CryptoPunk – All-in-one Crypto Manager

Ankit Kumar[1], Vicky[1], Jatin Goyal[1], Kushal Gupta[1]

Student, Department of Computer Science & Engineering, HMR Institute of Technology & Management, Delhi, India[1]

Abstract: Crypto is the future of transactions in the upcoming future and people are moving towards it. But people should have knowledge of how to use and where to use their crypto. It is a great responsibility of ours to provide a resourceful and trust based platform to the people. In order to provide these services, we're implementing Cryptopunk to help the people with no charges and no service fees.

Keywords: Cryptocurrency, Deep Learning, NFT, LSTM, Prediction

I. INTRODUCTION

Cryptocurrencies are digital financial assets whose ownership and transfer of ownership are guaranteed by cryptographic decentralized technology. The rising values of cryptocurrencies in the market and their growing popularity around the world have created many challenges and concerns in business and industrial economics. This introductory article describes key trends in academic research related to cryptocurrencies and uses both neo-classicalism and behavioral theory to highlight the contribution of selected papers to the literature.

Since then, blockchain has been seen as a new technology for decentralized transactional data exchange across large networks of un-trusted participants. This enables a new form of distributed software architecture that allows consensus on common states without relying on central integration points. It enables the creation of a decentralized environment, where transactions and data are not under the control of any third party organization. Any transaction ever completed is recorded in a public ledger in a verifiable, secure, transparent and permanent way, with a timestamp and other details.

For these features blockchain has developed into one of today's biggest groundbreaking technologies with potential to impact every industry from financial to manufacturing to educational institutions.

II. BACKGROUND

As crypto is the future of the market, we choose crypto and blockchain-based applications to implement and deploy the platform, giving people a great vision of crypto. Our vision is to make sure people know the crypto and invest wisely. The best way to learn something is to present it, which is what we wanted to do. To develop a platform where people can better understand and invest in cryptography. Through this application, we will provide you with a list of the most used cryptocurrencies in the world and their prices. We will be giving access for tracking these various cryptocurrencies in real time. The main goal of this application is to provide an interface for learning about crypto and blockchain as a technology. We will also provide some basic knowledge about how blockchain is supposed to function, what are its properties, current values of crypto coins etc. Crypto is written in python and it will be released under the AGPL licence. Our backend is written in node.js.

The current goal is to learn how to use the platform to send transactions, generate blocks, and build decentralized applications (dapps), so how many to run and try the software? I need that cryptocurrency! Therefore, users can use their test network to explore how applications work. Cryptocurrencies are the safest way to send money over the internet. Today, people are asking many questions about cryptocurrencies. For example, why crypto is better than other cryptocurrencies, what are some good crypto wallets, and where to buy crypto.
 Cryptocurrencies are called virtual currencies because they are digital currencies. Unlike fiat money that has some physical properties like USD, Euro, or GBP etc., this money is not printed physically as we do with cash or banknotes. However, digital money has been in use for years now. This kind of currency has been in use since 2009 with the introduction and rise of Bitcoin. It has become such a beneficial way to make transactions that it's almost impossible to ignore Cryptocurrency in the age of the internet.

III. OBJECTIVE

Following the expected usage and future use cases of blockchain and cryptocurrencies, we decided to create an application based on deep learning and cryptocurrency using blockchain technology. This application consists of crypto market analysis, crypto prediction, NFT listing and ownership, sending and receiving crypto blockchain smart contracts, and blockchain authenticated crypto user profiles.

Market analysis can mitigate risk, identify new trends, and support the sale of cryptocurrencies. You can use marketing analysis at several stages of your business. It is also beneficial to do it every year to keep up with the big changes in the market.

People can alternate contracts that pay primarily based totally on the effects of unknown destiny events. The marketplace costs generated from those contracts may be understood as a type of collective prediction amongst marketplace participants. These costs are primarily based totally on the character expectancies and willingness of traders to place their cash on the road for the ones expectancies.

NFTs are a new type of digital asset. Ownership is recorded on the blockchain — a digital ledger that resembles the underlying network of Bitcoin and other cryptocurrencies. Each NFT is unique and cannot be duplicated. Therefore, they can be thought of as unique digital items that no one else owns. Sure, for example, some people may have images of artwork purchased as NFTs, but they don't own the original. That's what makes them attractive. And since they can be rare, you may be able to sell them later for more money. Like art.

A smart contract is a program stored in the blockchain and is executed when certain conditions are met. These are typically used to automate the execution of contracts, so all involved parties can see the results immediately, without the involvement of intermediaries or the loss of time. You can also automate the workflow and trigger the next action when the conditions are met.

IV. METHODOLOGY

Long Short Term Memory (LSTM) is a specific recurrent neural network architecture (RNN) is designed to model time sequences and their long-distance dependencies more accurately than traditional RNNs.

