# String Matching Algorithms for Cryptocurrency News Sentiment Analysis and Price Prediction

Makalah IF2211 Strategi Algoritma, Semester II Tahun 2024/2025


Andrew Tedjapratama - 13523148
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: andrewtedj@gmail.com, 13523148@std.stei.itb.ac.id



***Abstract*—Volatility in the cryptocurrency market is heavily influenced by public sentiment as it is spread in the news. Therefore, the rapid growth of the market has increased the demand for intelligent systems that can analyze news sentiment in real time to support investment decisions on the market. This paper proposes a string-matching based approach for efficient sentiment analysis of the latest crypto news articles. On this paper, efficient algorithms are implemented such as Aho-Corasick, fuzzy matching (Levenshtein distance), and regular expressions (regex) to analyze and detect the presence of positive, neutral, and negative sentiment keywords from live news obtained via the CryptoPanic API. The sentiment scores are computed based on the balance of detected keywords, and the results are visualized in an interactive bubble chart. The visualization effectively illustrates the prevailing sentiment for various coins and shows a demonstrable correlation between sentiment scores and potential price movements. This paper highlights the practical implications of string matching and regex for efficient techniques that can provide lightweight, and interpretable sentiment insights, while also serving as a foundation for real-time financial analytics in the crypto market.**



***Keywords—cryptocurrency; news sentiment analysis; string matching; Aho-Corasick; price prediction***


## I. INTRODUCTION

### *A. Background*

Cryptocurrency and blockchain technology have been widely considered to have a significant potential to reshape various sectors in the future, especially in finance. With rapid growth of the new technology, the modern financial space in the crypto market is characterized by rapid growth and extreme volatility. Cryptocurrencies could also provide access to financial services for those currently excluded, and has a decentralized nature so trust can be fostered better among users. Other than that, global accessibility is also the main prevailing superiority of blockchain and cryptocurrencies compared to traditional market and technologies, allowing faster facilitation and cheaper cross-border transactions.

This means that everyone in the world can access and have financial transactions within the crypto domain through decentralized technology, supported by super security with smart contracts. Using smart contracts, transactions are also automated, transparent, immutable, ensuring consistent and fair outcomes.

The price movements in the cryptocurrency market are primarily driven by supply and demand dynamics, similar to the stock market or other financial markets. However, the unique characteristic of cryptocurrencies, including their limited supply (especially for coins like Bitcoin, with maximum cap of 21 million coins), high volatility, growing adoption, and, most notably, their susceptibility to public sentiment, which is largely shaped by the flow of digital news. For instance, a statement from the U.S. Federal Reserve during a FOMC meeting can and has historically trigger a sharp decline in both Bitcoin's or even other altcoin's price, as tighter monetary policy reduces investor risk appetite. Similarly, a geopolitical tension such as the Israel-Iran conflict have led to market-wide fear around the world, causing uncertainty and resulting in sudden *sell-offs* (a large-scale withdrawal causing significant drops in the market).

On the other hand, news like President Donald J. Trump's public endorsement of building national crypto reserves in a tweet on X caused a surge in investor optimism and speculative buying, causing a pump or high value increase in some coins, reflecting how influential figures can sway market sentiment and cause unpredictable market swings. Psychological phenomena such as FOMO (Fear of Missing Out) and FUD (Fear, Uncertainty, and Doubt) can trigger drastic price changes in a short period, driven by the latest news. Therefore, news are also one of the most important aspect that cannot be disregard in cryptocurrency trading and investing. For investors and traders, the primary challenge lies in the overwhelming volume of information.

Manually processing hundreds of news articles daily to analyze market sentiment is an inefficient task. Forgetting or delays in responding to crucial information can lead to missed opportunities or financial losses. Consequently, there is a pressing need for an automated system that can efficiently extract, analyze, and present sentiment from crypto news to support faster, data-driven decision making. To explore the role of sentiment in crypto markets, this study develops a full-stack web application called ChainPulse, which performs real-time sentiment analysis on cryptocurrency news using efficient string matching algorithms.

*B. Problem Foundation*

Based on the background described, this paper formulates the following key problems:

1. How can real-time cryptocurrency news can be efficiently extracted with automation, and can analyze the news within?

2. How can string matching algorithms, specifically Aho-Corasick and Levenshtein Distance, be effectively implemented in calculating the sentiment scores from unstructured news text data?

3. How can the results of the sentiment analysis be presented and visualized in a user interface, to be easily understood and provide important insights for users?

4. How does the insights of the sentiment scores correlate to the price of cryptocurrencies and be used for further predictive models and innovation?

