SentiChain: Decentralized Sentiment Gathering

Edward Chen

rchen@yototec.com

Abstract

SentiChain is a decentralized network that combines blockchain technology with natural language processing (NLP) to collect, validate, and store sentiment data. This innovative approach ensures that sentiment data remains secure, tamper-proof, and compliant with privacy policies. The network addresses key challenges in the use of sentiment data, including **survivorship bias**—where sentiment data is lost or modified over time—**privacy concerns** related to storing personally identifiable information (PII), and the complexity of converting qualitative sentiment into numerical representations. As such, SentiChain offers a verifiable, consistent, and trustworthy source of sentiment data for applications requiring accurate sentiment analysis.

Introduction

In today's data-driven world, sentiment analysis has become an essential tool for making informed decisions across various sectors, particularly in finance, marketing, and public relations. By understanding public sentiment, organizations can anticipate market trends, tailor their communications, and respond effectively to consumer demands. This capability is especially valuable in the financial sector, where market participants rely on insights derived from a combination of economic and corporate sources to make decisions. Social media, with its rapid dissemination of formal and informal information, has emerged as an increasingly important source of sentiments.

However, extracting and analyzing sentiments poses significant challenges. **Survivorship bias** occurs when original posts are modified or deleted, leading to distorted historical data. **Privacy concerns** arise from restrictions data vendors impose to protect personally identifiable information (PII). Moreover, **data conversion** is a critical hurdle that requires consistent methods to transform qualitative text into quantifiable insights.

To address these challenges, we introduce SentiChain, a decentralized network that integrates blockchain technology with natural language processing (NLP). In

SentiChain, sentiment data is anonymized and vectorized, with both the processes and outcomes secured by cryptographic signatures. The processed sentiment data is then verified by the community and recorded on the blockchain. To promote fair usage and encourage positive contributions, the protocol issues tokens as incentives.

In the following sections, we will provide an overview of SentiChain's features, delve into its technical innovations, and present a practical use case.

An Overview of the Features

SentiChain is a decentralized network designed to store verifiable sentiment data on-chain. It operates across multiple community-owned nodes, allowing participants to submit, validate, and query sentiment data in a decentralized manner. When participants submit sentiment data, the protocol systematically anonymizes and authenticates the data (see **"Sentiment Authentication"**) and generates numerical representations of the data (see **"Ensemble Embedding"**). The protocol secures the sequence of processes and their outcomes through cryptographic signing (see **"Sequential Signing"**). The community then collaborates to verify these signatures and record the verified data on-chain using a consensus mechanism called **Proof of Verification** (see **"Proof of Verification"**).

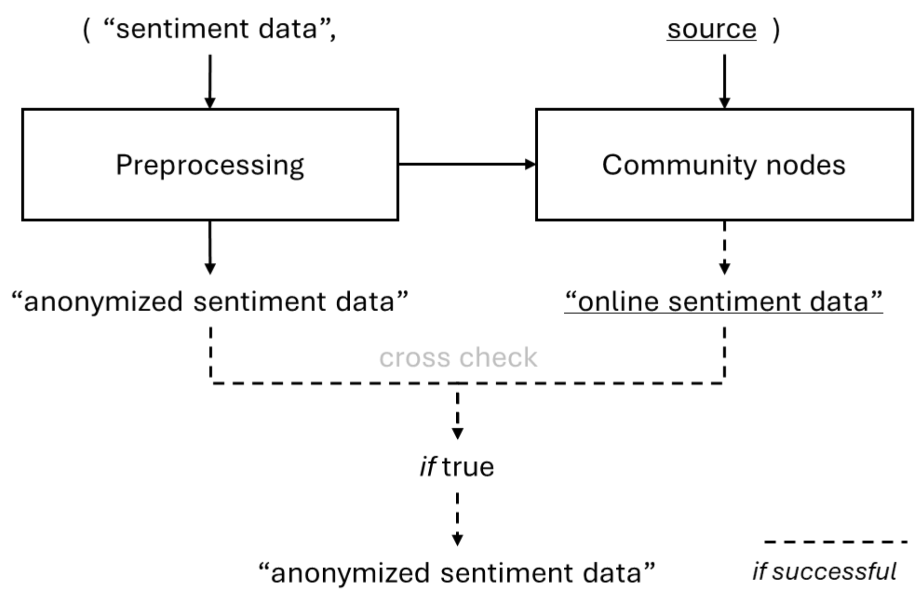
To promote fair network usage and incentivize positive contributions, the protocol issues tokens and will initiate a decentralized exchange (DEX) pool upon launch (see **"Tokenomics"**).

Sentiment Authentication

The lifecycle of decentralized sentiment gathering begins with the submission of sentiment data, which the protocol must authenticate before recording on the network. Ideally, this authentication relies on proofs generated by the data submitter, such as complete traffic flow data from a browser, webpage checksums from a trusted third party, or full HTTPS session captures. However, such proof can often be manipulated by the submitter, rendering them unsuitable as non-repudiable evidence. As a result, the protocol relies on a network of community-owned nodes to access the source of information and perform online authentication checks.

Before the protocol proceeds with authentication, it first removes any personally identifiable information (PII) to comply with general data privacy policies. This is accomplished using rule-based language processing that anonymizes the data as soon as it enters the protocol. The anonymized data is then cached in the protocol's memory. The protocol checks the submitted metadata to ensure it meets minimum requirements, such as the sentiment's timestamp being within the current block time, having attracted a significant number of views, containing sufficient extractable content, and originating from a publicly accessible source.

Once the sentiment data is anonymized and the metadata is verified, the protocol sends both the anonymized data and the verified metadata to the community for authentication. A node is randomly selected from the community to access the source of information and authenticate the data. Upon successful authentication, the node generates a cryptographic signature using the anonymized sentiment data (see **"Sequential Signing"**) and automatically forwards both the anonymized sentiment data and the signature to the protocol for further processing. The public key of each community node is pre-registered on the protocol for signature verification.



Anonymizing and authenticating sentiment data

Ensemble Embedding

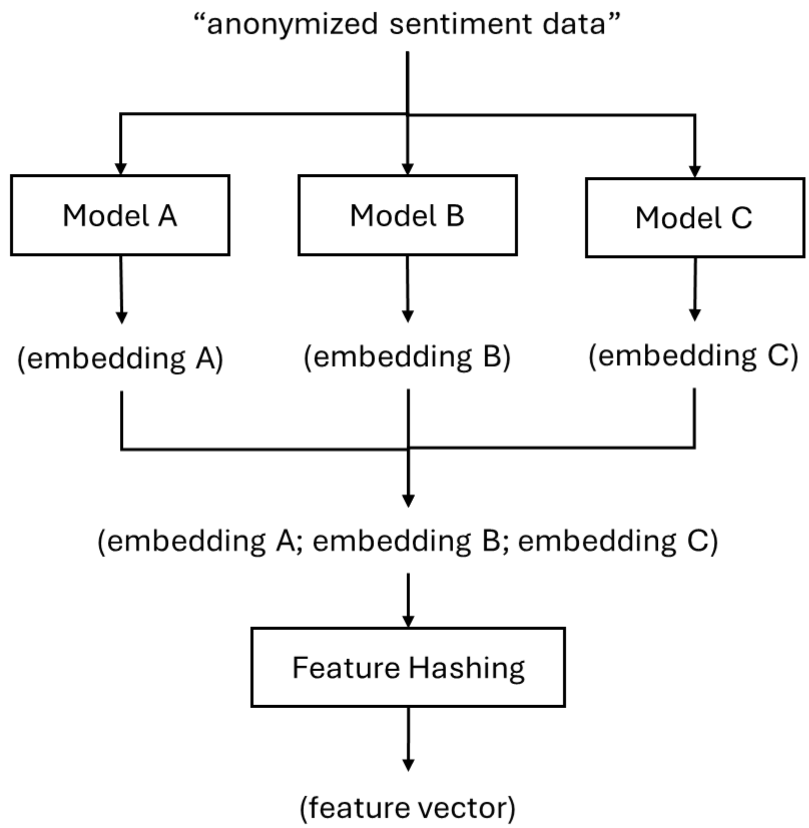
Upon receiving anonymized sentiment data, the protocol verifies its signature and uses an ensemble of language models to generate numerical representations of the data, followed by dimensionality reduction. This ensemble approach is chosen because it often provides richer and more robust representations compared to using a single model embedding, even when the vectors are of the same size. According to N. Harris (2024), adopting an ensemble approach with text enrichment before embedding with ChatGPT 3.5 improves cosine similarity measures by 8.21% over the baseline model on the TwitterSemEval dataset [1]. X. Li (2023) demonstrated that a hybrid model combining different types of embeddings enhances contextual understanding in content generation, as measured by Recall@3 and Mean Reciprocal Rank (MRR) [2].

