Secure Trade Mediation System

Arul Blessy S R[1], Abinaya K[2], Kaviya P[3], Kalaiselvi S[4]
[1, 2, 3, 4] Dept of Computer Science and Engineering
[1, 2, 3, 4] Vivekanandha College of Technology for Women, Tiruchengode, Namakkal, Tamil Nadu, India

Abstract- *In this modern era, we have faced a lot of e-commerce scams, so we have introduced a system to reduce the this scam. Wehave built a dual escrow mediation where both the buyer and seller have to initiate 10% of the deposit inside the escrow wallet for cash on delivery For the prepaid, the buyer has to pay the whole amount price, and that price will be locked in the escrow. The 10% of the deposit, which is called the security money, will be deposited by the seller into the escrow wallet. In this system, the agent delivery acts as a judge, a genuine return request and the unreasonable request will be handled by the agent delivery. If the return request was unreasonable by the buyer, then a penalty will be given to the buyer.If the agent judges a return request as unreasonable, then 10% of productprice is redistributed to the three parties The 3% of the amount goes to the agent for delivery as a service fees. 7% of the product price is transferred to the seller as a compensation and then the remaining 90% of the product price is returned to the buyer. Because of the unreasonable request, the buyer's reputation score will be reduced to 0.1 leading to a 9.9 reputation score. Whenever the buyer or seller tries to scam, their reputation score will be reduced to 0.1 . The reputation score will help both the buyer and seller to identify that the particular party have scammed. The system was built using Python3.11+, Streamlit, FastAPI, Uvicorn, and the Requests library. The system results confirm the escrow lock, dual escrow mediation, penalty redistribution and repetition scoring.*

Keywords: Escrow, Return Fraud, Penalty Redistribution, ReputationSystem, E-commerce Security, Dual-Escrow

I. INTRODUCTION

Online shopping in India has become a massive industry, but it has a major problem with scams that happen during the transaction. Two specific patterns are causing the most damage: Cash-on-Delivery (COD) fraud and return fraud. In COD fraud, a buyer orders an item but refuses to pay or accept it when the delivery agent shows up at their house. This leaves the seller with unsold items and the agent with a wasted trip. In return fraud, the buyer gets a perfectly good product but then files a fake complaint saying it’s damaged just to get their money back. These issues make it very hard for honest sellers to survive.

Most e-commerce platforms today only try to fix these problems after they have already happened. Usually, a buyer makes a complaint, and a customer service team reviews it days later. This doesn't really work because the seller and the delivery agent have no way to prove what actually happened at the door. Sellers and agents are usually at a disadvantage because the current system almost always sides with the buyer, even if the buyer is lying. This lack of evidence makes the whole process unfair for the people shipping the goods.

Our system takes a different approach by stopping the fraud before it even starts. Instead of waiting for a dispute, we require both the buyer and the seller to lock a 10% deposit into a shared "escrow" wallet before the order is even processed. This means both sides have "skin in the game." During the delivery, the agent acts as a neutral witness who can see the product. If there is a return request, the agent decides if it is genuine or not, and the money is distributed based on that decision. This paper explains how we designed and built this Secure Trade Mediation System to make online trading safer.

II. OBJECTIVES

The first goal is to make sure that Cash-on-Delivery scams actually cost the buyer something. Right now, a buyer can cancel a COD order for no reason and lose nothing. By forcing them to lock a 10% deposit before the order is confirmed, they now have something to lose. If they refuse a delivery without a good reason, they won't get that deposit back. This makes it much harder for dishonest buyers to play around with orders.

The second goal is to give sellers a fair way to get their money back when they are scammed. If a seller sends a good product but the buyer files a fake return, the seller usually loses money on shipping and handling. In our system, the seller gets 7% of the product price as compensation if the return is found to be fake. This money helps cover their costs and makes the platform fairer for people selling goods.

The third goal is to give the delivery agent a proper role in solving disputes. Since the agent is the one standing at the door, they are the best person to see if a return request is

real or fake. Our system pays the agent a 3% fee when they correctly identify a scam. This encourages them to be honest and take their job as a witness seriously, rather than just ignoring the problem.

