mathematics

MDPI

Article

A Novel Auction Blockchain System with Price Recommendation and Trusted Execution Environment

Dong-Her Shih [1], Ting-Wei Wu [1], Ming-Hung Shih [2,*], Wei-Cheng Tsai [1] and David C. Yen [3]

[1] Department of Information Management, National Yunlin University of Science and Technology, Douliu 64002, Taiwan; shihdh@yuntech.edu.tw (D.-H.S.); portraits1129@gmail.com (T.-W.W.); kk3329188@gmail.com (W.-C.T.)
[2] Department of Electrical and Computer Engineering, Iowa State University, 2520 Osborn Drive, Ames, IA 50011, USA
[3] Jesse H. Jones School of Business, Texas Southern University, 3100 Cleburne Street, Houston, TX 77004, USA; David.Yen@tsu.edu
* Correspondence: dannysmh@gmail.com

Abstract: Online auctions are now widely used, with all the convenience and efficiency brought by internet technology. Despite the advantages over traditional auction methods, some challenges still remain in online auctions. According to the World Business Environment Survey (WBES) conducted by the World Bank, about 60% of companies have admitted to bribery and manipulation of the auction results. In addition, buyers are prone to the winner's curse in an online auction environment. Since the increase in information availability can reduce uncertainty, easy access to relevant auction information is essential for buyers to avoid the winner's curse. In this study, we propose an Online Auction Price Suggestion System (OAPSS) to protect the data from being interfered with by third-party programs based on Intel's Software Guard Extensions (SGX) technology and the characteristics of the blockchain. Our proposed system provides a smart contract by using α-Sutte indicator in the final transaction price prediction as a bidding price recommendation, which helps buyers to reduce the information uncertainty on the value of the product. The amount spent on the smart contract in this study, excluding deployed contracts, plus the rest of the fees is less than US$1. Experimental results of the simulation show that there is a significant difference ($p < 0.05$) between the recommended price group and the actual price group in the highest bid. Therefore, we may conclude that our proposed bidder's price recommendation function in the smart contract may mitigate the loss of buyers caused by the winner's curse.

Keywords: online auction; winner's curse; blockchain; price recommendation; SGX technology

Citation: Shih, D.-H.; Wu, T.-W.; Shih, M.-H.; Tsai, W.-C.; Yen, D.C. A Novel Auction Blockchain System with Price Recommendation and Trusted Execution Environment. *Mathematics* **2021**, *9*, 3214. https://doi.org/10.3390/math9243214

Academic Editor: Jan Lansky

Received: 11 November 2021
Accepted: 9 December 2021
Published: 13 December 2021

1. Introduction

With advancing modern technology, E-commerce has become a part of daily life and has made considerable progress in recent years. In 2020, it is estimated that online transactions accounted for 25% of all business transactions. While more and more users have explored the business opportunities on the internet, the online auctions market has become an important business entity among them. Unlike traditional auctions, online auctions generate a lot of data during the transactions, including information about the products, participants, and related behaviors. If we studied and analyzed it properly, the data could bring huge benefits to buyers, sellers, and auction platforms.

The World Business Environment Survey (WBES) conducted by the World Bank has shown that approximately 60% of companies have been bribed and admitted to manipulating the online auction process, which ultimately affects the final results. A previous study has suggested that transparency of the overall process is a critical part of auctions [1].

Despite many benefits over traditional auctions, online auctions still cannot overcome some existing challenges. The winner's curse, for example, is a phenomenon where the

winner overpaid to win the auctions. It has been a challenge to bidders in traditional auctions and now in online auctions as well [2]. Past studies have indicated that when participants overestimate the value of an item in a competitive bidding environment, they will pay more than the market value and suffer the loss [3].

Providing relevant information, such as the price of the merchandise, can reduce the uncertainty and suppress the winner's curse. While past research has studied the causes and impacts of the winner's curse, few have suggested how to reduce or avoid it. In addition, how to leverage the data collected from online auctions to provide the bidders with more accurate information and recommendations remains in question.

In this study, we design an online auction system that provides auction procedures by writing smart contracts in general. Our system aims to avoid loss from the winner's curse using the characteristics of a blockchain and price recommendation on the auction items, which reduces the information uncertainty for bidders. In addition, we protect the system environment with Intel's Software Guard Extensions (SGX) technology to ensure all the information can be safely processed without manipulations from third parties or malicious intruders.

2. Background and Related Work

2.1. Online Auctions

Online auctions have grown substantially since the late 1990s. As an alternative form of retail with a dynamic pricing mechanism, electronic auctions have attracted many businesses and individual users who can buy and sell almost anything on the internet. In the second quarter of 2020, eBay (www.ebay.com, accessed on 1 June 2020), the current leader in electronic auctions, had 157 million active buyers worldwide, with 800 million products listed and 25 million sellers daily. The tremendous growth of electronic auctions has undoubtedly aroused great research interest.

Auctions have existed in human history for thousands of years. Klein and O'Keefe [4] believe auction is a standardized transaction procedure. With the restrictions of auction rules, participants bid and set the item price interactively. Traditionally, there are four main types of auctions [5]:

1. First Price Sealed Bidding Auction (FPSBA): The buyer seals the bid in an envelope and delivers it to the auctioneer. Subsequently, the auctioneer opens the envelope to determine the winner with the highest bid.
2. Second Price Sealed Bidding Auction (Vickrey auction): It is similar to FPSBA except that the winner will pay the second-highest bid.
3. Open Ascending Auction (English auction): Bidders make increasingly higher bids and stop bidding if they are unwilling to pay higher than the current highest bid.
4. Open Descending Auction (Dutch auction): The auctioneer initially sets a high price and then gradually reduces it until any buyer decides to pay at the current price.

2.2. Winner's Curse

The first study on the winner's curse was to discuss the rights of oil drilling [6]. Without enough information about the auction item, the buyer may have given up benefits or even suffered losses when the winner's curse occurred. Bazerman and Samuelson [7] conducted an experiment to prove the existence of the winner's curse. In a bidding auction, the person with the highest bid wins and the reason behind the high bidding price is because the person expects the item to be of a higher value.

According to a past study [8], during an auction, the participating buyers usually have insufficient information about the values of the auction items. The information obtained by each buyer is imbalanced, and the most optimistic buyer tends to win the bid. Therefore, it is common to overestimate the products, where the winner pays more than the actual value.

2.3. Price Recommendation and Prediction

The α-Sutte Indicator prediction method was proposed originally in 2017 as a new method to predict stock trends. Subsequent research has shown that this method can also be used to predict all-time series data [9]. During the prediction process, α-Sutte Indicator only needs the first four data points and does not require any hypothesis, providing the flexibility to analyze any type of data. We chose α-Sutte Indicator as our price prediction method in consideration of the auctioning environment and compatibility with the blockchains. This method provides better accuracy and is not limited to predicting stock price trends but can also be used to predict various time series data. Compared with ARIMA and other methods, this method is more capable of writing formula conditions in smart contracts and the cost of implementation is also lower. α-Sutte Indicator is described as the following Equation (1) [10]:

$$\alpha_t = \frac{\alpha\left(\frac{\Delta x}{\frac{\alpha+\delta}{2}}\right) + \beta\left(\frac{\Delta y}{\frac{\beta+\alpha}{2}}\right) + r\left(\frac{\Delta z}{\frac{r+\beta}{2}}\right)}{3} \quad (1)$$

where

δ = a(t − 4)
α = a(t − 3)
β = a(t − 2)
γ = a(t−1)
$\Delta x = \alpha - \delta$ = a(t − 3) − a(t − 4)
$\Delta y = \beta - \alpha$ = a(t − 2) − a(t − 3)
$\Delta z = \gamma - \beta$ = a(t − 1) − a(t − 2)
a(t) = the observation at time t
a(t − k) = the observation at time t − k

In the studies of machine learning and blockchains, most of the related algorithms are performed outside the blockchain [11] or experiment with external data sets [12]. However, the algorithm of α-Sutte Indicator can be integrated into the blockchain, can be used in a similar way to internal functions, can be compared with external use, is simpler, and is also the key to adopting this method in this study.

