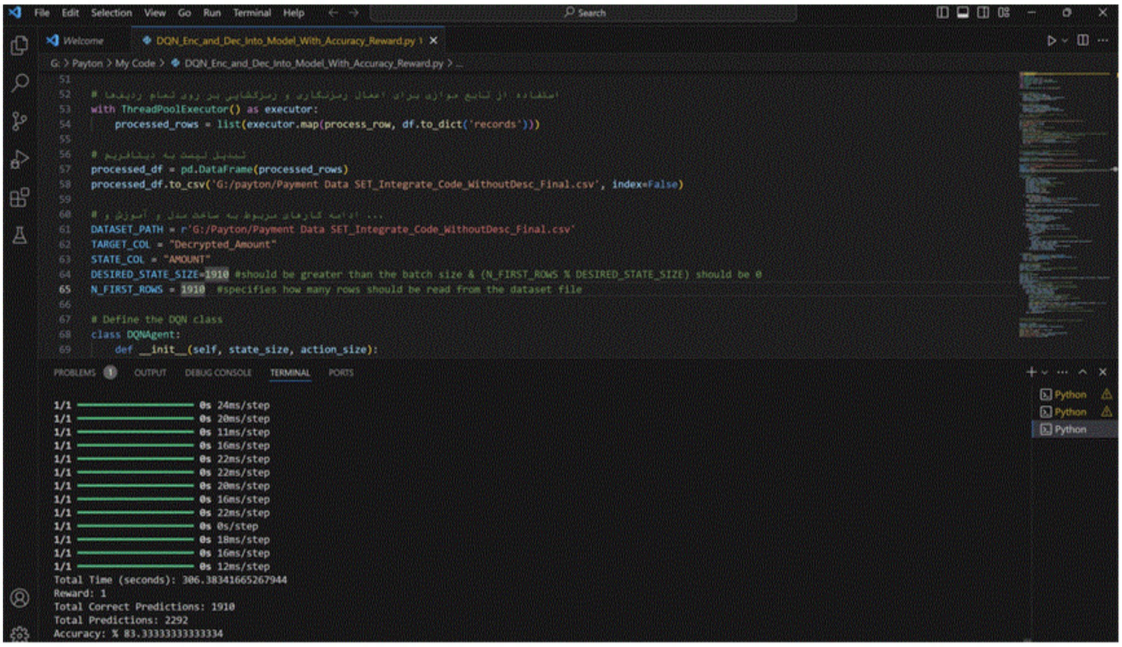
```python

import tensorflow as tf
from tensorflow.keras import layers, models
import numpy as np
import pandas as pd
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
from concurrent.futures import ThreadPoolExecutor
from collections import deque
import random
import time
from sklearn.model_selection import train_test_split


# تابع برای رمزنگاری متن
def encrypt_text(plain_text, key):
    cipher = AES.new(key, AES.MODE_ECB)
    padded_text = pad(plain_text.encode(), AES.block_size)
    encrypted_text = cipher.encrypt(padded_text)
    return encrypted_text

# تابع برای رمزگشایی متن
def decrypt_text(encrypted_text, key):
    cipher = AES.new(key, AES.MODE_ECB)
    decrypted_text = cipher.decrypt(encrypted_text)
    unpadded_text = unpad(decrypted_text, AES.block_size)
    return unpadded_text.decode()

# کلید برای رمزنگاری و رمزگشایی (باید 16، 24 یا 32 بایت باشد)
key = b'mysecretpassword'

# خواندن داده از فایل اکسل
df = pd.read_csv('G:/payton/Payment Data SET_Integrate_Code_WithoutDesc_Final.csv')
df['Encrypted_Amount'] = np.nan
df['Decrypted_Amount'] = np.nan
df['ISMatch'] = np.nan
# تابع برای رمزنگاری و رمزگشایی به صورت موازی
def process_row(row):
    amount = str(row['AMOUNT'])
```

Fig 17: An example of decryption code with modeling using reinforcement neural network and reward

The code's output consists of a loop that runs for the number of episodes and steps. During each step, the agent performs an action using the action function, calculates the reward based on the action taken, and stores information in memory. After some time, the model is reproduced and improved using the recovery function. The output includes logging that displays each step and operation performed by the agent. These logs typically include the step, selected action, received reward, and operation success (correct or incorrect recovery).