The ensemble embedding process begins with the careful selection of language models that capture various aspects of contextual features and complement each other. In its initial iteration, the protocol combines embeddings from BERT, GPT-2, and the Universal Sentence Encoder (USE). BERT serves as the baseline model, generating bidirectional representations that delve deeply into semantic relationships within text [3]. GPT-2, a unidirectional model trained for next-word predictions, adds a layer by enhancing the understanding of informal and conversational language, complementing BERT in broader textual contexts [4]. The Universal Sentence Encoder (USE) combines transformers with a deep averaging network to focus on capturing the semantic meaning of entire sentences, complementing BERT and GPT-2 by embedding complex meanings holistically [5].

The protocol processes the anonymized sentiment data through these models and concatenates the resulting embeddings into a single high-dimensional vector. This vector then undergoes dimensionality reduction through feature hashing, a technique that applies a hash function to reduce dimensionality to a predefined limit while preserving the essential characteristics of the original embeddings. The number of features is chosen based on quantitative research to minimize the risk of hash collisions, which are inevitable in feature hashing but are typically acceptable because the collisions are uniformly distributed across the feature space. Supported by the Johnson-Lindenstrauss lemma for random projections, the feature vector should maintain the Euclidean distance in the low-dimensional space as the concatenated vector in high-dimensional space [6].

The final output is a compact feature vector that preserves the rich, extractable sentiment of the original anonymized data. The protocol generates a cryptographic signature for the feature vector (see **"Sequential Signing"**) and forwards both the feature vector and its signature to the mempool for further verification before being included in the next block.

Since this process is irreversible, the original content cannot be reconstructed from the feature vector, which addresses general privacy concerns associated with storing sentiment data on a public network. To assist users in training their own models for sentiment extraction from these feature vectors, the protocol will publish notebooks detailing the ensemble embedding process, including the exact versions of the models used.

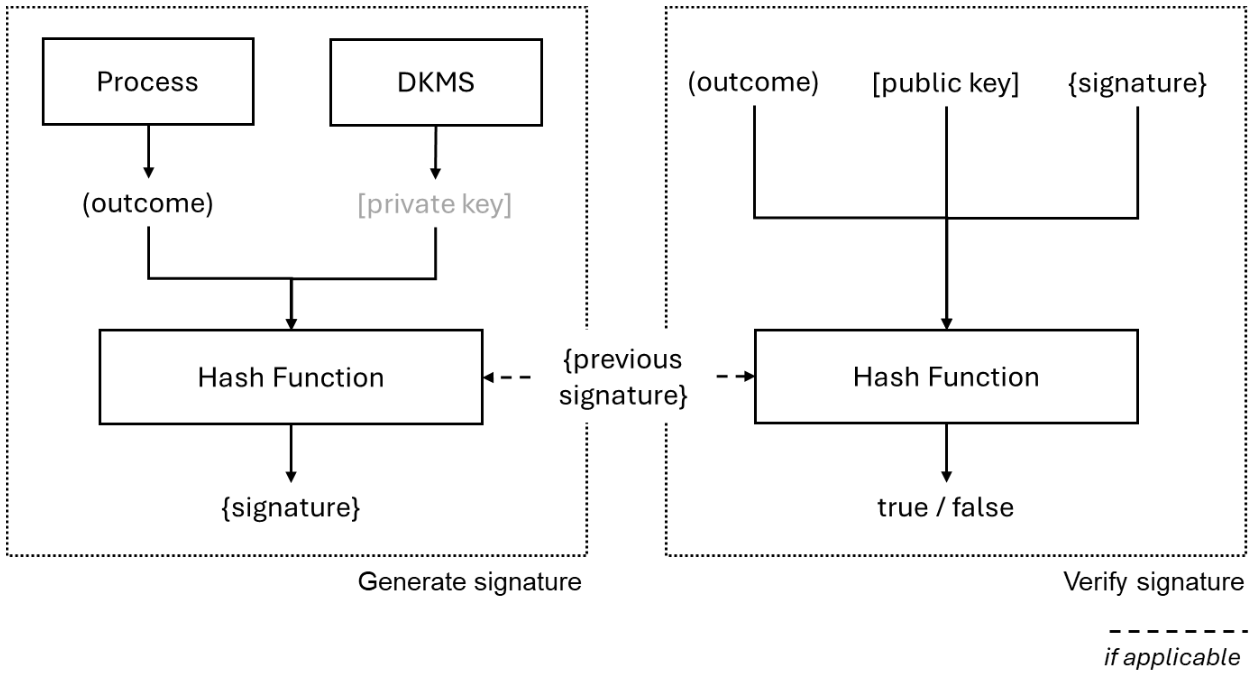


Generating feature vector from anonymized sentiment data

Sequential Signing

As mentioned in previous sections, the protocol generates cryptographic signatures to authenticate the outcomes of the anonymization, authentication, and embedding processes. The sequential order of these processes is secured by including the signature from the preceding step in the signing of the current step. The signing process uses **digital signatures**, which involve cryptographic hashing and encryption with a private key. This ensures that any alteration in the data, even at the bit level, results in a significantly different signature, thereby securing the integrity of the information.

The protocol uses a Decentralized Key Management System (DKMS) to manage its private key across community-owned nodes. This system utilizes **Shamir's Secret Sharing** [7] , which mathematically divides a private key into multiple shares. The key can only be reconstructed when a sufficient number of shares are combined, meaning that even if an attacker obtains some shares, they cannot reconstruct the key unless they have acquired the quorum number of shares. The corresponding public key is publicly available, allowing anyone to verify the outcomes of the processes and confirm that they have remained sequential and unaltered.