2.4. Blockchain

A blockchain with a distributed consensus protocol is a distributed ledger technology (DLT) that combines peer-to-peer networking, cryptography, and game theory, but the data structure of the blockchain itself is older than DLT [13]. It originated from Nakamoto's white paper [14]. When there is no verification or auditing mechanism, the trust issue to the information system will be extremely complex, especially with sensitive information, such as economic transactions using virtual currency. Nakamoto proposed two radical concepts in his research. The first is Bitcoin, a virtual cryptocurrency that maintains its value without the support of any centralized agency or financial entity. Instead, tokens are collectively and safely held by a decentralized network of P2P participants, which constitutes an auditable network. A blockchain is the second concept, and it has been more popular than cryptocurrency. Blockchain technology consists of six elements [15]:

1. Decentralized: The most basic feature of a blockchain is that the data do not rely on a centralized node but can be recorded and stored in a decentralized fashion.
2. Transparent: The data can be updated on any nodes in the blockchain system, which is the main contributor to the blockchain's trustworthiness.
3. Open source: Most of the blockchain systems are open to the public for inspection, verification, and usage to create other applications.
4. Autonomous: Based on the consensus algorithm, all nodes in the blockchain system can safely transmit and update data without intervention.

5. Unchangeable: All records will be stored forever and cannot be changed unless one party occupies at least 51% of the nodes at the same time.
6. Anonymous: A blockchain resolves the problem of trust between nodes, so data can be transmitted and traded in an anonymous manner, with only the blockchain address being known to each other.

2.5. Ethereum

In 1997, Szabo [16] defined a smart contract as a "computerized transaction agreement that enforces the terms of the contract." One key feature of a smart contract is having a way to execute contract terms on its own, which was not technically feasible until the blockchain was proposed. In fact, a blockchain is an ideal technology to support smart contracts, where smart contracts also contributed to the development of the blockchain, commonly known as blockchain 2.0. In the absence of centralized control, automated contract execution in a trusted environment could potentially change the traditional ways of business.

In summary, Ethereum technology has the ability to remove third parties from the environment while executing developers' applications on the blockchain. Smart contracts can execute different conditions according to different roles; the online auction situation also requires multiple roles and exclusive behaviors. In this study, we use Ethereum's smart contract with α-Sutte Indicator as the core of our system.

2.6. Blockchain-Based E-Auction

In the research of blockchain-based online auctions, Foti and Vavalis [17] proposed the design of the decentralized, real-time, unified-price double-auction energy market. Desai et al. [18] proposed a novel hybrid framework that combines private and public blockchains to help protect the privacy of participants. Wang and Mu [19] proposed a system framework that uses blockchain technology and smart contract to solve the privacy and security problems of E-bidding systems. Jiao et al. [20] proposed an auction-based market model for efficient computing resource allocation. Braghin et al. [21] developed an online auction system based on Ethereum smart contracts. Smart contracts are executable codes that run on the blockchain to facilitate and execute agreements between untrusted parties without the participation of trusted third parties.

In most of the studies on blockchain and online auctions, mainly aimed at the protection of privacy, the bidding process can be integrated into the blockchain without a third party, but it has not yet tried to integrate decision-making in the blockchain. This study attempts to present time series forecasting methods through smart contracts and provide the function of the bidder's price recommendation.

Most of the research on online auctions with blockchains tends to be decentralized, real-time, and smart-contract driven. However, it is quite rare to find price-related recommendations in the auction research with a blockchain. Ethereum provides different roles and smart contracts to help integrate the online auction situation process into the blockchain. α-Sutte Indicator, the time series forecasting method, is easier to write into smart contracts than other methods. This study tends to add a price recommendation function to smart contracts, which may mitigate the loss of buyers caused by the winner's curse.

2.7. Trusted Execution Environment

Trusted execution environments (TEE), such as Intel's Software Guard Extensions (SGX) and ARM TrustZone [22], are widely used in personal computers, servers, and mobile platforms, respectively. TEE provides an isolated execution environment that runs in parallel with the host operating system and standard cryptographic functions. In this study, we use Intel SGX as the TEE for our system.

Intel SGX is the technology developed by Intel with the main purpose of enhancing the security of executing programs. While it cannot identify or isolate all malicious programs on the platform, it packages the safe operations of legitimate programs in an enclave to protect them from malicious programs. Neither privileged nor unprivileged programs can

access this area. In other words, once the programs and data enter this security zone, they will not be affected even by the operating system. The security zone created by SGX can be considered a trusted execution environment.

In the past research on online auctions and blockchains, the bidding process was generally transplanted to the blockchain. This study mainly integrates the time series method α-Sutte Indicator on the Ethereum platform, provides price recommendations through smart contracts during the bidding process, and helps bidders reduce the chances of creating a winner's curse.

3. System Framework

3.1. System Environment

This study uses the online IDE environment Remix to write and test smart contracts and disassemble the α-Sutte Indicator formula and integrate it into the smart contract to provide price recommendations. After the bidding process is tested without errors, the final step is to conduct cost and safety analysis. Table 1 presents the system environment of this study.

Table 1. System environment.

Parameter	Value
OS	Windows 10
CPU	8-Core Intel(R) i7
RAM	32 GB
TEE	Intel SGX
Language	Solidity
IDE	Remix IDE

3.2. Roles

In this section, we define the roles in the environment of online auction bidding.

1. Buyer/Bidder: The role with the capability of bidding on items in an online auction.
2. Bidder of Decision Support: The role with bidding capability and price recommendations suggested by the system. The price predictions are based on data collected from past auctions.
3. Seller: The role of publishing product auctions and collecting payments with the capability of specifying item auction price, auction time, and other information in detail.
4. Auction Manager: The role of verifying information of the auctioned products or identities of all the other participating roles.

3.3. Auction Scenario

Figure 1 shows the complete process of online auctions, from the listing of the auctioned item to the receipt of the payment by sellers, and the description is as follows:

1. The seller sends the system a request to list the auctioned item.
2. Upon receipt of the seller's listing request, the auction manager verifies the product information and checks if there is any missing information.
3. Each buyer can pay for the registration to access the price recommendations before the auction starts.
4. Buyers who paid for the registration are converted to the role of "bidders of price suggestions."
5. Bidders of price suggestions can request a price recommendation.
6. The price recommendation request is sent to SGX for secure processing.
7. The price recommendation is calculated and returned to the bidder of price suggestions.
8. The auction manager starts the auction.