Finally, we want to build a reputation score for every buyer. The system will track how many times a buyer's return request was marked as "unreasonable." If someone keeps trying to scam, their score will drop. This way, sellers can check the score before they agree to a deal. It works like a warning system to help identify who is a "safe" buyer and who is not.

III. PROBLEM STATEMENT

Cash-on-Delivery is still the favorite way to pay for most people in India, especially in smaller cities. However, many people take advantage of this. Since a buyer doesn't have to pay anything upfront for a COD order, they don't lose anything if they change their mind. They might find a better price elsewhere or just decide they don't want it anymore, so they refuse the package at the door. When this happens, the seller is the one who suffers because they have to pay for the shipping costs and the product is stuck in transit instead of being sold to someone else.

Return fraud works in a similar way. Once the buyer has the product, it is very hard for the seller to protect themselves. Most shopping apps usually take the buyer's side, and it's almost impossible for a seller to prove that they sent a perfect item. Some buyers have even learned how to use these return rules to "rent" products for free by using them and then sending them back. Sellers have no way to fight these fake claims.

Delivery agents are in a tough spot too. They do all the hard physical work of carrying packages, but they don't get anything out of solving disputes. Even though they are the ones who see what actually happens during the delivery, nobody asks for their proof or opinion. If a trip is wasted because of a fake order, they just lose time and effort without any protection or payment.

Our system fixes all three of these problems at once. We make everyone financially responsible before the delivery even starts. We give the delivery agents the power to make a final decision on disputes, and we use a simple math formula to share the penalty money fairly if someone tries to scam.

IV. METHODOLOGY

We built this system as a web app using Python 3.11. We used Streamlit to create the three main pages: the Buyer Marketplace, the Seller Portal, and the Agent Panel. For the backend, we used FastAPI to handle the logic and Uvicorn to run the server. To make the frontend talk to the backend, we used the Requests library to send data back and forth. The whole project was made in VS Code. All the wallet balances and order info are saved in the backend database, so your money stays there even if you close the browser. For this prototype, we used simple login details to keep things easy to test.

A.System Architecture

The platform is split into five main parts that talk to the three user pages. The Order module handles new orders and tracks if they are delivered or cancelled. The Escrow module locks the deposit money and releases it later. The Penalty module calculates the 3% and 7% splits if a scam is found. The Wallet module keeps track of how much "real" money and "escrow" money each user has. Finally, the Reputation module updates the user scores after every deal

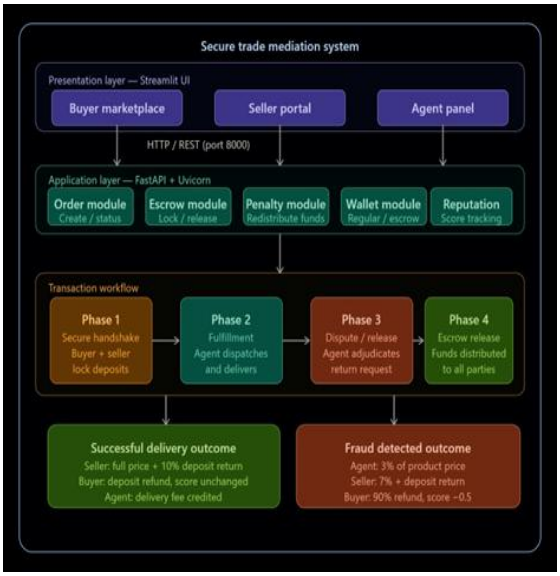

Fig. 1. System Architecture of the Secure Trade Mediation Platform

B. Transaction Workflow

Every deal has four steps. First is the Secure Handshake, where everyone puts in their deposit. For prepaid, the buyer pays the full price plus the seller puts in 10%. For COD, both sides put in a 10% deposit. The deal only starts once both sides have locked their money. Second, the agent picks up the item and delivers it. If it's COD, the buyer pays

the agent in cash at the door. Third, if everything is fine, the money is released to the seller and deposits are returned. Fourth, if the buyer wants to return it, the agent has to step in and judge the case.