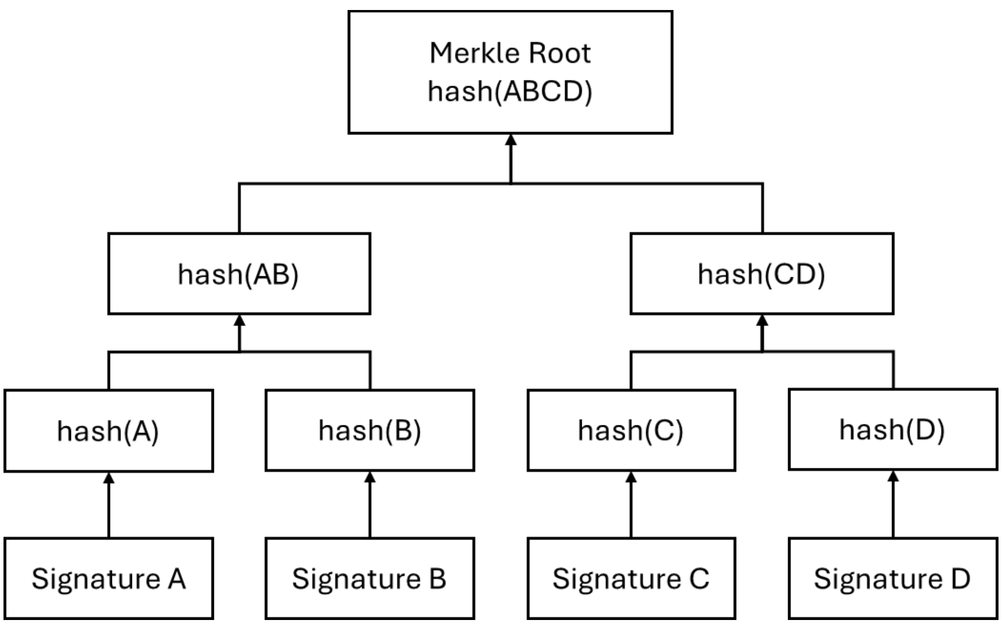
Generating and verifying signature

Proof of Verification

The Proof of Verification consensus mechanism utilizes cryptographic verification and Merkle Trees to ensure the integrity and authenticity of sentiment data before it is recorded on the blockchain. A Merkle Tree is a specialized data structure that uses cryptographic hashes to create a hierarchical arrangement of data blocks [8]. This structure enables efficient and secure verification of large datasets by producing a **Merkle Root**—a single hash representing all the data within the tree. The protocol uses the Merkle Root as a single representation of multiple sequential authentications for the community to vote on and reach consensus before the block is finalized.

To ensure this verification process is decentralized, the consensus mechanism is conducted by a group of participants known as **verifiers**. Any participant with a sufficient stake can apply to become a verifier and is incentivized to provide prompt and truthful verifications through staking rewards and penalties for malicious behavior. Verifiers are responsible for verifying entries in the mempool to ensure all feature vectors and signatures are authenticated. They are obliged to propose new blocks, participate in the consensus process by voting on proposed blocks, and contribute to the finalization of blocks.

The consensus process begins at the end of each block interval, where each verifier can propose a new block. To propose a block, verifiers must first verify the sequential signatures of feature vectors in the mempool, remove any duplicate entries, and reorganize the remaining data in strict chronological order. Once the data is verified and sorted, verifiers construct **balanced Merkle Trees** from these verified data points. The root hash of these trees, known as the Merkle Root, serves as a compact representation of all the entries [8]. Verifiers then submit their proposed Merkle Roots for voting. The Merkle Root that receives the most votes becomes the **Consensus Root** and is included in the block header.



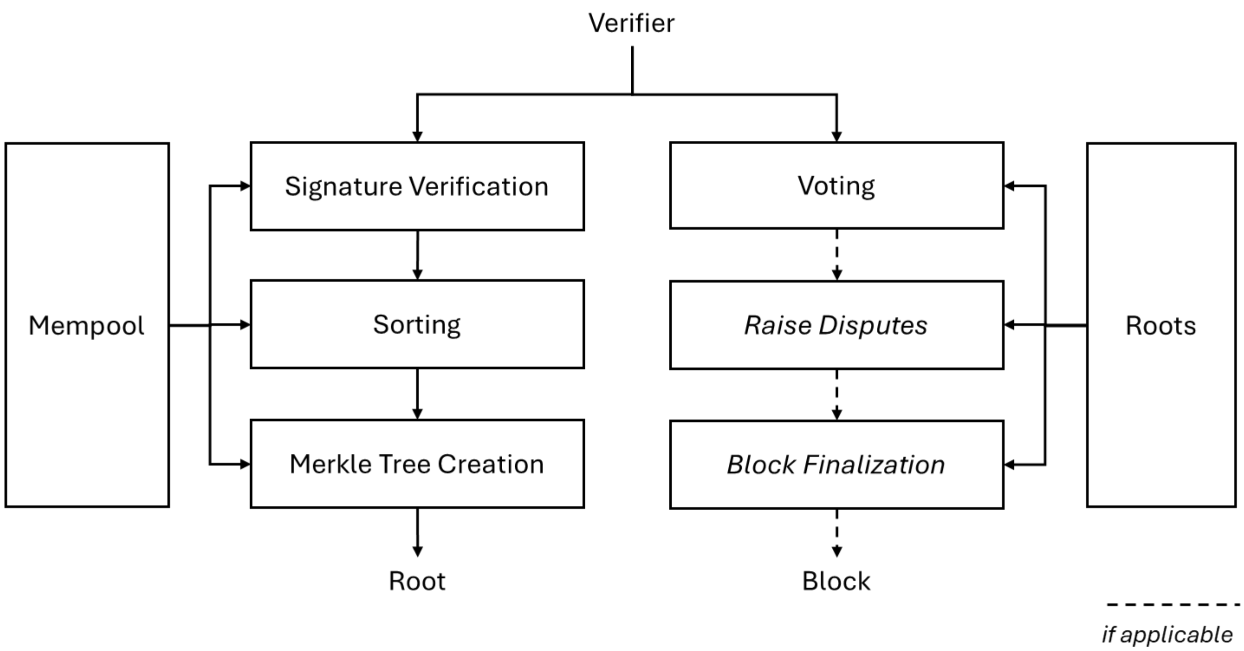
Generating Merkle Tree and Merkle Root

The first verifier who proposed a Merkle Root matching the Consensus Root becomes the **Lead Verifier**, responsible for submitting the proposed block—including the

Consensus Root—for finalization. However, before the Lead Verifier proceeds to block finalization, there is a short window for other verifiers to dispute the block proposal.

If a dispute arises, the protocol follows an open-source dispute resolution procedure to resolve it. This procedure involves the protocol proposing a new block that includes a standardized method for conducting the verification, constructing a new Merkle Tree, and comparing the new Merkle Root with the Consensus Root. If the new Merkle Root matches the Consensus Root, the dispute is considered unsuccessful, and the original Lead Verifier proceeds with block finalization. However, if there is a discrepancy between the two roots, the dispute is successful, and the verifier who raised the dispute becomes the new Lead Verifier, who will then use the revised block for finalization.

Once consensus is reached and any disputes are resolved, the Lead Verifier finalizes the block and adds it to the blockchain.