*C. Research Objectives*

This research aims to address the previous formulated problems by achieving several main objectives:

1. To implement the Aho-Corasick algorithm for efficient keyword matching, supported with the Levenshtein Distance algorithm and regular expressions (*regex*) to handle text variations within a sentiment analysis system.

2. To build a full-stack prototype application named ChainPulse to demonstrate the visual functionality and the feasibility of the designed system.

3. To analyze the sentiment results generated by the algorithms and systems, and explore their potential correlation with market price movements as a case study.

*D. Scope and Focus*

To ensure the research remain focused and in-depth on the main problems, the following limitations are established:

1. The news data source is limited to English-language articles provided by the public API of CryptoPanic, with the sentiment keywords also being in english.

2. The sentiment analysis is lexicon-based, meaning it relies on keyword matching against a predefined sentiment dictionary, and does not involve deep contextual understanding as performed by complex *Natural Language Processing* (NLP) models.

3. This study does not aim to build a statistically validated price prediction model but rather to demonstrate the potential correlation and relationship between these two variables using string matching algorithms.

## II. THEORETICAL FRAMEWORK

To build the ChainPulse system, a combination of concepts and algorithms from computer science and finance is utilized. This chapter outlines the theoretical foundations that underline the system's architecture and functionality, including sentiment analysis, the string matching algorithms, and the data visualization techniques used.

*A. Sentiment Analysis*

Sentiment Analysis, or could also be called “opinion mining”, is a field of study that analyzes people's opinions, sentiments, evaluations, attitudes, and emotions from written language. In the financial domain, sentiment analysis is a critical tool for analyzing market mood, which can be a significant predictor of price movements. Investor sentiment, as reflected in news articles, social media, and forums, often precedes or coincides with major price swings.

There are primarily two approaches to sentiment analysis:

1. Machine Learning-Based Approach

   This method involves training a model (e.g., Naive Bayes, Support Vector Machines, or deep learning model) on a large, pre-labeled dataset of texts. While powerful in understanding context and sarcasm, this approach requires a lot of computational resources and a high-quality training dataset, which can be time-consuming to create.

2. Lexicon-Based Approach

   This method relies on a predefined dictionary (lexicons) of words, where each word is assigned a sentiment score (e.g., "bullish" = positive, "crash" = negative). The overall sentiment of a text is calculated based on the number and weight of the positive and negative words found. This research use the lexicon-based approach due to its high efficiency, transparency, and independence from large training datasets, making it highly suitable for real-time applications.

```
keywords = {
    "positive":
    [
        "bullish", "pump", "surge", "rally", "moon", "breakout", "ath", "all time high",
        "whale pumped", "support held", "uptrend", "rebound", "bounce", "recovery",
        "accumulation", "god candle", "parabolic", "demand zone", "double bottom",
        "consolidation", "higher low", "partnership", "listing", "airdrop", "upgrade",
        "adoption", "integration", "favor", "approval", "gains", "sparks", "increase",
        "extending", "inflow", "boost", "anticipated", "bull run", "green candle",
        "break resistance", "trend reversal", "ETF approved", "record high", "buy
        pressure", "new listing",
    ],

    "negative":
    [
        "crash", "dump", "bearish", "rugpull", "decline", "bear market", "whale dumped",
        "resistance failed", "exit liquidity", "bleed", "correction", "plunge",
        "pampedak", "downtrend", "down", "fud", "doubt", "fear", "uncertainty", "war",
        "loss", "bleeds", "attack", "hacked", "fake", "decrease", "scam", "pressure",
        "liquidated", "market turmoil", "flash crash", "exploit", "lawsuit", "under
        investigation", "sell pressure",
    ]
}
```

Fig 2.1 Lexicon dictionary for positive and negative sentiment keywords used in this project

### B. *Aho-Corasick Algorithm*

The Aho-Corasick algorithm is a highly efficient string searching algorithm that can find all occurrences of a large set of keywords within a text in a single pass. Developed by Alfred V. Aho and Margaret J. Corasick, the algorithm constructs a finite automaton, specifically a trie (prefix tree), with additional failure links if the patterns or string doesn't match. The process involves two stages:

1. Trie Construction

   Trie is a tree that is used to efficiently store and search through strings or patterns. All keywords from the sentiment lexicon are inserted into a trie. Each node represents a prefix, and a path from the root to a node represents a potential keyword.

2. Failure Link Creation

   Each node is equipped with a "failure link" that points to the longest proper suffix of the current string that is also a prefix of another keyword in the trie. These links allow the automaton to avoid backtracking. When a mismatch occurs, instead of restarting the search, the machine follows a failure link to an alternative state, preserving the information from the characters already scanned.