C. Fraud Detection and Penalty Redistribution

When a return request is submitted, the delivery agent receives a notification on their dashboard showing the order details and the buyer's claim. The agent is presented with two clearly labelled options: REASONABLE and NOT REASONABLE. If the agentselects REASONABLE, the buyer receives a full refund of the product price from escrow, and both security deposits are returned to their owners. No reputation penalty is applied, and the transaction closes without any financial loss to any party.

If the agent selects NOT REASONABLE, the system immediately applies the penalty redistribution formula. The buyer's reputation score is reduced by 0.1. From the product price held in escrow, three percent is credited to the agent as a service fee for their time and judgment. Seven percent is credited to the seller as partial compensation for logistics and opportunity costs. The remaining ninety percent is returned to the buyer. Both security deposits are then released back to their respective owners. This formula ensures that the full product price is accounted for across all three parties without any funds being withheld by the platform.

V. MATHEMATICAL MODEL

The financial behaviour of the system is governed by a small set of transparent equations. Let P represent the product price, Ds represent the seller's security deposit equal to 10% of P, and Db represent the buyer's security deposit also equal to 10% of P, which applies only in COD transactions.

The total escrow amount E locked before a prepaid order proceeds is defined as E = P + Ds, meaning the buyer contributes the full product price and the seller contributes their deposit. For a COD order, E = Db + Ds, because the buyer pays P in cash at the point of delivery and only the two deposits are held in escrow beforehand.

When the NOT REASONABLE verdict is delivered, the redistribution is computed as follows. The agent compensation Ca = P × 0.03. The seller compensation Cs = P × 0.07. The buyer refund Rb = P × 0.90. The conservation check Ca + Cs + Rb = P confirms that no money disappears from the system.The full product price is redistributed across the three parties in fixed proportions.

TABLE I. ESCROW OUTCOME SUMMARY

Scenario	Buyer Locks	Seller Locks	Agent Gets	Seller Gets	Buyer Gets Back
Prepaid - Success	Full Price	10% Deposit	Nil	Full Price + 10% refund	Nil
COD - Success	10% Deposit	10% Deposit	Nil	Full Price + 10% refund	10% refund
Prepaid - Fraud	Full Price	10% Deposit	3% of Price	7% of Price + 10% refund	90% of Price
COD - Fraud	10% Deposit	10% Deposit	3% of Price	7% of Price + 10% refund	0% (deposit used)

VI. RESULTS AND DISCUSSION

The system was tested using three simulated user accounts representing the three roles: a buyer account named Blessy_buyer, a seller account named Blessy_seller, and a delivery agent account. All three accounts began with a simulated wallet balance of Rs. 30,000. A green leather hand bag priced at Rs. 1,000 was used as the test product across all scenarios. The following subsections walk through each stage of the transaction lifecycle as it was observed during testing.

A. Product Listing by Seller

The seller logs into the Seller Portal and opens the Include Item section. Here, the seller types the product name, sets the price, and pastes an image URL or base64 image string. After clicking Include Item, the platform confirms that the product is now visible to buyers with a green notification. The seller's reputation score appears as 10.0 at the top right of the portal, reflecting a clean transaction history.

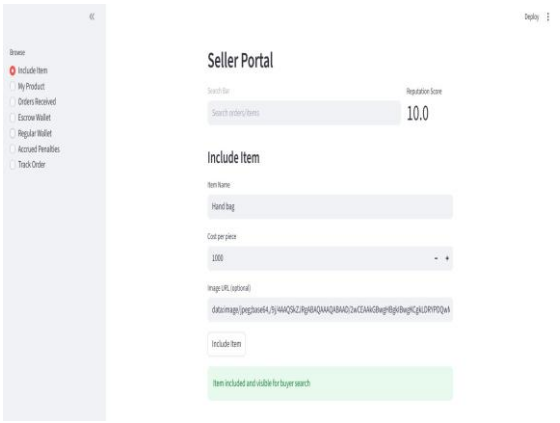

Fig. 2. Seller Portal showing the product listing form

B. Buyer Marketplace and Order Placement

When the buyer opens the Buyer Marketplace, the listed products appear with their images, prices, and seller names. The buyer chooses a payment mode from the dropdown (either PREPAID or COD) and clicks the Buy button. For a prepaid order, the full product price of Rs. 1,000 is immediately transferred from the buyer's regular wallet into escrow. A confirmation message appears: Deal sent to seller (PREPAID), and the buyer's reputation score shows 10.0 on their fresh account.