A. *LSTM Network Architectures*

I) *Conventional LSTM*

The LSTM contains a special unit called a memory block in the repeating hidden layer. The memory block contains a self-connected memory cell that stores the temporal state of the network, in addition to a special productive unit called a gate that controls the flow of information. Each memory block in the original architecture contained an input gate and an output gate. The input gate controls the flow of input to memory cells [7]. The output gate controls the output flow of cell activation to the rest of the network. Then the forget gate was added to the memory block. This fixes a weakness in the LSTM model that was unable to handle continuous input streams that were not segmented into subsequences. The forgetting gate scales the internal state of the cell and then adds it as input to the cell via the cell's self-repeating connection. This adaptively forgets or resets the cell's memory [9]. In addition, the latest LSTM architecture includes a peephole connection from an internal cell to the gate of the same cell to learn the exact timing of the output.
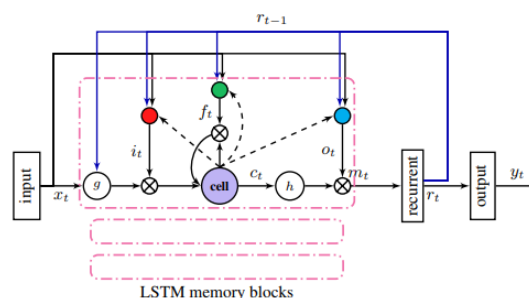

Fig: 1 Architecture of Conventional LSTM

II) Deep LSTM

Deep LSTMRNNs are built by stacking multiple LSTM layers. Note that LSTM-RNNs are already a deep architecture in the sense that each layer can be thought of as a time-expanded feed forward neural network with the same model parameters. Inputs to the model go through multiple non-linear layers as in DNN, but features at a particular point in time can only be processed by a single non-linear layer before contributing to the output at that point in time to understand. Therefore, the depth of the deep LST MRNN [7] has additional implications. Inputs to the network at a particular time step not only propagate through the time and LSTM layers, but also through multiple LSTM layers. It has been argued that the deep layers of RNNs allow networks to learn about inputs on different time scales [2]. Deep LSTM RNNs offer another advantage over standard LSTM RNNs. In other words, you can make better use of your parameters by distributing them in a space that spans multiple layers [8]. For example, instead of doubling the memory size of the standard model, you can have four layers with about the same number of parameters. This causes the input to undergo more non-linear operations per time step.
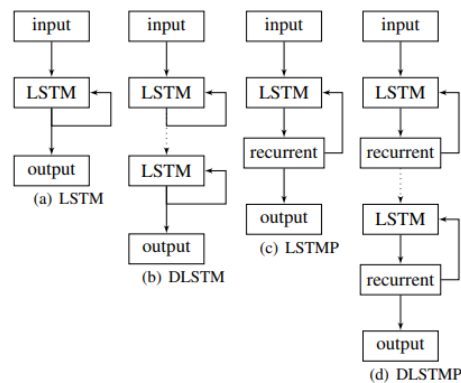

Fig: 2 Architecture of DeepLSTM

B. *Blockchain transaction flow*

Smart contracts work by following simple "if/when…then…" statements that are written into code on a blockchain. A network of computers executes the actions when predetermined conditions have been met and verified. These actions could include releasing funds to the appropriate parties, registering a vehicle, sending notifications, or issuing a ticket. The blockchain is then updated when the transaction is completed. That means the transaction cannot be changed, and only parties who have been granted permission can see the results.

Within a smart contract, there can be as many stipulations as needed to satisfy the participants that the task will be completed satisfactorily. To establish the terms, participants must determine how transactions and their data are represented on the blockchain, agree on the "if/when...then…" rules that govern those transactions, explore all possible exceptions, and define a framework for resolving disputes [4].

Then the smart contract can be programmed by a developer – although increasingly, organizations that use blockchain for business provide templates, web interfaces, and other online tools to simplify structuring smart contracts.

A smart contract is a sort of program that encodes business logic and operates on a dedicated virtual machine embedded in a blockchain or other distributed ledger [2].

Step 1: Business teams collaborate with developers to define their criteria for the smart contract's desired behavior in response to certain events or circumstances.

Step 2: Conditions such as payment authorization, shipment receipt, or a utility metre reading threshold are examples of simple events.

Step 3: More complex operations, such as determining the value of a derivative financial instrument, or automatically releasing an insurance payment, might be encoded using more sophisticated logic.

Step 4: The developers then use a smart contract writing platform to create and test the logic. After the application is written, it is sent to a separate team for security testing.

Step 5: An internal expert or a company that specialises in vetting smart contract security could be used.

Step 6: The contract is then deployed on an existing blockchain or other distributed ledger infrastructure once it has been authorized.