This output is useful for tracking and debugging the agent's performance or monitoring training progress. It is necessary information for evaluating and optimizing the model and the agent's performance. The presented output shows the timing information for each step and operation in the agent's training process. It includes start and end times for each operation, time taken for each step in different training episodes, time required to perform an action and calculate its reward, and time needed for model training.

Finally, the calculated accuracy and time are displayed in the output. To add accuracy calculation, the necessary information (such as the number of correct predictions and the total number of predictions) is calculated at each step and the accuracy is displayed at the end of the training episode.


File
Edit
Selection
View
Go
Run
Terminal
Help
Search
Welcome
DQN_Enc_and_Dec_Into_Model_With_Accuracy_Reward.py 1
G: > Payton > My Code > DQN_Enc_and_Dec_Into_Model_With_Accuracy_Reward.py > ...
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
Python
Python

```

# استفاده از تابع موازی برای اعمال رمزنگاری و رمزگشایی بر روی تمام ردیف‌ها
with ThreadPoolExecutor() as executor:
    processed_rows = list(executor.map(process_row, df.to_dict('records')))

# تبدیل لیست به دیتافریم
processed_df = pd.DataFrame(processed_rows)
processed_df.to_csv('G:/payton/Payment Data SET_Integrate_Code_WithoutDesc_Final.csv', index=False)

# ادامه کارهای مربوط به ساخت مدل و آموزش و ...
DATASET_PATH = r'G:/Payton/Payment Data SET_Integrate_Code_WithoutDesc_Final.csv'
TARGET_COL = "Decrypted_Amount"
STATE_COL = "AMOUNT"
DESIRED_STATE_SIZE=1910 #should be greater than the batch size & (N_FIRST_ROWS % DESIRED_STATE_SIZE) should be 0
N_FIRST_ROWS = 1910  #specifies how many rows should be read from the dataset file

# Define the DQN class
class DQNAgent:
    def __init__(self, state_size, action_size):
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 20ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 22ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 12ms/step
Total Time (seconds): 306.38341665267944
Reward: 1
Total Correct Predictions: 1910
Total Predictions: 2292
Accuracy: % 83.33333333333334
```

Fig 18: An example output, including the calculated accuracy and time for rewards

Comparison of the Output Performance Metrics of Multiple Neural Network Models

The table compares the results of the proposed DRL models based on the datasets. The results are compared with policy gradient methods, reinforcement learning, and other models such as BiLSTM and LSTM reported in various studies. The results are compared based on performance metrics such as accuracy.The table below shows the accuracy and training time (in seconds) of the proposed DQN model and other reinforcement neural network models in the paper.As shown, our model has higher accuracy compared to the other methods, while having a lower training time. However, the worst accuracy achieved by the LSTM method is around 71%. Both the PG and Reinforcement methods have high accuracy of around 80% and 79% respectively. However, these methods have lower training times compared to our model.These algorithms were evaluated with code and 1910 records, and the results are shown in the table below.

Table2 : Comparison of the DQN Model with Reinforcement Neural Network Models and Display of Accuracy and Time Used (with 1910 test data)

Model Name	Dynamic Learning	Data Set	Accuracy	Time(second)
PG	×	Bank Dataset	80 %	496.04
Reinforcement	×	Bank Dataset	79 %	510.03
LSTM	×	Bank Dataset	71 %	624.24
BILSTM	×	Bank Dataset	78 %	320.01
DQN	✓	Bank Dataset	83.33 %	306.02

CONCLUSION

Our research has beencentered on the use of reinforcement neural networks to decrypt payment systems using the AES algorithm. The study uses the DQN algorithm to train the reinforcement neural network model to enhance the security of payment systems, improve the quality and speed of decryption, and reduce the time required to perform payment operations. The research addresses the critical issue of security in online payments and utilizes advanced neural network methods and the DQN algorithm to solve it. The goal of this research is to enhance the decryption of payment information, improve the security of online payments, and enhance the speed and efficiency of the payment system. The results obtained can significantly improve the experience of online payments.