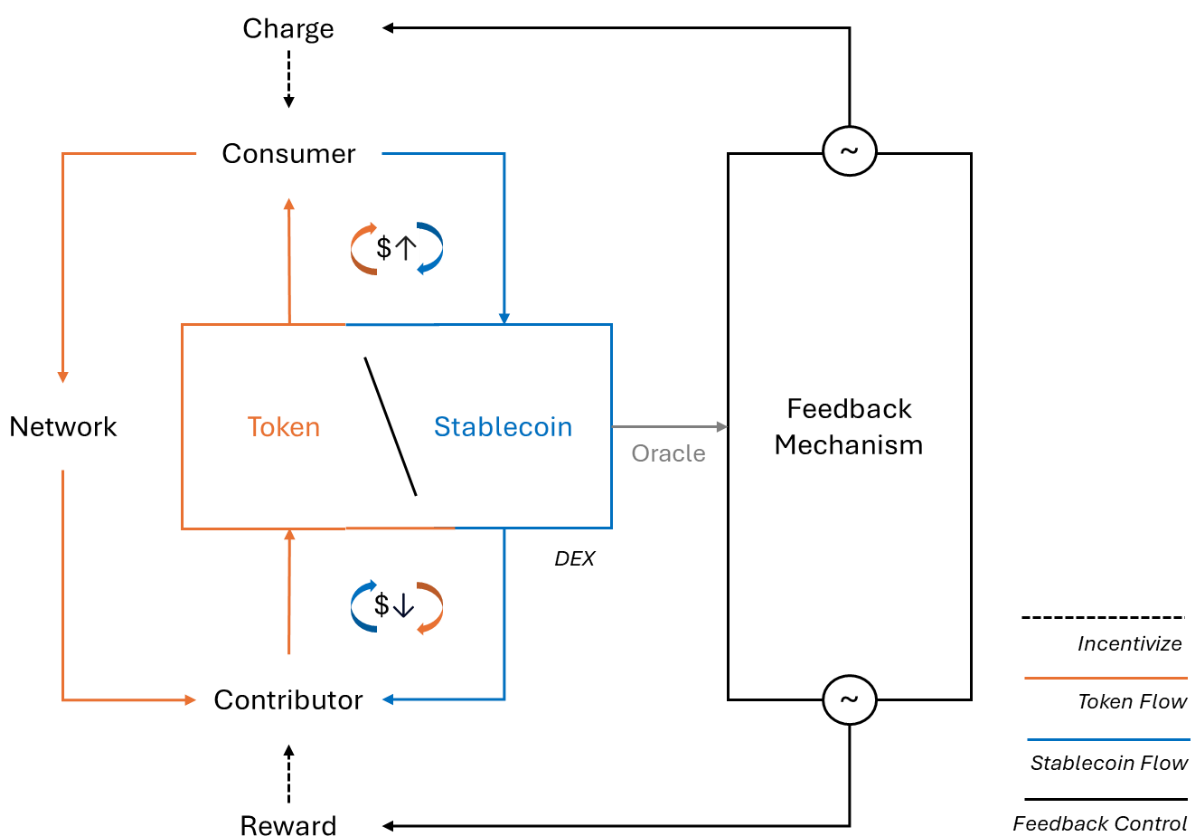


Proof of Verification

Tokenomics

The protocol issues a fixed supply of tokens to reward network participants who contribute sentiment data that is incorporated into the blockchain and to charge participants who query full block information. The number of tokens rewarded and charged per block is governed by a feedback mechanism, which automatically adjusts rewards and charges based on price fluctuations caused by unstable token circulation. The goal is to encourage fair network usage by maintaining stable token circulation.

For example, if the feedback mechanism detects a decline in token price due to excessive token circulation—likely resulting from a higher supply of sentiment data than demand—it will reduce the charges for data queries to boost demand and lower the rewards for sentiment data contributors to discourage supply. Conversely, if the feedback mechanism detects a sharp increase in token price due to a reduction in token circulation—likely resulting from higher demand for sentiment data than supply—it will increase the charges for data queries to control demand and increase the rewards for sentiment data contributors to encourage supply. These adjustments help reduce the circulating supply of tokens, thereby stabilizing or recovering the token price.



Tokenomics and the feedback mechanism

Under the current fee structure, anyone can freely access block headers, including the block hash and Consensus Root. However, access to the full block information is available only through the platform as a service. Participants who query the block information can regenerate the Merkle Root and compare it with the publicly available Consensus Root to verify authenticity.

Participants holding a sufficient number of tokens can apply to become **verifiers**, who have the potential to be selected as the **Lead Verifier**. Verifiers must meet specific criteria, such as holding a minimum token stake and possessing the necessary computational resources to process transactions and participate in the consensus process. The selection of the Lead Verifier is based on a combination of stake size and random selection to ensure fairness and security.

The Lead Verifier earns a substantial portion of the tokens accrued from block queries within a designated timeframe following block finalization. The exact timeframe for this reward distribution will be announced on the protocol's official channels prior to launch, keeping all participants informed.

To facilitate token exchange and price discovery, the protocol will initiate a decentralized exchange (DEX) pool and pair the token with a stablecoin upon launch. The initial fixed supply of tokens will be allocated through a combination of methods, including an initial token offering, allocations to early contributors, and reserves for future development and ecosystem growth.

Use Cases

A direct use case of SentiChain is in quantitative trading. Suppose a quantitative trading company wants to develop a sentiment-based strategy. The first step they need to take is to find a sentiment data vendor that provides data suitable for backtesting and generating signals in real time. In quantitative trading, it is crucial that the sentiment data remains consistent between historical and forward feeds, meaning the current state of sentiment data must match how it was at any point in the past. The data must also be verifiable to ensure its authenticity.

One might suggest that the company could directly source data from social media platforms, which could potentially solve both issues. However, this approach presents a significant challenge: mitigating **survivorship bias**, which occurs when original posts are modified or deleted. Social media platforms often cannot track or display these changes due to privacy concerns, as some posts may contain personally identifiable information (PII), making it impossible to show the modification history without violating privacy policies.

SentiChain effectively addresses both issues by integrating blockchain with natural language processing (NLP). All sentiment data on SentiChain is anonymized and represented as feature vectors, which are verifiable through cryptographic signatures. PII is removed during preprocessing, and the feature vectors are structured to prevent reverse engineering to recover the original content. Additionally, data integrity is maintained through the Proof of Verification consensus mechanism.

In quantitative trading, where the original content must be converted into numerical representations for analysis, querying feature vectors from SentiChain places users at no disadvantage compared to accessing raw text data. Moreover, SentiChain's architecture allows for updates to embedding techniques, such as incorporating the latest models like ChatGPT embeddings, in future iterations if they prove to be consistent and licensing permits.

As an example, on 27th of November 2020 (Black Friday), Citron Research, a prominent short seller, posted a bearish outlook on Palantir Technologies (PLTR), predicting that the stock would drop to $20 by the end of the year. If this post had been recorded on SentiChain's blockchain, users can apply sentiment analysis to the associated feature vector to extract sentiments such as *Bearish*, *Violative*, *Subjective*, *Volatile*, *Confident*, *Speculative*, *Upcoming*, and *Technical* (see **"Appendices [B]"**). Topic modeling could link this feature vector to PLTR. These insights could be used to initiate a speculative short on PLTR and prepare for potential take-profit or stop-loss strategies due to its volatile sentiment. Shortly after Citron Research's post, PLTR experienced heavy selling, causing the stock price to drop from $33.50 per share to $27.66 by the close of trading. PLTR continued to decline over the next five trading days before rebounding.

PLTR stock price movements (in UTC timestamp) on 27th of November 2020

As an active short seller, Citron Research has historically posted views on social media that have moved markets, including posts on GameStop (GME) in 2021, Inovio Pharmaceuticals (INO) in 2019, and Clean Energy Fuels Corp (CLNE) in 2017. However, Citron Research has since deleted all its old posts after dealing extensively with class-action lawsuits and internet trolls. If a quantitative trading company were to acquire data directly from social media platforms, they would not be able to access Citron Research's posts prior to 14th of March 2023, which played a crucial role in market movements at the time.

Citron Research deleted all posts prior to 14th of March 2023

Beyond event-driven trading, there is substantial empirical evidence supporting the broader impact of sentiment analysis on quantitative trading. For example, Tetlock (2007) demonstrated that negative sentiment in financial news could predict stock price declines, with sentiment-based models outperforming the market by enhancing annualized returns [9]. Bollen, Mao, and Zeng (2011) found that Twitter sentiment could predict stock market movements, with strategies incorporating this data achieving 15% higher returns compared to baseline models [10]. Heston and Sinha (2017) showed that using news sentiment could improve risk-adjusted returns by 5-10% [11].

Roadmap

As the initial phase of SentiChain's development, a prototype has been implemented and launched at sentichain.com. This prototype includes all the features detailed in the section **"An Overview of the Features "** and allows users to interact with the pre-launch phase archived Testnet. The archived Testnet contains 20,880 blocks of sentiment data gathered from 1st of October 2019, to 1st of February 2022. Through the platform's frontend, users can explore the blocks, generate feature vectors with signatures, and verify these signatures. Additionally, users can apply for API access to test the full lifecycle of the protocol, from submitting sentiment data to conducting Proof of Verification (see **"Appendices [A]"**).

The Testnet will be officially launched once the client, tools, and documentation are finalized and released, enabling SentiChain to be distributed and operated by a network of verifier nodes. To support the network's growth, SentiChain will host multiple rounds of incentive games, allowing potential participants to experience the chain, provide feedback, and develop tools. The Mainnet launch will follow the conclusion of these incentive games.

Future iterations of SentiChain will focus on further decentralizing the network by innovating a decentralized natural language processing pipeline (**dNLP**), which will replace the services currently provided by the community-owned nodes. This will enable the embedding tasks to be serialized and distributed among multiple nodes, with each node responsible for part of the process and generating a **Zero-Knowledge Proof** to ensure its integrity.