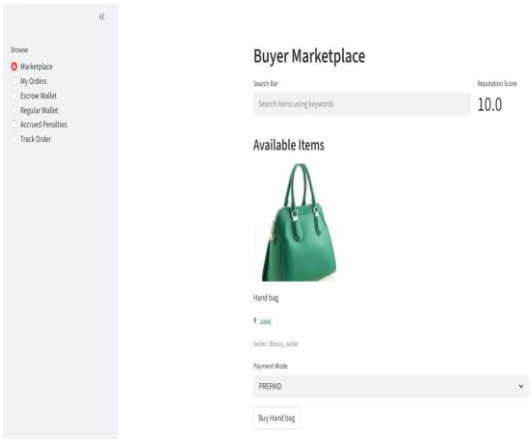

Fig. 3. Buyer Marketplace showing the purchase interface and reputation score

C. Escrow Lock Confirmation

After the deal is submitted, the buyer's Escrow Inflow/Outflow log records the transaction. The regular wallet balance drops from Rs. 30,000 to Rs. 28,900, and the escrow log entry reads: Deal Sent for Rs. 1,000 as an inflow to escrow for Hand bag in PREPAID mode. This confirms that the escrow lock is applied immediately and the buyer cannot recover these funds until the order is resolved through the system.

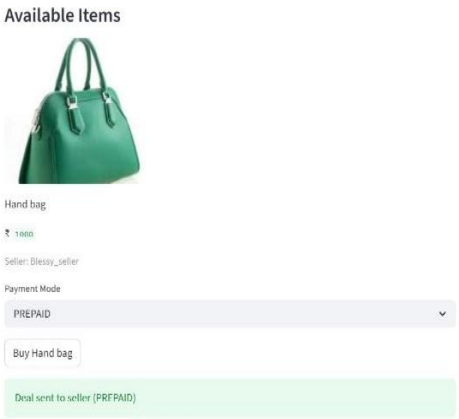

Fig. 4. Buyer wallet reflecting the Rs. 1,000 escrow lock

D. Seller Agreement and Dispatch

On the Seller Portal, the incoming order appears under Orders Received. The order card displays the buyer's name, payment mode, current status as NOT YET DISPATCHED, and a Seller Agree: Yes flag, which confirms that the seller's ten percent security deposit has been successfully locked into escrow. The seller may cancel the order if the item is out of stock, but doing so carries a reputation consequence that discourages unnecessary cancellations.

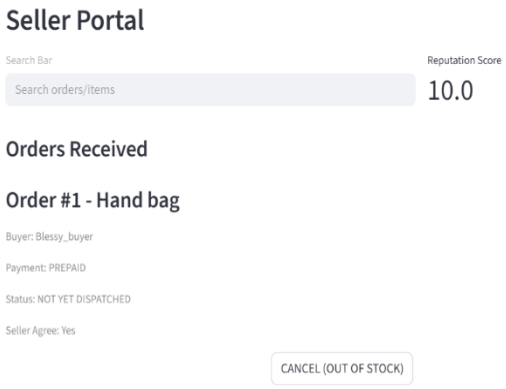

Fig. 5. Seller Portal showing order received with both deposits locked

E. Agent Delivery Confirmation

The delivery agent logs into the Agent Panel, where the pending order for the Hand bag is listed with the buyer and seller names and a status of NOT YET DISPATCHED. The agent uses the Mark IN TRANSIT button when they pick up the item, and the Mark DELIVERED button when they complete the handover. After clicking Mark DELIVERED, the escrow releases automatically. A green confirmation message reads: Order delivered successfully! Seller received the full product amount plus 10% security deposit.