9. The system takes bids from all buyers.
10. The auction manager verifies the auction time and bid counts during the auction. The auction is closed at the end of auction time or if there exists a winner.
11. The auction manager verifies the winner's information.
12. The winning buyer submits the payment.
13. The seller verifies the payment.

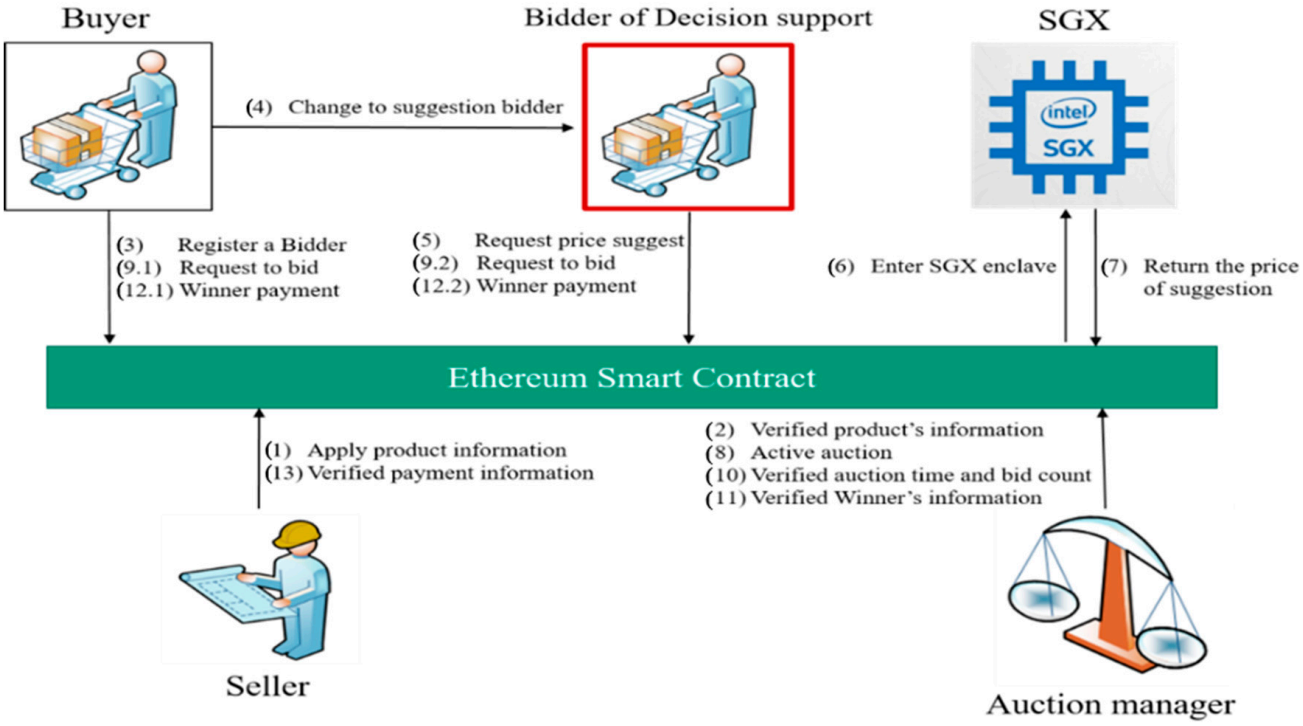

Figure 1. Online auction process scenario.

3.4. Ethereum Smart Contract

In this study, the smart contract is a program deployed on the Ethereum blockchain network, which contains pre-defined states, transition rules, execution conditions, and execution logic. When the conditions are met, the execution logic is automatically executed [23]. We designed a smart contract system called the Online Auction Price Suggestion System (OAPSS). Figure 2 shows the process of calling events for the entire auction. In the beginning, the auction manager deploys the smart contract and the auction platform using Deploy(). The seller calls SellerRegister() and the buyer calls BidderRegister() to register as seller and buyer, respectively. The seller calls the ApplyProduct() function when listing an item for auction, which executes VerifiedProductInformation() subsequently to verify the seller's information about the item. Before the auction starts, the buyer can call changeToSuggestBidder() to convert into the role of the bidder of the price suggestions. Buyers with successful conversions can then start RequestToPriceSuggest() to get price recommendations. To start the auction, the auction manager calls ActiveAuction(). During the active auction, any buyer can call RequestBid() to bid on the items. At the end of the auction, the auction manager calls AnnouncementWinner() to announce the winner of the auction and notify the buyer and the seller of the final price. The buyer calls WinnerPayment() to submit the payment to the seller. Based on smart contracts, the characteristics of the OAPSS framework are unchangeable, unalterable, and truthful. Table 2 is the overall OAPSS smart contract functions used in this study. And, Algorithms 1–8 are their detailed algorithm in smart contracts.