This research presents an innovative and efficient solution to enhance the security and performance of online payment systems. Since the security of sensitive information and conducting online financial transactions are of great importance, this method not only enables the preservation of user information but also improves the speed and efficiency of payments. Continued research and studies can lead to the enhancement of security and efficiency in online payment systems, playing a crucial role in technology development and the advancement of the digital economy.

LIMITATIONS

Like any other behavioral or social research, the current study has certain limitations that need to be taken into account when generalizing the findings. The limitations of this study are:

- The topic of decoding banking payment systems is relatively new and innovative, so there is limited specialized knowledge and experience in this field.
- Due to the sensitivity and confidentiality of banking data, access to information related to payment transactions, customer data, and accounts is challenging.
- In the field of using reinforcement neural networks for decrypting payment transactions, there is limited specialized knowledge and experience.
- Respecting individuals' privacy in financial information and banking payments is of utmost importance and should be given special attention throughout the project.
- There is not enough information and statistics available on decrypting payment systems by examining the existing codes in two banks in this field.
- Collaboration and coordination between specialists, bank managers, and bank customers in this project is limited and needs to be strengthened.

RECOMMENDATIONS

The following recommendations are made based on the findings of the current study:

- Examination of other machine learning techniques and algorithms for decrypting payments in financial transactions.
- Comparison of different decryption technologies for payment systems, such as classical cryptography, quantum cryptography, and multi-factor decryption.
- Examination of the application of biometric technologies such as fingerprint and facial recognition in online financial transactions, and their impact on the security and decryption of banking transactions.
- Evaluation of the potential to use a combination of different decryption methods, such as quantum cryptography and artificial intelligence, to improve the security of financial transactions.
- Research and compare various encryption methods, including public key encryption, two-factor authentication, and multi-party encryption, for securing sensitive information and payment transactions.
- Explore the potential of artificial intelligence methods such as neural networks in detecting and preventing fraud in banking transactions.
- Analyze the accuracy, speed, and reliability of reinforcement neural networks in decrypting

ACKNOWLEDGMENT

As a result of this, all the people who helped us in this research, especially the board members of banks and all senior managers of credit and human resources management, deputies, and employees working in Iranian governmental banks in Tehran, are paid the utmost gratitude.