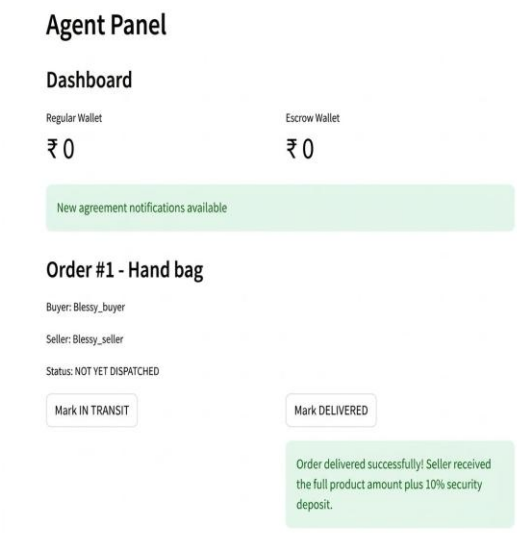

Fig. 6. Agent Panel after clicking Mark DELIVERED

F. Seller Balance After Successful Delivery

Following a successful delivery, the seller's regular wallet updates to Rs. 31,070. This figure can be broken down as follows: the starting balance of Rs. 30,000, minus the Rs. 100 security deposit locked at the time of the handshake, plus Rs. 1,000 released from escrow as the product price, plus the Rs. 100 security deposit returned, and an additional Rs. 70 from a seven percent seller gain credited during a prior fraud scenario in the same session. This confirms that the escrow release and deposit return logic both execute correctly.

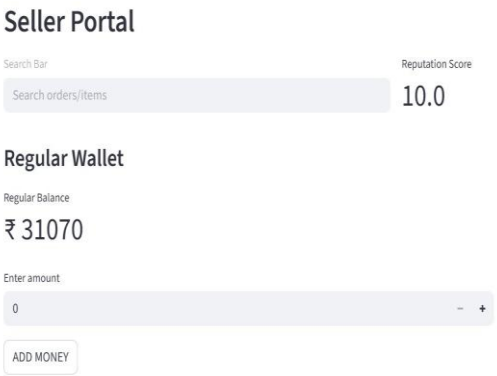

Fig. 7. Seller wallet balance after successful delivery

G. Buyer Return Request

After the item is delivered, the buyer can view their completed orders under My Orders. The page shows the Hand bag under Delivered Packages with a PREPAID payment label and an image of the product. The buyer selects the order and submits a return request, which changes the order status to RETURN REQUESTED. The system prevents duplicate submissions by displaying the message: Return already in process/completed for this order.

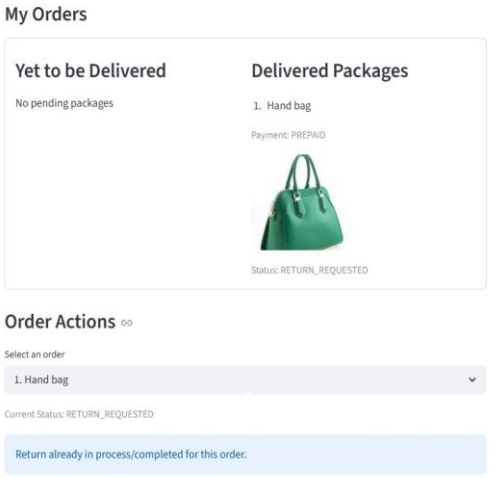

Fig. 8. Buyer order page showing RETURN REQUESTED status

H. Agent Adjudication

When the return request reaches the agent, their dashboard updates to show the order with a RETURN REQUESTED status. Two buttons appear: REASONABLE and NOT REASONABLE. In the fraud test scenario, the agent clicks NOT REASONABLE. The system immediately applies the penalty formula and displays a confirmation message: Return rejected. Buyer penalized 10%, seller received 7%, and you received 3% fee. This confirms that the adjudication logic executes in real time and that all three affected parties receive an outcome notification without any manual intervention.