Looking ahead, we also plan to introduce **SentiMove**, a no-code sentiment analytics platform designed to help users extract actionable sentiments that could influence market movements using data from SentiChain. An experimental prototype of SentiMove has already been launched at sentimove.com, where users can paste feature vectors generated by SentiChain and extract sentiment insights (see **"Appendices [B]"**).

Conclusions

SentiChain represents a significant advancement in sentiment data gathering by providing a decentralized, secure, and privacy-conscious solution that effectively addresses challenges such as data integrity, privacy concerns, and the complexities of sentiment data conversion. By integrating blockchain technology with advanced natural language processing (NLP) techniques, SentiChain ensures the authenticity and immutability of sentiment data, making it a reliable resource for applications in quantitative trading, branding, public relations, and more. As the protocol continues to evolve, with plans for a decentralized natural language processing pipeline (dNLP), SentiChain is poised to become a computationally efficient decentralized platform that delivers real-time sentiment analysis to meet the ever-growing demands of a data-driven world.

References

[1] A. B. S. H. Nicholas Harris, "Enhancing Embedding Performance through Large Language Model-based Text Enrichment and Rewriting," arXiv, 2024.

[2] A. H. *. M. D. J. N. a. Y. W. Xiu Li, "Evaluating Embeddings from Pre-Trained Language Models and Knowledge Graphs for Educational Content Recommendation," Department of Computer and Systems Sciences, Stockholm University, Stockholm, 2023.

[3] M.-W. C. K. L. K. T. Jacob Devlin, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv, 2018.

[4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, "Language Models are Unsupervised Multitask Learners," OpenAI, 2019.

[5] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope and R. Kurzweil, "Universal Sentence Encoder," arXiv, 2018.

[6] A. G. F. K. M. C. Benyamin Ghojogh, "Johnson-Lindenstrauss Lemma, Linear and Nonlinear Random Projections, Random Fourier Features, and Random Kitchen Sinks: Tutorial and Survey," arXiv, 2021.

[7] A. Shamir, "How to Share a Secret," *Communications of the ACM,* vol. 22, no. 11, pp. 612-613, 1979.

[8] J. B. E. F. A. M. S. G. Arvind Narayanan, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton: Princeton University Press, 2016.

[9] P. C. Tetlock, "Giving Content to Investor Sentiment: The Role of Media in the Stock Market," *The Journal of Finance,* vol. LXII, no. 3, pp. 1139-1168, 2007.

[10] H. M. a. X. Z. Johan Bollen, "Twitter Mood Predicts the Stock Market," *Journal of Computational Science,* vol. 2, no. 1, pp. 1-8, 2011.

[11] S. L. H. a. N. R. Sinha, "News vs. Sentiment: Predicting Stock Returns from News Stories," *Financial Analysts Journal,* vol. 73, no. 3, pp. 67-83, 2017.

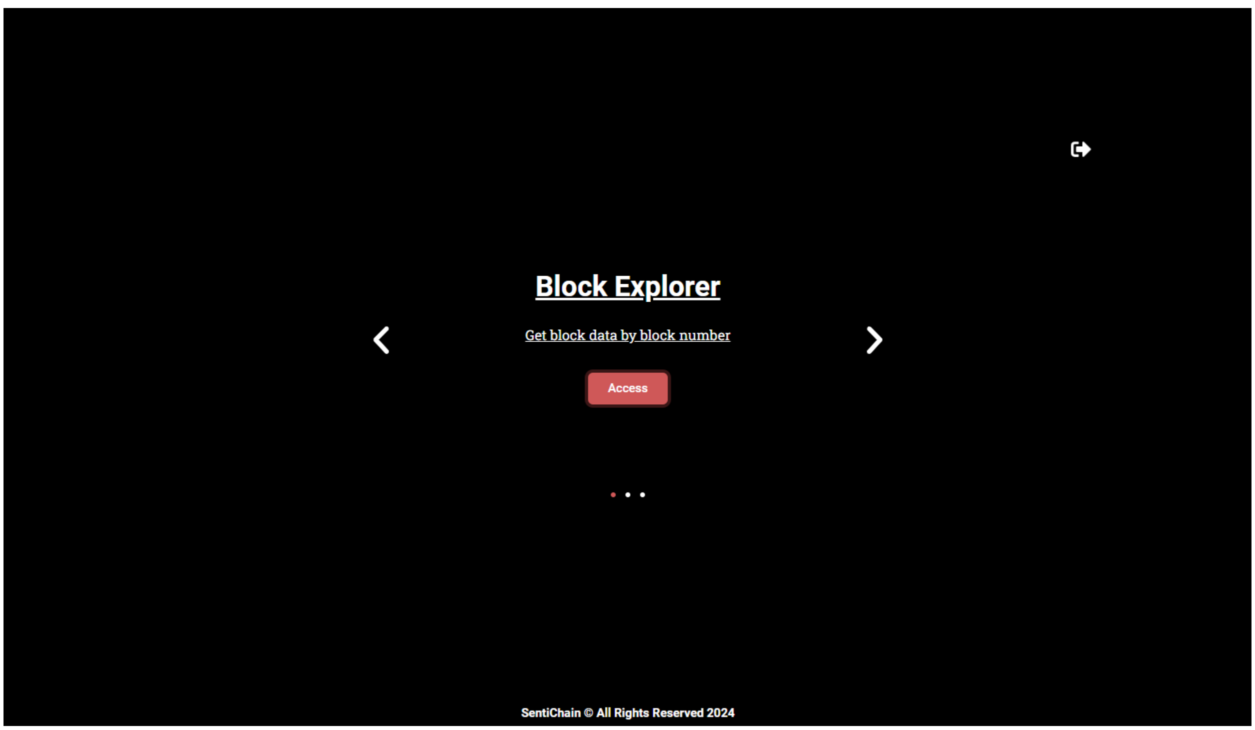
Appendices

[A] SentiChain Prototype

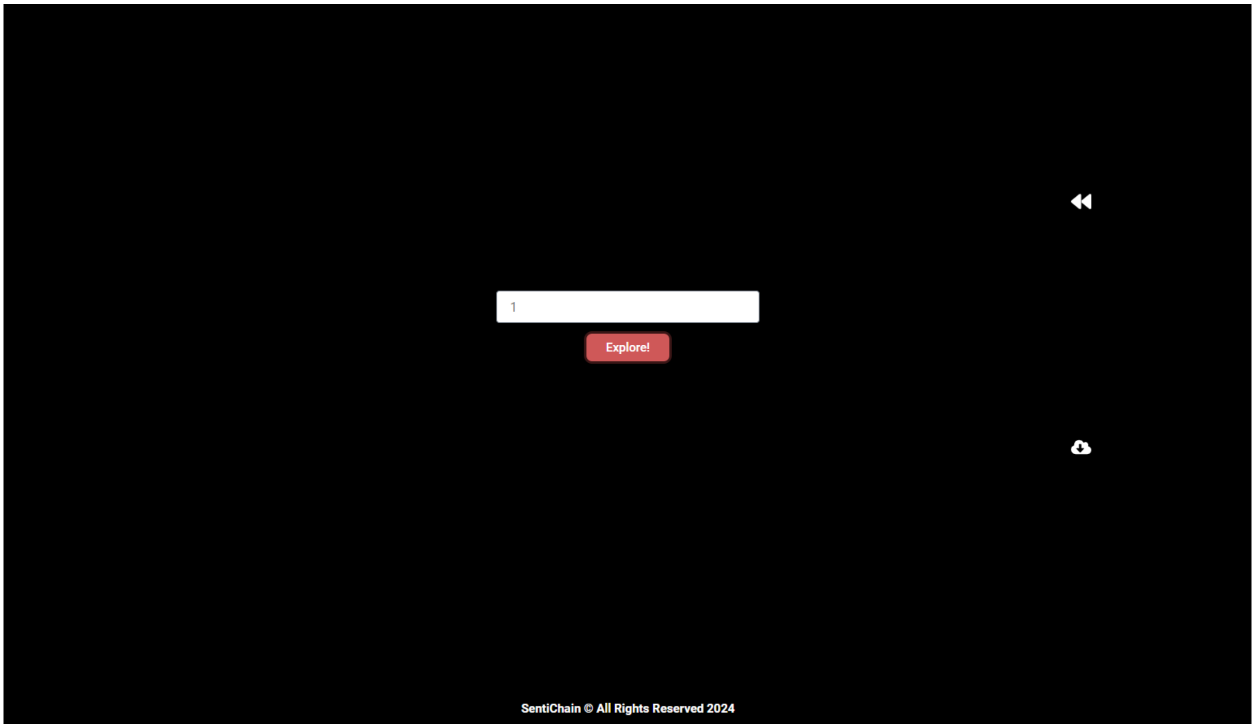
As the initial phase of SentiChain's development, a prototype has been implemented and launched at sentichain.com, where users can apply for an account to interact with its features detailed in the section **"An Overview of the Features"**. Currently, the prototype offers three main functionalities: **"Block Explorer"**, which allows users to explore the archived Testnet; **"Vectorize and Sign"**, which enables users to convert any text into a feature vector and create a cryptographic signature; and **"Verify Signature"**, which allows users to verify the authenticity of a feature vector using its signature. Examples with screenshots for each feature are provided below:

1. Block Explorer

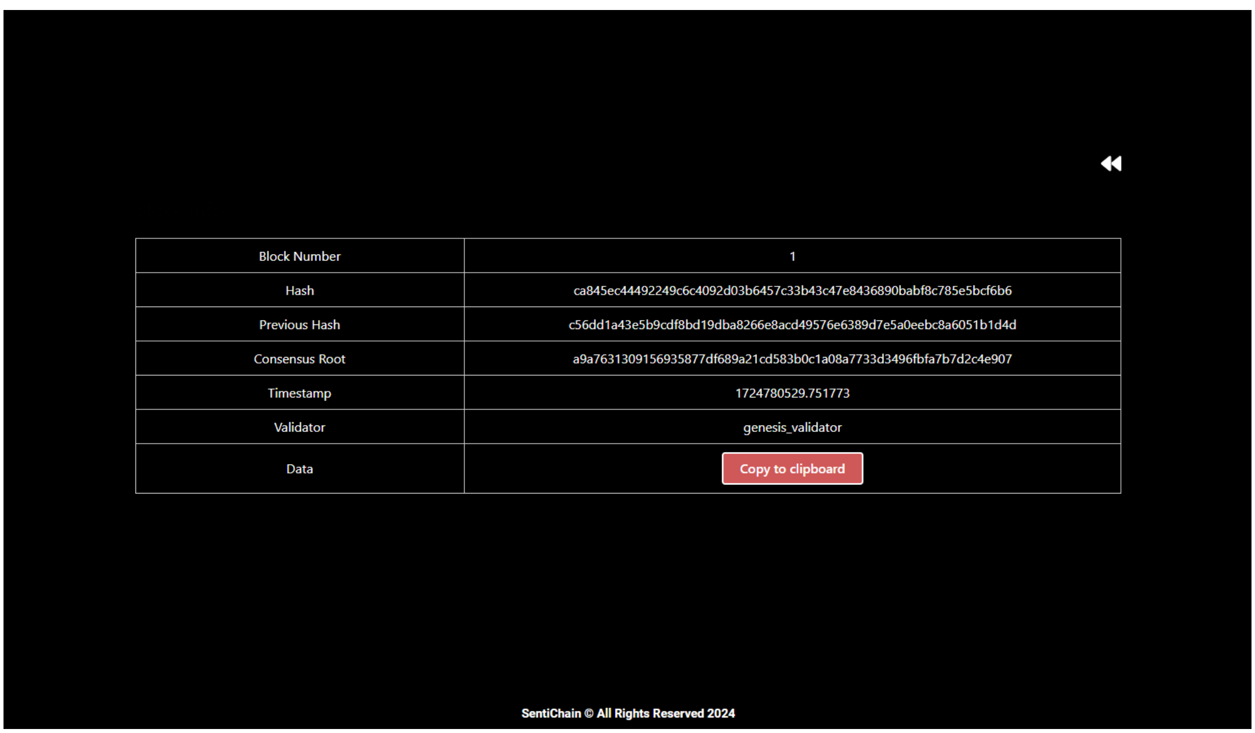
The **"Block Explorer"** feature allows users to explore all 20,880 archived blocks:



In this example, we explore the first block after genesis:

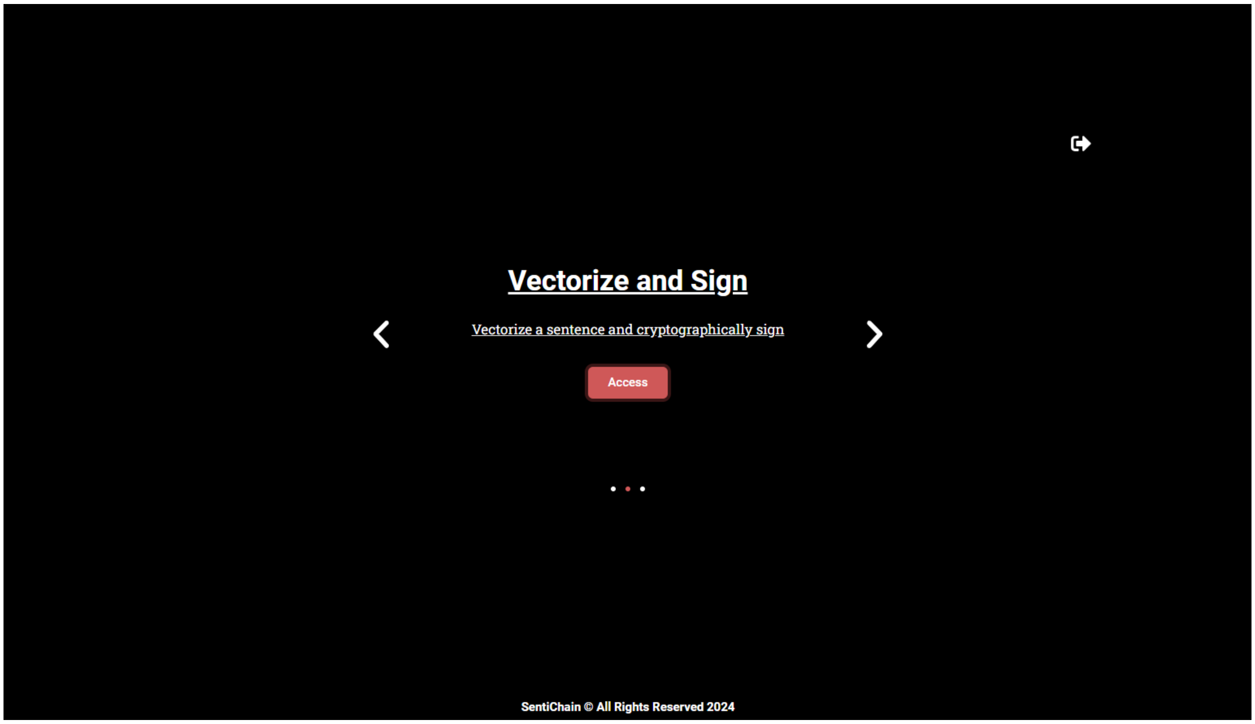


Part of the block header, including the Consensus Root, is displayed on the frontend. More detailed block information can be copied directly to the clipboard:

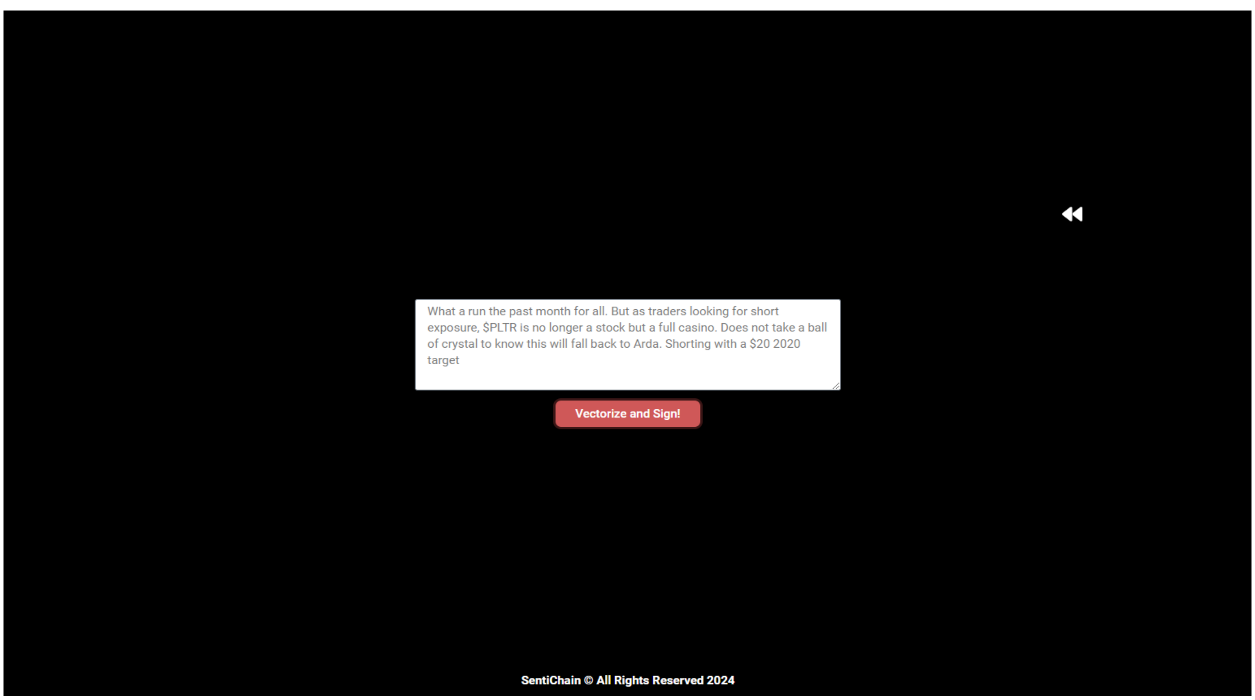


2. Vectorize and Sign

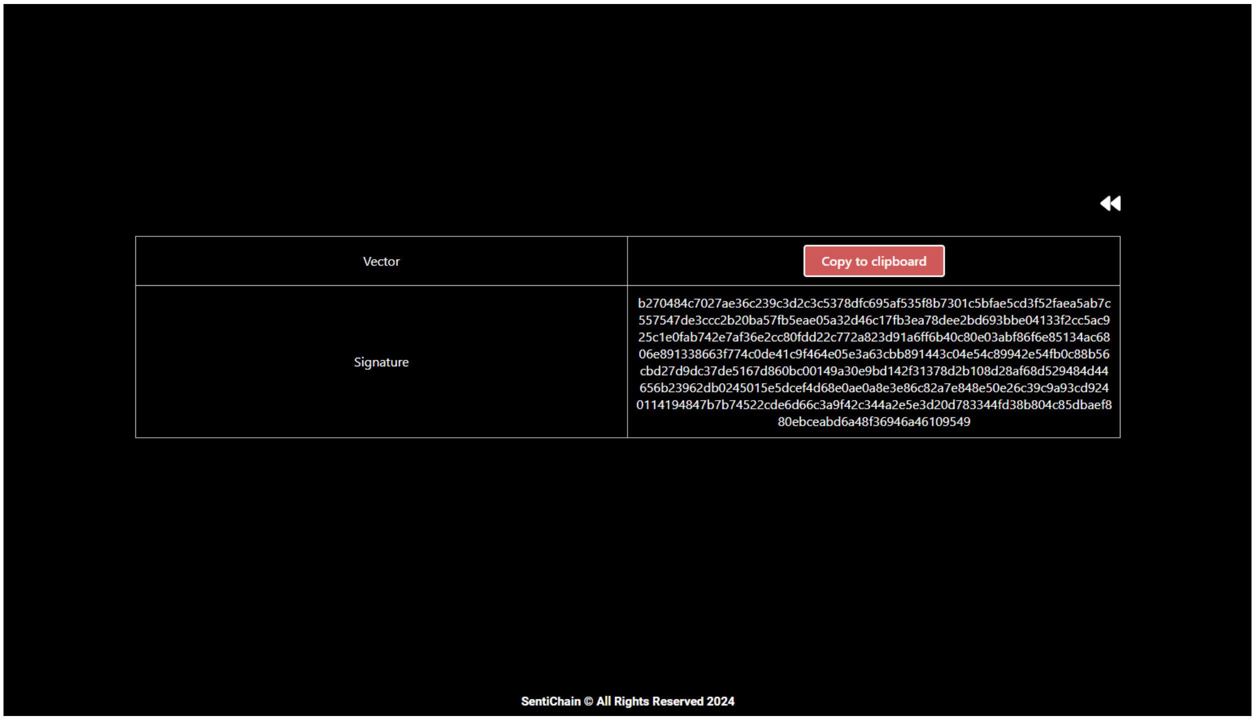
The **"Vectorize and Sign"** feature enables users to convert any textual data into a feature vector and generate a cryptographic signature:



In this example, we input Citron Research's post on 27th of November 2020:

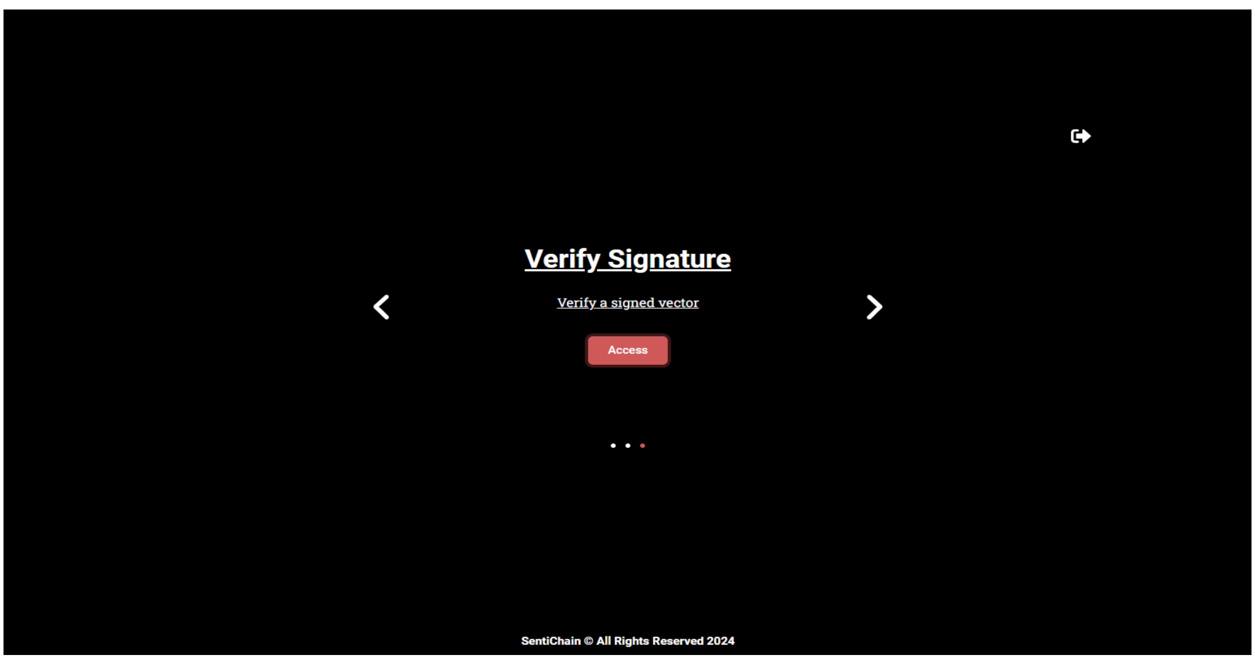


The feature vector can be copied to the clipboard, and the cryptographic signature is displayed on the frontend:

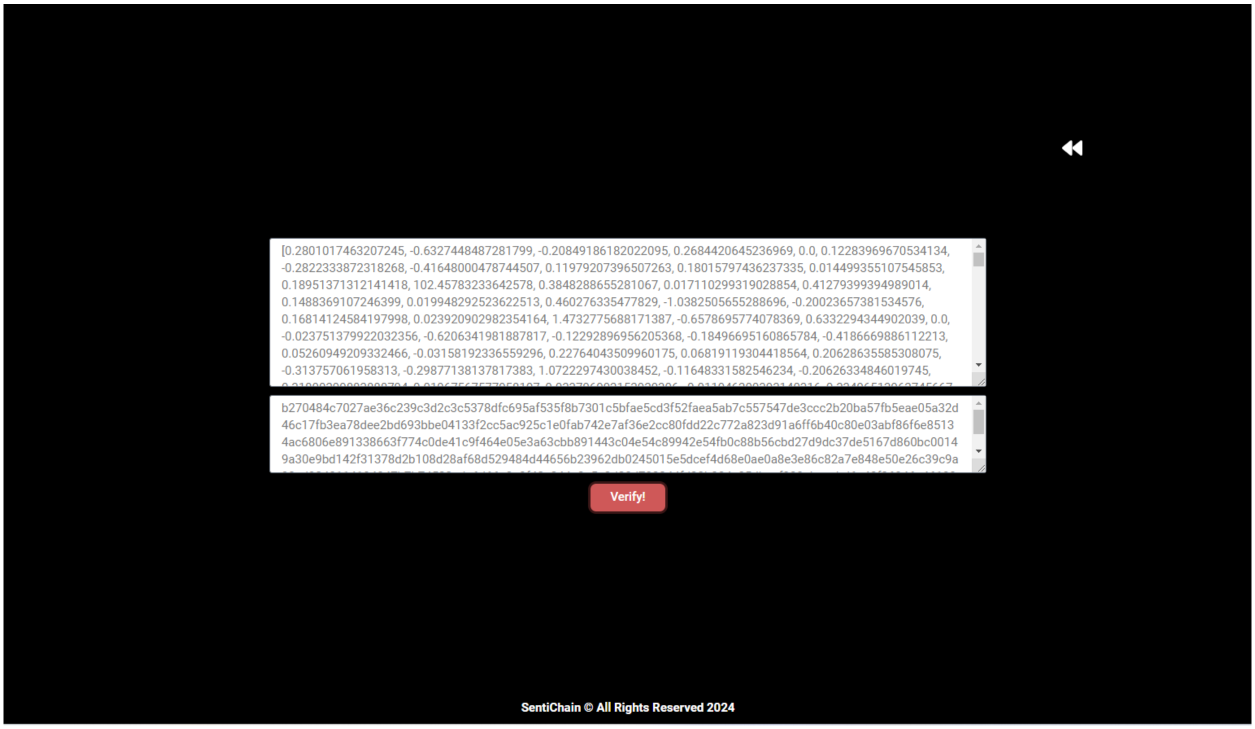


3. **Verify Signature**

The **"Verify Signature"** feature allows users to verify a SentiChain-generated feature vector using its signature:



In this example, we paste the feature vector and signature from Citron Research's post:



If the signature is successfully verified, **"Yes"** is displayed on the screen:

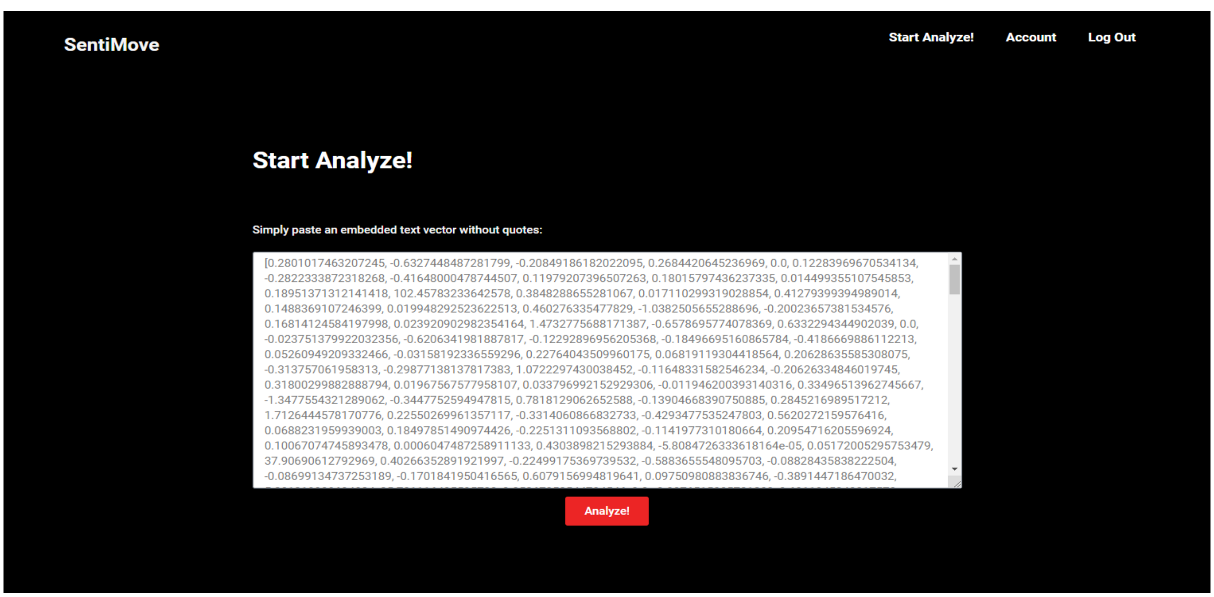
[B] SentiMove Prototype

As a no-code sentiment analytics platform, the prototype of **SentiMove** has been implemented and launched at sentimove.com, where users can apply for an account to use its features. Currently, users can use its **"Start Analyze!"** feature to extract sentiments from a SentiChain-generated feature vector across multiple dimensions, including:

- Bullish vs. Bearish
- Compliant vs. Violative
- Objective vs. Subjective
- Stable vs. Volatile
- Confident vs. Cautious
- Informed vs. Speculative
- Upcoming vs. Past
- Fundamental vs. Technical

The displayed results are final votes derived from an ensemble method that combines the predictions of multiple classifiers using a voting scheme. Please note that these predictions are sensitive to the training samples used. In this prototype, the training process utilized a relatively small sample generated by Large Language Models (LLMs), which may affect the accuracy when extracting sentiments from feature vectors generated from non-financial topic sentences.

In this example, we input a feature vector generated from Citron Research's post:



This feature vector is predicted to have the following sentiments: *Bearish*, *Violative*, *Subjective*, *Volatile*, *Confident*, *Speculative*, *Upcoming*, and *Technical*, which align with the original content:

Topic	Score
Bullish (100) vs Bearish (0)	30
Compliant (100) vs Violative (0)	22
Objective (100) vs Subjective (0)	13
Stable (100) vs Volatile (0)	12
Confident (100) vs Cautious (0)	70
Informed (100) vs Speculative (0)	21
Upcoming (100) vs Past (0)	66
Fundamental (100) vs Technical (0)	7