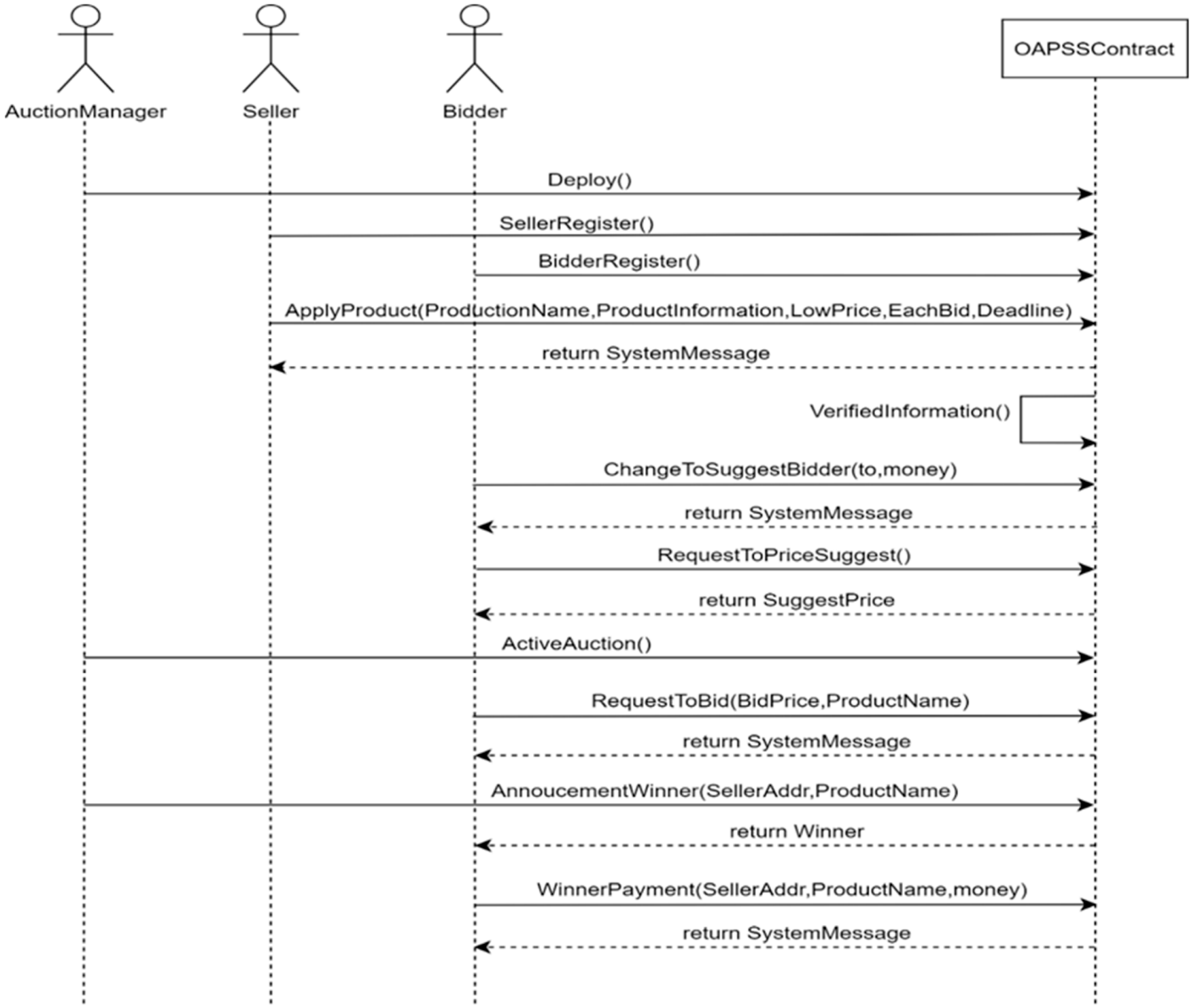

Figure 2. Sequence diagram of the OAPSS system.

Table 2. Overall OAPSS smart contract functions.

Function	Smart Contract Algorithm
Deploy the contract.	Deploy()
Register the seller.	SellerRegister()
Register the buyer.	BidderRegister()
List an auctioned item.	ApplyProduct()
Verify the auctioned item.	VerifiedProductInformation()
Register the buyer for a price suggestion.	ChangeToSuggestBidder()
The buyer requests a price suggestion.	RequestToPriceSuggest()
Start the auction.	ActiveAuction()
Bid on an item.	RequestToBid()
Announce the winner.	AnnoucementWinner()
The winner pays the seller.	Winner Payment()

ApplyProduct(): Sellers use this function to put the product information of the auctioned item on the system and wait for the system and auction managers to review it.

Algorithm 1 ApplyProduct

Input: Ethereumaddress(EA) of SellerAddr
ProductName, AuctionLowPrice, AuctionStartTime, AuctionEndTime
1. if *SellerAddr* = *Seller Address* **then**
2. Add product information to ProductHashtable
3. Setting Auction Time
4. str = Identity verification success
5. else
6. str = Identity verification failed
7. end

VerifiedProductInformation(): The auction manager verifies the product information and rejects the request with incomplete or incorrect information.

Algorithm 2 VerifiedProductInformation

Input: Ethereumaddress(EA) of AuctionManagerAddr
ProductName, AuctionLowPrice, AuctionTime
1. if *AuctionManagerAddr* = *AuctionManager Address* **then**
2. **if** *Product Information* <> null **then**
3. *AuctionReadyState* = true;
4. str = Product apply success
5. **else**
6. *AuctionReadyState* = false;
7. str = Product apply fail
8. else
9. str = Identity verification failed
10. end

ChangeToSuggestionBidder(): Buyers use this function to apply for conversion to a bidder of price suggestions. The system checks if the related fee has been collected, and it either approves or declines the conversion request.

Algorithm 3 ChangeToSuggestionBidder

Input: Ethereumaddress(EA) of BidderAddr
Fee
1. if *BidderAddr* = *Bidder Address* **then**
2. **if** *Fee* = true
3. Add Bidder to SuggestionBidderArrayList
4. str = change to Suggestion Bidder is success
5. **else**
6. str = change to Suggestion Bidder is fail
7. **end**
8. else
9. str = Identity verification failed
10. end

RequestToPriceSuggest(): Buyers who have converted can use this function to request a price recommendation for specific auctioned products. The system makes a prediction of the final price using α-Sutte Indicator, and the result is returned to the buyer.

Algorithm 4 RequestToPriceSuggest

Input: Ethereumaddress(EA) of BidderPriceSuggestionAddr
ProductName
1. if BidderPriceSuggestion*Addr* = *Bidder of price suggestion Address* **then**
2. EnterSGXenclave
3. collect product information
4. use α-Sutte indicator to predict price
5. return *suggest price;*
6. else
7. str = Identity verification failed
8. end

ActiveAuction(): The auction manager uses this function to start the auction when ready. Buyers can then start to place bids on the items.

Algorithm 5 ActiveAuction

Input: Ethereumaddress(EA) of AuctionManagerAddr
ProductName, AuctionTime, AuctionReadyState
1. if *AuctionManagerAddr* = *AuctionManager Address* **then**
2. **if** *AuctionReadyState* = true
3. **while** *AuctionActive* = false
4. **if** *AuctionStartime* = now **then**
5. *AuctionActive* = true
6. str = Auction Start
7. **else**
8. *AuctionActive* = false
9. str = Auction time is not up yet
10. **end**
11. **else**
12. str = Auction not ready, please check product information
13. **end**
14. else
15. str = Identity verification failed
16. end

RequestToBid(): Both types of buyers can use this function to place a bid on the auctioned item. This function checks whether the bid amount is higher than the current highest price, update the current highest price if it exceeds it, and keep the bidder's information.

Algorithm 6 RequestToBid

Input: Ethereumaddress(EA) of BidderAddr
ProductName, BidPrice, AuctionTime
1. if *BidderAddr* = *Bidder Address* **then**
2. **if** *BidPrice* $\geq$ *Base standard* && *BidPrice* > *CurrentHighestPrice* **then**
3. **if** *now* < *AuctionEndTime* then
4. CurrentHighestPrice = BidPrice
5. CurrentHighestBidder = Bidder
6. str = Bid success
7. **else**
8. str = Bid fail
9. **end**
10. **else**
11. str = Bid fail
12. **end**
13. else
14. str = Identity verification failed
15.end

AnnouncementWinner(): The auction manager can use this function to conclude the auction with the highest bidding price and the winner. This function first checks whether the auction time exceeds the originally scheduled time. If the time has been exceeded, it stops the buyer from bidding and announces the current highest bidder and the final price.

Algorithm 7 AnnoucementWinner

Input: Ethereumaddress(EA) of AuctionManagerAddr
if *AuctionManagerAddr = AuctionManager Address* **then**
  if *now < AuctionEndTime* **then**
    Get HighestBidder
    Winner = HighestBidder
    Add Winner to WinnerNoPayArrayList
    Notify Winnerto payment
    str = Auction End
  else
    str = Auction time is not up yet
  end
else
    str = Identity verification failed
end

WinnerPayment(): The winner can use this function to make the payment to the seller after a successful bid.

Algorithm 8 WinnerPayment

Input: Ethereumaddress(EA) of BidderAddr
PaymentAmount
if *BidderAddr = WinnerNoPayArrayList* **then**
  if *PaymentAmount = CurrentHighestPrice* **then**
    transfer winner money to smart contract
    str = wait to seller receive payment
  else if *PaymentAmount > CurrentHighestPrice* **then**
    return PaymentAmount - CurrentHighestPrice
    str = wait to seller receive payment
  else
    str = Amount is enough
  end
else
    str = Identity verification failed
end

4. Testing and Security Analysis

4.1. Deploy Results

We present the deployment results of our system, OAPSS. First, we set the account addresses for each role in the auction scenario, as shown in Table 3: buyers (B), buyers with price prediction (BP), the seller (S), and the auction manager (AM). Then we use the accounts to test the smart contracts through Remix IDE.