Step 7: The smart contract is configured to listen for event updates from an "oracle," which is effectively a cryptographically secure streaming data source, once it has been deployed.

Step 8: Once it obtains the necessary combination of events from one or more oracles, the smart contract executes.

C. *NFT minting flow*

Minting is the process of creating an NFT. You're essentially creating the smart contract code itself. The NFT's attributes are determined by the smart contract code, which then contributes them to the appropriate blockchain where the NFT is coded [5].
For smart contracts, several standards have been defined. Ethereum was one of the first cryptocurrencies to implement standards.

Following are the steps of creating an NFT Smart Contract:

Step 1: Prepare for the Project
To get started, we'll use the hardhat, which will give us a baseline for deploying the contract, performing tests, and so on.

Step 2: Installing the Hardhat
We'll use yarn to install hardhat as a development dependency. You'll also need to perform some commands over here.

Step 3: Making the Necessary Files and Directories

Let's begin by making a contract directory in the root folder. We can construct our first contract file inside the folder after we've created it.

Step 4: Developing the smart contract

We are now moving on to the most exciting portion of the article that is creating the contract. Let's begin by specifying the pragma version. The openzeppelin contracts module will be installed to facilitate this.
When your NFT production is offered on a marketplace, NFT royalties pay you a share of the purchase price. NFT royalty transactions are paid in perpetuity and are carried out automatically using smart contracts. Several marketplaces allow you to select your royalty percentage. A normal royalty is between 5% and 10%.

D. *Application flow*

This application uses a crypto wallet, so it is recommended to the user to have a metamask wallet with some Ethereum. Users will first land on the homepage where they can navigate throughout the application. Users can just explore the services if they have not logged in. To use all services, they must log in.

Step1: To transfer the Ethereum, users have to visit the transfer page where they input the receiver's address and amount to transfer. On submitting the form, users have to confirm the transaction in the metamask generated pop up. After confirmation the transaction will be initiated to the user's wallet.

Step2: To create NFT and collection, users have to navigate to the mint NFT and create collection where they have to fill in the details and after submitting a minting confirmation pop up. After confirmation, the NFT will be created on a decentralized server in some time. For the confirmation, users can visit the open sea NFT account.

Step3: To explore the market related services like crypto details, news and exchange values, users can navigate through the market sections where all the data is visible for real time.

Step4: To view the upcoming value of the crypto coin, users can visit prediction section

V. CONCLUSION

At the time of this writing, investing and trading in cryptocurrencies is constrained by lack of accessibility, infrastructure and education, and we want to change that by providing them all concisely. Our platform is very useful for future people. For this purpose, we will open this platform to the public. Our goal is to bring artificial intelligence and machine learning to the cryptocurrency market and build a complete system to help predict cryptocurrency prices. Use this web3 tech stack to build a platform that can provide users with accurate analysis and information about cryptocurrencies. This is achieved by implementing state-of-the-art artificial intelligence, machine learning, natural language processing, blockchain, and encryption, and Web3 technologies in the system. In addition, we will create an ecosystem where users can invest in cryptocurrencies simply by providing technical details about them. Here are just a few of the ways to optimize your platform for Cryptopunk investors: market data and analysis using blockchain, Web3, and coin ranking APIs.

Users can interact with our platform through virtual agents that provide accurate analysis and cryptocurrency pricing information in their native language. The community rating system used as a part of the platform ensures that only quality data is provided by these professionals. Users can also ask these experts directly about Cryptocurrency's credentials such as news, current status, values, community activities, and technical patterns

REFERENCES

[1]. Bader L, Bürger JC, Matzutt R, Wehrle K (2018) Smart contract-based car insurance policies. In: 2018 IEEE Globecom workshops (GC wkshps), pp 1–7
[2]. Bertini T, Butkute K, Canessa F Smart flight insurance—insureth. Available online at http://mkvd.s3.amazonaws.com/apps/InsurEth.pdf (2015)
[3]. Buterin V et al (2014) A next-generation smart contract and decentralised application platform. White paper

[4]. DHarz D, Knottenbelt W (2018) Towards safer smart contracts: A survey of languages and verification methods. arXiv preprint arXiv:1809.09805

[5]. Dryja T, Glasbergen G-J, Lovejoy J Layer 2 - the lightning network. Available online at https://dci.mit.edu/lightning-network/ (2019).

[6]. Feng X, Wang Q, Zhu X, Wen S (2019) Bug searching in smart contract. arXiv preprint arXiv:1905.00799

[7]. T. Banerjee, S. Sinha, and P. Choudhury, "Long term and short term forecasting of horticultural produce based on the LSTM network model," Applied Intelligence, Jan. 2022

[8]. Sherstinsky, A., 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, p.132306.

[9]. Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9.1735-80.10.1162/neco.1997.9.8.1735