REFERENCES

1. Alavizadeh, H., Alavizadeh, H., Jang-Jaccard, J. (2022). Deep Q-learning-based reinforcement learning approach for network intrusion detection. Computers, Vol 11, No. 3, pp. 41.
2. Alfiah, F., Usanto, S., Setiadi, A., Supriadi, A., Suhanda, Y., Nurlaela, L. (2023, November). Data Mining to Predict the Ability of Prospective Customer Credit Payments. In 2023 11th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-6). IEEE.
3. Amarnadh, S., Prudhvi, Raju., Santosh, Kumar., Sai, Charan. (2022). "Encryption and Decryption Algorithm Based on Neural Network", Journal of Engineering Sciences, Vol 13, No. 3, ISSN NO:0377-9254.
4. Bengio, Y., Lecun, Y., Hinton, G. (2021). Deep learning for AI. Communications of the ACM, Vol 64, No. 7, pp, 58-65.
5. Brown, R., Davis, C. (2020). Deep reinforcement learning for cryptanalysis. Journal of Cryptographic Engineering, Vol 28, No. 4, pp. 234-256.
6. Brown, R., Davis, C. (2019). Reinforcement Learning Approaches for Payment Decryption using Neural Networks. International Journal of Machine Learning, Vol 56, No. 2, pp. 67-89.
7. Chen, S., Wang, L. (2019). Adversarial Training for Neural Network-based Decryption Models. ACM Transactions on Privacy and Security, Vol 42, No. 1, pp. 112-129.
8. Chen, L., Wang, H. (2021). Federated Learning for Privacy-Preserving Payment Decryption with Neural Networks. IEEE Transactions on Information Forensics and Security, Vol 10, No. 4, pp. 234-256.
9. Daniel, R. M., Rajsingh, E. B., Silas, S. (2021). A forward secure encryption scheme with ciphertext authentication for e-payment systems using conic curve cryptography. Journal of King Saud University-Computer and Information Sciences, Vol 33, No. 1, pp. 86-98.
10. Fang,J.,Rao,Y.,Luo,Q.,Xu,J.(2023).Solvingone-dimensionalcuttingstock problems with deep reinforcement learning. Mathematics, Vol 11, No. 4, pp. 1028.
11. Gokcay, E., Tora, H. (2024). A novel data encryption method using an interlaced chaotic transform. Expert Systems with Applications, 237, 121494.
12. Hassan, M. A., Shukur, Z., Hasan, M. K. (2020). An efficient secure electronic payment system for e-commerce. computers, Vol 09, No. 3, pp. 66.
13. Kadjie, C. F., Hikouatcha, P., Njamen Kengdo, A. A., Nchofoung, T. N. (2023). Determinants of adoption of electronic payment by small and medium-sized enterprises (SMEs) in Cameroon. African Journal of Science, Technology, Innovation and Development, Vol 15, No. 2, pp. 185-197.
14. Kathamuthu, N. D., Chinnamuthu, A., Iruthayanathan, N., Ramachandran, M., Gandomi, A. H. (2022). Deep Q-learning-based neural network with privacy preservation method for secure data transmission in the Internet of Things (IoT) healthcare application. Electronics, Vol 11, No. 1, pp. 157.
15. Khadivi, M., Charter, T., Yaghoubi, M., Jalayer, M., Ahang, M., Shojaeinasab, A., Najjaran, H. (2023). Deep reinforcement learning for machine scheduling: Methodology, the state-of-the-art, and future directions. arXiv preprint arXiv:2310.03195.
16. Lakew, D. (2023). Factors Affecting Adoption of Online Electric Bill Payment Systems. Horn of African Journal of Business and Economics (HAJBE), Vol 06, No. 1, pp. 1-14.
17. Lin, M., Chen, T., Chen, H., Ren, B., Zhang, M. (2023). "When architecture meets AI: A deep reinforcement learning approach for a system of systems design". Advanced Engineering Informatics, 56, 101965.
18. Li, C., Zheng, P., Yin, Y., Wang, B., Wang, L. (2023). Deep reinforcement learning in smart manufacturing: A review and prospects. CIRP Journal of Manufacturing Science and Technology, 40, pp. 75-101.
19. Malallah, F. L., Abduljabbar, A. I., Shareef, B. T., Al-Janaby, A. O. (2023, February). QR Code Encryption for improving Bank information and Confidentiality. In 2023 27th International Conference on Information Technology (IT) (pp. 1-4). IEEE.
20. Malyadri, M., Raghavi, C., Reddy, P. S., Kumar, G. (2022). "Encryption and Decryption Algorithm Based on Neural Network", International Journal of Advances in Engineering and Management, Vol 04,No. 6, pp. 795-798.