Table 3. Role account address.

Account	Address
B	0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2
BP	0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db
S	0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB
AM	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

4.1.1. Deploy Contracts

When creating an OPASS smart contract, the creator will be set as the auction manager and the smart contract will be deployed. The result of the creation screen as an example is shown in Figure 3.

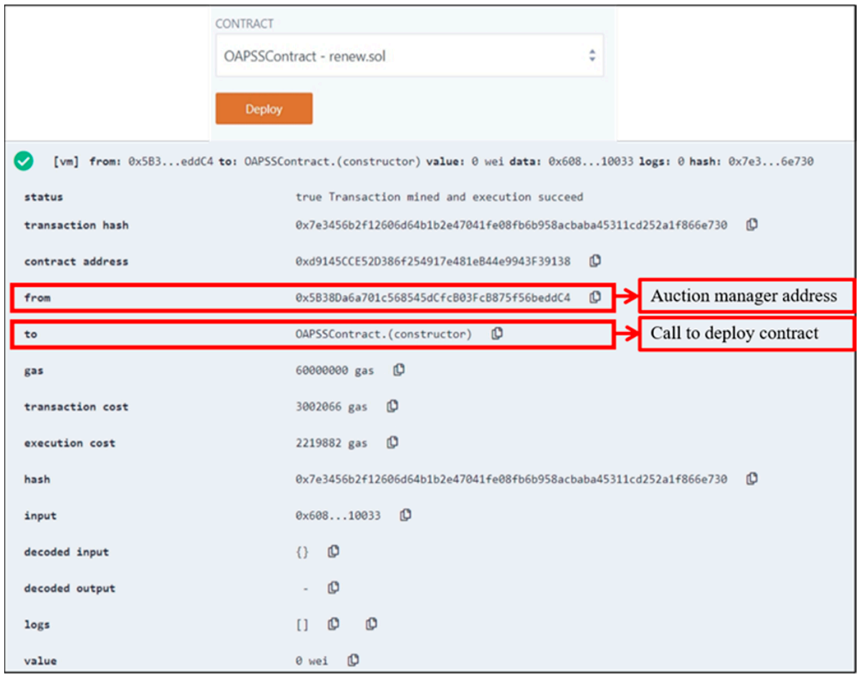

Figure 3. Deploy contract.

4.1.2. Winner Announcement

The auction manager enters the product name and seller address in the auction to end the auction, settle the winning bid amount, and announce the winner. The result of the winner announcement is shown in Figure 4.

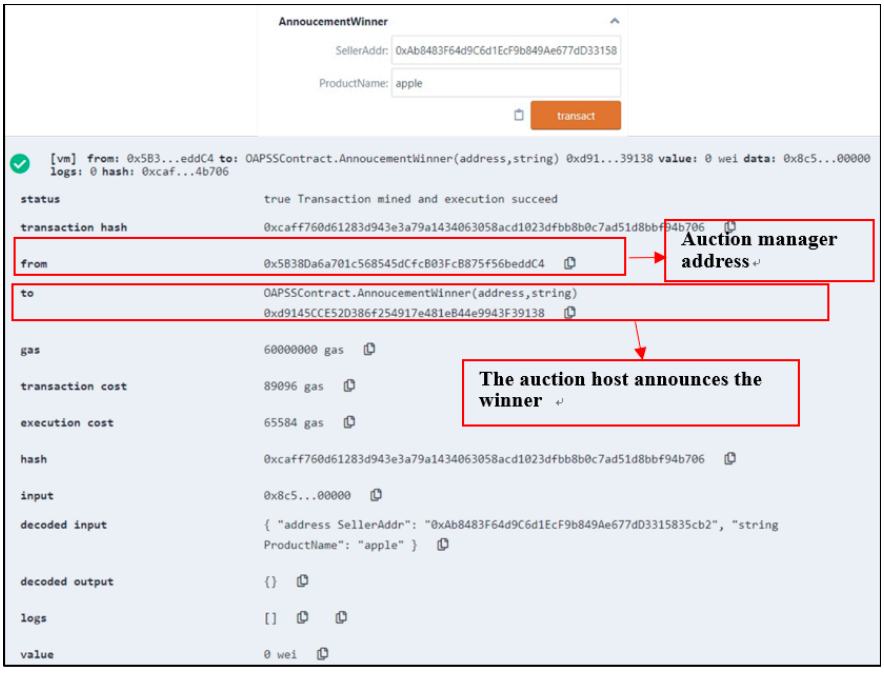

Figure 4. Winning bidder announcement.

4.2. Experiment on Winner's Curse

To understand whether the final transaction price prediction function of the OAPSS system can help the bidder avoid the winner's curse in a practical auction environment, we simulate the online auction platform of the eBay environment. The flow chart of experiment is shown in Figure 5. The purpose of this quasi-experiment evaluation is as follows:

1. Confirm the existence of the winner's curse.
2. Compare the difference between two scenarios, with and without the final transaction price prediction.