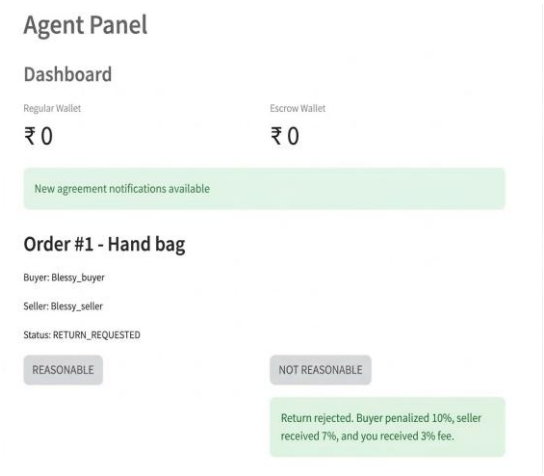

Fig. 9. Agent Panel showing NOT REASONABLE verdict and penalty confirmation

I. Agent Escrow Credit

After the NOT REASONABLE verdict is recorded, the agent's escrow wallet shows a balance of Rs. 30, representing three percent of the Rs. 1,000 product price. This amount is visible on the agent's dashboard under Escrow Wallet. The agent's regular wallet remains at zero because the prototype does not pre-load agent wallets with a starting balance. The credit flowing correctly to the agent's escrow account validates that the three-percent service fee calculation and transfer are functioning as designed.

Fig. 10. Agent escrow wallet showing 3% service fee after fraud verdict

J. Buyer Reputation Penalty

Following the fraud verdict, the buyer's reputation score visible at the top right of the Buyer Marketplace drops from 10.0 to 9.9, reflecting the 0.1 deduction applied for a detected fraudulent return. The buyer's regular wallet also reflects the ninety percent refund of the product price, settling at Rs. 28,900 after all escrow transactions conclude. This validates the complete penalty cycle: fund redistribution, deposit return, and reputation deduction all execute correctly within a single transaction lifecycle.

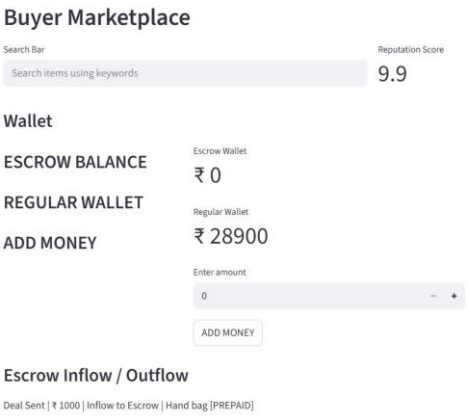

Fig. 11. Buyer Marketplace showing reputation score of 9.9 after fraud penalty

VII. CONCLUSION

This paper described the design, implementation, and experimental validation of a Secure Trade Mediation System that uses a dual-escrow mechanism and a structured penalty-redistribution model to reduce Cash-on-Delivery fraud and return fraud in online commerce. The system was built using Python 3.11+, Streamlit, FastAPI, Uvicorn, and the Requests library and tested through a prototype environment with simulated wallet balances. All key scenarios were validated: successful delivery, fraudulent return detection, penalty redistribution, deposit refund, and reputation scoring.

The core contribution of this work is the idea of creating financial accountability before delivery rather than reacting to fraud after it occurs. By requiring both parties to lock deposits upfront, the system ensures that dishonest behavior carries a real cost. The delivery agent's formal role as an adjudicator, supported by a financial incentive in the form of a three-percent service fee, makes dispute resolution faster, more transparent, and harder to game than conventional platform review processes.

Looking ahead, this system could be extended in several meaningful directions. Integrating it with a real payment gateway would allow the prototype to handle actual transactions. A machine learning layer trained on historical agent verdicts could assist agents in borderline cases by flagging patterns of buyer behavior. The reputation algorithm could be refined to weight recent transactions more heavily than older ones, so that a buyer who reforms their behavior is not permanently penalized for past mistakes. The model could also be adapted for multi-seller or multi-agent logistics scenarios where the escrow and penalty logic would need to account for more complex fund flows.