21. Matsunami, N., Okuhara, S., Ito, T. (2021). Reward Design for Multi-Agent Reinforcement Learning with a Penalty Based on the Payment Mechanism. Transactions of the Japanese Society for Artificial Intelligence, Vol 36, No. 5, AG21-H_1.
22. Meng, T. L., Khushi, M. (2019). Reinforcement learning in financial markets. Data, Vol 4, No. 3, pp. 110
23. Meraouche, I., Dutta, S., Tan, H., Sakurai, K. (2021). Neural networks-based cryptography: A survey. IEEE Access, 9, 124727-124740.
24. Mittal, S., Jindal, P., Ramkumar, R. (2021). "Data Privacy and System Security for Banking on Clouds using Homomorphic Encryption," 2021 2nd International Conference for Emerging Technology (INCET), Belagavi, India, 2021, pp. 1-6, doi: 10.1109/INCET51464.2021.9456345.
25. Muttaqin, K., Rahmadoni, J. (2020). Analysis and design of file security system AES (advanced encryption standard) cryptography based. Journal of Applied Engineering and Technological Science (JAETS), Vol 1, No. 2, pp. 113-123.
26. Ogheneruemu, A. S., Taiye, A. O. (2023). Electronic Payment System Using Visual Cryptographic Scheme. International Journal of Advances in Engineering and Management (IJAEM), Vol 5, No. 2, pp. 36-39. www.ijaem.net ISSN: 2395-5252.
27. Pal, S. K., Datta, B., Karmakar, A. (2022). An Artificial Neural Network Technique of Modern Cryptography. Journal of Scientific Research, 14(2).
28. Pathak, B., Pondkule, D., Shaha, R., Surve, A. (2020). Visual cryptography and image processing-based approach for bank security applications. In Second International Conference on Computer Networks and Communication Technologies: ICCNCT 2019 (pp. 292-298). Springer International Publishing.
29. Qin, Z., Ye, H., Li, G. Y., Juang, B. H. F. (2019). Deep learning in physical layer communications. IEEE Wireless Communications, Vol 26, No. 2, pp. 93-99.
30. Razumov, P., Cherckesova, L., Revyakina, E., Morozov, S., Medvedev, D., Lobodenko, A. (2023). Ensuring the security of web applications operating based on the SSL/TLS protocol. In E3S Web of Conferences (Vol. 402, p. 03028). EDP Sciences.
31. Sanchez, F. R., Wang, Q., Bulens, D. C., McGuinness, K., Redmond, S., O'Connor, N. (2023). Learning and reusing primitive behaviors to improve Hindsight Experience Replay sample efficiency. arXiv preprint arXiv:2310.01827.
32. Sutton, R. S., Barto, A. G. (2018). Reinforcement learning: An introduction. MIT Press.
33. [Tsmots, I., Teslyuk, V., Łukaszewicz, A., Lukashchuk, Y., Kazymyra, I., Holovatyy, A., Opotyak, Y. (2023). Neural network technology for cryptographic protection of data transmission at uav. https://doi.org/10.20944/preprints202304.0252.v1
34. Verma, R., Sharma, A. K. (2020). Cryptography: Avalanche effect of AES and RSA. International Journal of Scientific and Research Publications, Vol 10, No. 4, pp. 119-122.
35. Verma, R., Nagar, T., Sharma, M. K., Kumar, M. (2024, January). Cryptography encryption algorithm for augmented security. In AIP Conference Proceedings (Vol. 2978, No. 1). AIP Publishing.
36. Vishwakarma, P. P., Tripathy, A. K., Vemuru, S. (2020). Designing a cryptosystem for data at rest encryption in mobile payments. International Journal of Applied Science and Engineering, Vol 17, No. 4, pp. 373-382.
37. Vimal, S., Kayathwal, K., Wadhwa, H., Dhama, G. (2021). Application of deep reinforcement learning to payment fraud. arXiv preprint arXiv:2112.04236.
38. Yeow, S. Q., Ng, K. W. (2023). Neural Network Based Data Encryption: A Comparison Study among DES, AES, and HE Techniques. JOIV: International Journal on Informatics Visualization, 7(3-2).
39. Zhou, R., Tian, Y., Wu, Y., Du, S. (2022). Understanding Curriculum Learning in Policy Optimization for Solving Combinatorial Optimization Problems. arXiv arXiv:2202.05423.