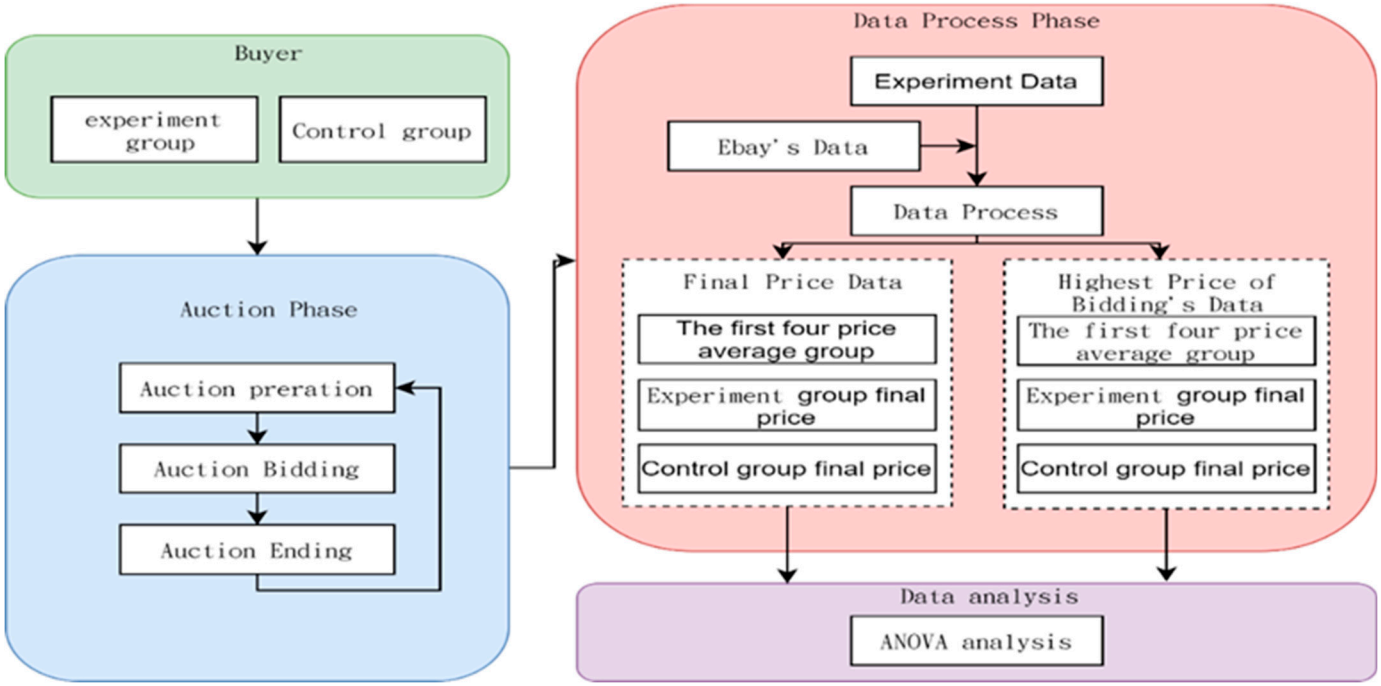

Figure 5. Flowchart of experiment.

4.2.1. Framework and Methods

- Buyer: The buyers are divided into the experimental group (with price prediction) and the control group (without price prediction) for comparison before the auction stage. Detailed group descriptions are shown in Table 4.
- Auction Stage: Fifteen items of various types are introduced to each buyer group for the auctions. The auction procedure includes price recommendation, bidding, product acquisition, winner announcement, and final payment.
- Data Aggregation: The two sets of 15 final transaction prices and the highest bids obtained from both buyer groups are collected. There are three groups, which are with prediction price (PdP) help group, without prediction price (NPdP) help group, and past prices (PP). Each price in PP is the average of the past four final prices collected from eBay. Table 3 summarizes the data groups and their definitions.
- Data Analysis: We use analysis of variance (ANOVA) and Tukey's post-analysis [24] to evaluate the differences between groups. The goal of this analysis is to verify our price predictions on the final transaction price and how it impacts the winner's curse. The ANOVA analysis of this experiment is shown in Tables 5 and 6.

Table 4. Definitions of groups.

Groups	Abbreviation	Definitions
Average of the past four final prices	PP (past price)	The past four final transaction prices for each of the 15 auctioned items from eBay
Experimental group	PdP (with prediction price help)	The final price and highest biddings with buyers using price recommendations for each of the 15 auctioned items
Control group	NPdP (without prediction price help)	The final price and highest biddings without price recommendations for each of the 15 auctioned items

Table 5. Descriptive statistics of the transaction prices.

	N	Avg	Std	Min	Max
PP	15	4699.85	5165.55	634.25	21,301.5
PdP	15	6220.67	3942.58	1690	15,060
NPdP	15	6942	4867.39	1010	15,510

Table 6. Descriptive statistics of the highest bid price.

	N	Avg	Std	Min	Max
PP	15	4699.85	5165.56	634.25	21,301.5
PdP	15	7468	5266.40	1800	20,000
NPdP	15	10,434.67	5866.46	2700	20,000

4.2.2. Experimental Results

To verify the difference of groups, ANOVA with the clusters as a covariate was performed to verify the significance [25]. Table 7 summarizes the results from ANOVA. The probability of a type I error was set to 0.05. We can see that there is no significant difference in the final transaction prices between different groups in Table 8 as there is no significant value below 0.05, and Tukey's post-analysis on the transaction price has shown similar results in Table 9. The asterisk represents their difference is significant ($p < 0.05$).

However, judging from the comparison of the highest bids among groups, it can be seen in Table 9 that there is a significant difference in the highest bids between the PP and the NPdP groups ($p = 0.17 < 0.05$). It indicates that without final price prediction (or recommendation), buyers may overbid in an auction. In addition, there is no significant difference between PP and PdP groups in the highest bid ($p = 0.354 > 0.05$) in Table 8. It means that if giving the final transaction price prediction (or recommendation) can cause the highest bid to be close to the final transaction price, the buyer may reduce the loss or escape the winner's curse.

Table 7. ANOVA.

	Sum of Squares	F	Sig.
Comparison between the transaction prices	39,302,208.67	0.894	0.417
Comparison between highest bids	246,759,438.67	4.167	0.022 *

Table 8. Tukey's test on the transaction price.

(I) Group	(J) Group	Mean Difference (I–J)	Sig.
PP	PdP	−1520.82	0.650
	NPdP	−2242.15	0.398
PdP	PP	1520.82	0.650
	NPdP	−721.33	0.907
NPdP	PP	2242.15	0.398
	PdP	721.33	0.907

Table 9. Tukey's test on the highest bid.

(I) Group	(J) Group	Mean Difference (I–J)	Sig.
PP	PdP	−2768.15	0.354
	NPdP	−5734.81 *	0.017 *
PdP	PP	2768.15	0.354
	NPdP	−2966.67	0.304
NPdP	PP	5734.82 *	0.017 *
	PdP	2966.67	0.304

4.3. Cost Analysis

Executing a smart contract and calling the functions in the Ethereum environment will consume gas. The gas consumption is based on the complexity of each function, which can be considered as a handling fee. The cost of gas consumption is calculated by the amount of gas consumed times the unit gas price. During the execution of a transaction, gas consumption could be restricted by the gas limit parameter to avoid malicious users from attacking the smart contract by executing functions arbitrarily and preventing the extra consumption caused by executing the wrong process. In this study, we set the gas limit to 6,000,000 units when testing OAPSS. Table 10 summarizes the costs of each function call in our proposed OAPSS. Note that the conversions between gas units and US dollars are according to the data from the CoinGecko website in February 2021, where 1 gas unit of ETH was equivalent to US$1545.82 for conversion. The amount spent on the smart contract, excluding deployed contracts, plus the rest of the fees is less than $1.

Table 10. The cost of function gas of the proposed OAPSS system.

Function Name	Transaction Cost	Execution Cost	USD
Deploy Contract	2,982,237	2,220,393	4.61
SellerRegister()	63,787	42,515	0.10
BidderRegister()	45,034	23,762	0.07
ApplyProduct()	195,003	170,787	0.30
VerifiedInformation()	33,802	11,122	0.05
ChangeToSuggestionBidder()	99,075	76,203	0.15
RequestToPriceSuggest()	44,350	21,926	0.07
ActiveAuction()	33,078	40,654	0.05
RequestToBid()	102,525	79,909	0.16
AnnoucementWinner()	104,576	110,744	0.16
WinnerPayment()	38,580	14,556	0.06

4.4. Security Analysis

The research by Luu et al. [26] addressed the security concerns of smart contracts and proposed solutions for specific attacks. The common vulnerabilities of a smart contract

are reentrancy vulnerability, replay attack, access restriction, and timestamp dependency. Many viewpoints, such as confidentiality, data integrity, availability, authorization, and non-repudiation, have been put forward in the security analysis [27–31].

- Reentrancy Vulnerability

When a user makes function calls in a smart contract, it could involve transferring remittances, where the reentrancy vulnerability could be caused by the sequence of calls. In other words, if the remittance is transferred before the states change, an attacker can create a new contract through the loophole to steal the Ether in the victimized contract. In this study, our smart contract verifies the identities of each role and related data using the require() function. Only if verified can a user make function calls. Identity verification prevents reentrancy vulnerability from causing damages and financial loss to smart contracts.