REFERENCES

[1] H. Alzoubi, M. Alshurideh, B. A. Kurdi, K. Alhyasat, and T. Ghazal, "The effect of e-payment and online shopping on sales growth: Evidence from the banking industry," International Journal of Data and Network Science, vol. 6, no. 4, pp. 1369-1380, 2022.

[2] D. N. Nandal, M. N. Nandal, D. K. Mankotia, and M. N. Jora, "Investigating digital transactions in the interest of a sustainable economy," International Journal of Modern Agriculture, vol. 10, no. 1, pp. 1150-1162, 2021.

[3] S. Idayanti, T. Haryadi, and E. Indriasari, "Online Dispute Resolution (ODR) as an alternative for electronic transaction dispute settlement," Wacana Hukum, vol. 27, no. 2, pp. 18-25, Aug. 2021.

[4] I. A. Omar, H. R. Hasan, R. Jayaraman, K. Salah, and M. Omar, "Implementing decentralized auctions using blockchain smart contracts," Technological Forecasting and Social Change, vol. 168, p. 120786, Jul. 2021.
[5] M. M. Khan, N. T. RoJa, F. A. Almalki, and M. Aljohani, "Revolutionizing e-commerce using blockchain technology and implementing smart contract," Security and Communication Networks, vol. 2022, p. 2213336, 2022.
[6] D. Negi, A. Sah, S. Rawat, T. Choudhury, and A. Khanna, "Blockchain platforms and smart contracts," in Blockchain Applications in IoT Ecosystem, Cham: Springer, 2021, pp. 65-76.
[7] M. Muneeb, Z. Raza, I. U. Haq, and O. Shafiq, "SmartCon: A blockchain-based framework for smart contracts and transaction management," IEEE Access, vol. 10, pp. 23687-23699, Dec. 2021.
[8] M.Krichen, "Strengthening the security of smart contracts through the power of artificial intelligence," Computers, vol. 12, no. 5, p. 107, May 2023.
[9] O. Andersen, "How will artificial intelligence (AI) and smart contracts (SC) change transaction costs and governance mode?," in Proc. Global Business Conference, 2024, p. 10.
[10] B. Bera, M. Wazid, A. K. Das, and J. J. Rodrigues, "Securing the internet of drone networks using AI-envisioned smart-contract-based blockchain," IEEE Internet of Things Magazine, vol. 4, no. 4, pp. 68-73, Dec. 2021.
[11] Q. Cheng, Y. Gong, Y. Qin, X. Ao, and Z. Li, "Secure digital asset transactions: Integrating distributed ledger technology with safe AI mechanisms," Academic Journal of Science and Technology, vol. 9, no. 3, pp. 156-161, Mar. 2024.
[12] Y. Zhang, W. Yu, Z. Li, S. Raza, and H. Cao,"Detecting Ethereum Ponzi schemes based on improved LightGBM algorithm," IEEE Transactions on Computational Social Systems, vol. 9, no. 2, pp. 624-637, Jul. 2021.
[13] L. Wang, H. Cheng, Z. Zheng, A. Yang, and M. Xu, "Temporal transaction information-aware Ponzi scheme detection for Ethereum smart contracts," Engineering Applications of Artificial Intelligence, vol. 126, p. 107022, Nov. 2023.
[14] A. Kumar, K. Abhishek, P. Nerurkar, M. R. Ghalib, A. Shankar, and X. Cheng, "Secure smart contracts for cloud-based manufacturing using Ethereum blockchain," Transactions on Emerging Telecommunications Technologies, vol. 33, no. 4, p. e4129, Apr. 2022.
[15] R. F. Ibrahim, A. M. Elian, and M. Ababneh, "Illicit account detection in the Ethereum blockchain using machine learning," in Proc. 2021 International Conference on Information Technology(ICIT),Jul2021,pp. 488-489