- Replay Attack

A replay attack is a malicious action that repeats or delays legitimate data transmissions on the network. It can be performed by the initiator or the middleman who intercepts and retransmits the data as part of a spoofing attack through IP packet replacement. This attack has been resolved by the subsequent Geth 1.5.3 update on smart contracts, and thus we do not consider it as a threat to our system in this study.

- Access Restriction

Access restriction, or access control (AC), is to manage and restrict access to certain spaces or resources. In this study, we implement the modifier() function to restrict the identities from accessing function calls unless the identity has sufficient rights.

- Timestamp Dependency

In the smart contract design, block.timestamp or now is often used to obtain the timestamp of a block in blockchain. A malicious miner can obtain a certain degree of knowledge at the right time. Therefore, any usage of timestamps in calculations should be carefully reviewed. In this study, we did not use block.timestamp or now for any calculation of money or sequence, and hence our system is not vulnerable to this attack.

- Confidentiality

The auction participants in this study can register and change their roles through smart contracts without entering other private information and can watch the corresponding information during the bidding process. Each stakeholder will be authenticated by their Ethereum address to protect their identity.

- Data Integrity

Blockchain technology maintains the integrity and immutability of data because the distributed ledger does not allow modification, addition, and deletion of data [32]. Any data modifications required are re-entered into the ledger as a new transaction. Therefore, all participants can view the data history at any point in time.

- Availability

This describes that data can only be accessed by authorized users. It also refers to the ability of the technology to provide data even in the presence of malicious code or denial-of-service attacks. This study can only be accessed by registered roles, but it has not been tested in situations of malicious code.

- Authorization

Authorization is related to the access rights provided by different people in the network. In the OAPSS system of this study, different roles have relative smart contracts based on their character. Therefore, it must be authorized.

- Non-repudiation

Stakeholders in the blockchain network cannot deny the actions or transactions they perform. The roles involved in this study conduct transactions through Ethereum addresses, and the relevant results are also presented in the blockchain network. Therefore, members of OPASS cannot be denied that a specific payment has not been received.

- Sybil Attack

A Sybil attack is a type of attack seen in peer-to-peer networks in which a node in the network operates multiple identities actively at the same time and undermines the authority/power in reputation systems.

- Double-Spend Attack

The idea of a double-spend attack is to use the same money for two (or more) different payments, creating conflicting transactions. Double-spending can be thought of as fraudulently spending the same cryptocurrency, or units of value, more than once.

Integrating relevant results into an IDE environment for presentation is a common situation in many blockchain studies. For example, [27] built automated healthcare contracts on the blockchain network and implemented them through the Remix IDE. This study is compared with the safety analysis proposed in [27,28,31], as shown in Table 11. Refs. [28,31] explain the importance of scalability in security analysis, and [31] adds more attack methods on the blockchain.

Table 11. Security comparison of different schemes.

	[27]	[28]	[31]	Our Study
Confidentiality	✓	✓	✓	✓
Data integrity	✓	✓	-	✓
Availability	✓	-	✓	-
Authorization	✓	✓	✓	✓
Non-repudiation	✓	✓	✓	✓
Scalability	×	✓	✓	×
Sybil attack	×	×	✓	✓
Double-spend attack	✓	✓	✓	✓
Man in the middle attack	✓	×	✓	✓

(✓: stands for done. ×: represents not provided or done. -: stands for uncertainty).

5. Conclusions and Future Work

In the current auction environment, it is possible that specific persons or internal personnel may manipulate the auction process, thereby affecting the final price of the transaction and the winner. This study aims to provide the transparency of the auction process and prevent manipulation of the auction by establishing a transparent online auction system using blockchain technology to store records and auction data in a trusted execution environment.

In addition, the buyers are prone to the winner's curse in an auctioning environment. To mitigate the loss caused by the winner's curse, this study uses the α-Sutte Indicator prediction method to provide a system-recommended price on the auctioned item for registered buyers. We have proposed a systematic framework to provide a better online auction infrastructure. To the best of our knowledge, this is the first study to provide price recommendations in the blockchain environment for online auctions. The amount spent on the smart contract in this study, excluding deployed contracts, plus the rest of the fees is less than $1.

This study compares other studies that combine the blockchain with online auctions. From Table 12, it can be seen that although this study does not conduct a follow-up analysis for scalability, in the experiment with the highest bid, there is a significant difference

between the actual price group and the recommended price group ($p < 0.05$). This study provides a price recommendation in a smart contract that may mitigate the loss of buyers caused by the winner's curse.

Table 12. Comparison with different studies.

	[17]	[18]	[21]	**Our Study**
Environment setup	Ethereum	Hybrid blockchain architecture	Ethereum	Ethereum
Privacy protection	✓	✓	✓	✓
Decentralization	✓	✓	✓	✓
Scalability	✓	✓	×	×
Cost analysis	✓	✓	✓	✓
Final price recommendation	×	×	×	✓
Trusted execution environment	×	×	×	✓

(✓: stands for done. ×: represents not provided or done).

Due to the limitations of the Solidity language and the Remix IDE compiler, we were not able to apply deep learning methods to the blockchain systems for price prediction in our system. In addition, transactions in our system are based on Ethereum, which has a large fluctuation in the exchange rate to US dollars and it may not be a good and stable candidate for the trading currency of online auctions. As in the future, we plan to study and implement other prediction methods using the Solidity language and compare the performance of different methods.

The advantage of this study is that online auctions are integrated into the blockchain environment to provide price recommendations and write the time series forecasting method directly into the smart contract, instead of making predictions outside the blockchain. It is just that α-Sutte Indicator requires at least four pieces of historical data to make predictions, and if the historical record of the items to be auctioned in the future is relatively unpopular, the prediction effect will be limited.

This study chooses to integrate online auctions and time series forecasting into the blockchain. In the future, more time series research can also be conducted in other fields, such as renewable energy forecasting [9], COVID-19 confirmed cases, and stock market prices [33]. With different roles and smart contracts, it is possible to establish a stock price-related investment platform and an early warning platform for the number of infections.

This study is one of the few that incorporate time series forecasting methods into the blockchain and provide price recommendations to bidders, helping them reduce the occurrence of the winner's curse. In the future, in addition to α-Sutte Indicator, gray prediction theory can be integrated into the blockchain, providing appropriate decisions based on different situations.

The source code of our system is shared on Github: https://github.com/kk3329188/lib/blob/main/OAPSS.

Author Contributions: Conceptualization, D.-H.S.; data curation, W.-C.T.; formal analysis, T.-W.W. and W.-C.T.; investigation, M.-H.S. and W.-C.T.; methodology, D.-H.S. and T.-W.W.; project administration, D.-H.S. and D.C.Y.; resources, M.-H.S.; software, W.-C.T. and D.C.Y.; supervision, D.-H.S.; validation, T.-W.W.; visualization, D.C.Y.; writing—original draft, T.-W.W.; writing—review and editing, M.-H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Taiwan Ministry of Science and Technology (grants MOST 109-2410-H-224-022 and MOST 110-2410-H-224-010). The funder has no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Olaya, J.; Boehm, F. Corruption in Public Contracting Auctions: The Role of Transparency in Bidding Process. *Ann. Public Coop. Econ.* **2006**, *77*, 431–452.
2. Amyx, D.A.; Luehlfing, M.S. Winner's curse and parallel sales channels—Online auctions linked within e-tail websites. *Inf. Manag.* **2006**, *43*, 919–927. [CrossRef]
3. Milgrom, P.R.; Weber, R.J. A Theory of Auctions and Competitive Bidding. *Econometrica* **1982**, *50*, 1089. [CrossRef]
4. Klein, S.; O'Keefe, M. The Impact of the Web on Auctions: Some Empirical Evidence and Theoretical Considerations. *Int. J. Electron. Commer.* **1999**, *3*, 7–20. [CrossRef]
5. Krishna, V. *Auction Theory*; Academic Press: Cambridge, MA, USA, 2009.
6. Capen, E.C.; Clapp, R.V.; Campbell, W.M. Competitive Bidding in High-Risk Situations. *JPT J. Pet. Technol.* **1971**, *23*, 641–653. [CrossRef]
7. Bazerman, M.H.; Samuelson, W.F. I Won the auction but don't want the prize. *J. Confl. Resolut.* **1983**, *27*, 618–634. [CrossRef]
8. Goeree, J.; Offerman, T. Winner's curse without overbidding. *Eur. Econ. Rev.* **2003**, *47*, 625–644. [CrossRef]
9. Ahmar, A.S. A Comparison of α-Sutte Indicator and ARIMA methods in renewable energy forecasting in Indonesia. *Int. J. Eng. Technol.* **2018**, *7*, 20–22. [CrossRef]
10. Ahmar, A.S. Sutte indicator: An approach to predict the direction of stock market movements. *Songklanakarin J. Sci. Technol.* **2018**, *40*, 1228–1231. [CrossRef]
11. Bhowmik, M.; Chandana, T.S.S.; Rudra, B. Comparative Study of Machine Learning Algorithms for Fraud Detection in Blockchain. In Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 8–10 April 2021; pp. 539–541.
12. Cheema, M.A.; Ashraf, N.; Aftab, A.; Qureshi, H.K.; Kazim, M.; Azar, A.T. Machine Learning with Blockchain for Secure E-voting System. In Proceedings of the 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), Riyadh, Saudi Arabia, 3–5 November 2020; pp. 177–182.
13. Haber, S.; Stornetta, W.S. How to time-stamp a digital document. *J. Cryptol.* **1991**, *3*, 99–111. [CrossRef]
14. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* **2008**, 21260. Available online: https://www.debr.io/article/21260.pdf (accessed on 1 December 2021).
15. Lin, I.-C.; Liao, T.-C. A survey of blockchain security issues and challenges. *Int. J. Netw. Secur.* **2017**, *19*, 653–659. [CrossRef]
16. Szabo, N. Formalizing and securing relationships on public networks. *First Monday* **1997**, *2*. [CrossRef]
17. Foti, M.; Vavalis, M. Blockchain based uniform price double auctions for energy markets. *Appl. Energy* **2019**, *254*, 113604. [CrossRef]
18. Desai, H.; Kantarcioglu, M.; Kagal, L. A Hybrid blockchain architecture for privacy-enabled and accountable auctions. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; pp. 34–43.
19. Wang, D.; Zhao, J.; Mu, C. Research on Blockchain-Based E-Bidding System. *Appl. Sci.* **2021**, *11*, 4011. [CrossRef]
20. Jiao, Y.; Wang, P.; Niyato, D.; Suankaewmanee, K. Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks. *IEEE Trans. Parallel Distrib. Syst.* **2019**, *30*, 1975–1989. [CrossRef]
21. Braghin, C.; Cimato, S.; Damiani, E.; Baronchelli, M. Designing smart-contract based auctions. In *International Conference on Security with Intelligent Computing and Big-Data Services*; Springer: Cham, Switzerland, 2019; pp. 54–64.
22. McKeen, F.; Alexandrovich, I.; Berenzon, A.; Rozas, C.V.; Shafi, H.; Shanbhogue, V.; Savagaonkar, U.R. Innovative instructions and software model for isolated execution. In Proceedings of the HASP'13: The Second Workshop on Hardware and Architectural Support for Security and Privacy, Tel-Aviv, Israel, 23–24 June 2013; p. 1. [CrossRef]
23. Sun, J.; Huang, S.; Zheng, C.; Wang, T.; Zong, C.; Hui, Z. Mutation testing for integer overflow in ethereum smart contracts. *Tsinghua Sci. Technol.* **2022**, *27*, 27–40. [CrossRef]
24. Magalhães, F.A.; Souza, T.R.; Araújo, V.L.; Oliveira, L.M.; de Paula Silveira, L.; de Melo Ocarino, J.; Fonseca, S.T. Comparison of the rigidity and forefoot—Rearfoot kinematics from three forefoot tracking marker clusters during walking and weight-bearing foot pronation-supination. *J. Biomech.* **2020**, *98*, 109381. [CrossRef]
25. Harb, H.; Makhoul, A.; Couturier, R. An enhanced K-means and ANOVA-based clustering approach for similarity aggregation in underwater wireless sensor networks. *IEEE Sens. J.* **2015**, *15*, 5483–5493. [CrossRef]
26. Luu, L.; Narayanan, V.; Zheng, C.; Baweja, K.; Gilbert, S.; Saxena, P. A Secure sharding protocol for open blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 17–30.
27. Omar, I.A.; Jayaraman, R.; Debe, M.S.; Salah, K.; Yaqoob, I.; Omar, M. Automating procurement contracts in the healthcare supply chain using blockchain smart contracts. *IEEE Access* **2021**, *9*, 37397–37409. [CrossRef]

28. Xiong, W.; Xiong, L. Data Trading Certification Based on Consortium Blockchain and Smart Contracts. *IEEE Access* **2021**, *9*, 3482–3496. [CrossRef]
29. Karpinski, M.; Kovalchuk, L.; Kochan, R.; Oliynykov, R.; Rodinko, M.; Wieclaw, L. Blockchain Technologies: Probability of Double-Spend Attack on a Proof-of-Stake Consensus. *Sensors* **2021**, *21*, 6408. [CrossRef]
30. Longo, R.; Podda, A.S.; Saia, R. Analysis of a Consensus Protocol for Extending Consistent Subchains on the Bitcoin Blockchain. *Computation* **2020**, *8*, 67. [CrossRef]
31. Cui, Z.; Fei XU, E.; Zhang, S.; Cai, X.; Cao, Y.; Zhang, W.; Chen, J. A hybrid blockchain-based identity authentication scheme for multi-WSN. *IEEE Trans. Serv. Comput.* **2020**, *13*, 241–251. [CrossRef]
32. Abu-Elezz, I.; Hassan, A.; Nazeemudeen, A.; Househ, M.; Abd-Alrazaq, A. The benefits and threats of blockchain technology in healthcare: A scoping review. *Int. J. Med. Inform.* **2020**, *142*, 104246. [CrossRef] [PubMed]
33. Ahmar, A.S.; del Val, E.B. SutteARIMA: Short-term forecasting method, a case: Covid-19 and the stock market in Spain. *Sci. Total Environ.* **2020**, *729*, 138883. [CrossRef]