The primary advantage of Aho-Corasick is its linear time complexity, O(n + m + z), where 'm' is the total length of all patterns, 'n' is the length of the text, and 'z' is the number of occurrences of the patterns in the text. This efficiency is crucial for this paper, as it needs to scan news articles against a large dictionary of hundreds of sentiment keywords without significant delay.

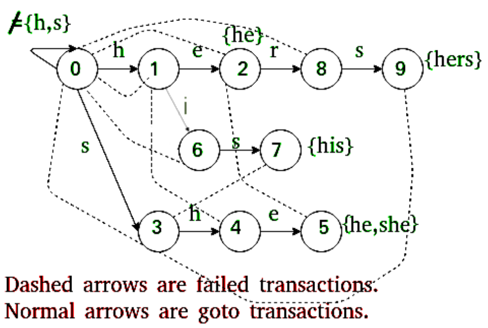


Fig 2.2 Aho-Corasick algorithm illustration
(Source: [1])

### C. *Levenshtein Distance (Edit Distance)*

While Aho-Corasick excels at finding exact matches, it is not robust against typo errors or word variations (for instance: Crrypto v.s. Crypto or B1tcoin v.s. Bitcoin). To address this, Levenshtein distance is utilized as a fuzzy matching mechanism. Levenshtein distance measures the "edit distance" between two strings, defined as the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word into the other.

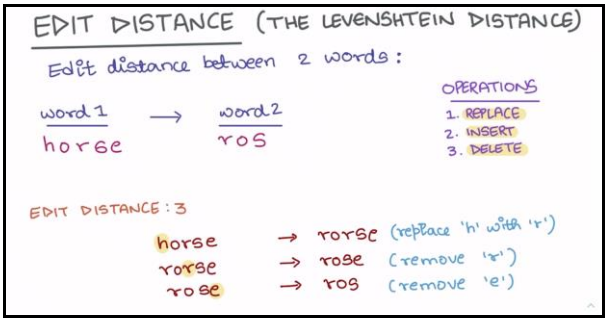


Fig 2.3 Levenshtein Distance or Edit Distance process illustration, with the operations used (replace, insert, delete)
(Source: [2])

For example, the Levenshtein distance between "rallly" and "rally" is 1. By converting this distance into a similarity score (e.g., 1 - (distance / max_length)), the system can identify words that are highly similar to a keyword even if they are not identical. In this research, this algorithm serves as a fallback, providing robustness and ensuring that sentiment is captured even from texts with minor errors, which are common in fast-paced news reporting or social media quotes.

### D. *Regular Expressions (Regex)*

Regex, or regular expressions, is a method to find patterns in text. With regex, it is possible to search for specific patterns, like word formats, numbers, symbols, or even variations of words that are not exactly the same but still mean the same thing. In this project, regex is used to detect keywords that may not appear in the positive/negative keyword list directly, but still follow a certain pattern.

Regex is useful because it is flexible. For example, the keyword “all-time high” can appear with spaces, with dashes, or written together, even with the abbrevation like “ATH”. Regex helps  detect all those forms without needing to write each one manually. This makes the matching process more efficient, and smarter when handling text variation.

Regex can also detect patterns that express emotions or emphasis, like repeated capital letters (“PUMP”, “DUMP”), exclamation marks (“!!!”), or emojis, which are often found in crypto-related news or social media posts. Since crypto news is not always formal and often comes with casual or emotional expressions, regex helps the system adapt to that kind of text.

### E. *Crypto News and Market Psychology*

The cryptocurrency market moves very fast and is very volatile. It is a market that can change prices in minutes, even seconds. One of the biggest reasons why the market moves is news. News can cause panic. News can create excitement or over optimism. News can push people to buy or sell coins quickly. That is why news is important. News shapes how people feel about the market. And how people feel affects prices.

In traditional finance, investors use news to make decisions. But in cryptocurrency, this effect is even stronger. This is because crypto is open 24/7, has less regulation, and relies heavily on public attention. News can spread fast on social media, forums, and news sites. That is why sentiment from news becomes one of the main signals for price movements. For example, if a big crypto exchange is hacked, negative news can quickly cause prices to fall. On the other hand, if a country like the United States approves a new Bitcoin ETF, that positive news can push the price of Bitcoin to go up. Even tweets from public figures, like Elon Musk or Donald Trump, can change the direction of the market.

This is why sentiment analysis is very useful. By checking if news is mostly positive or mostly negative, it is possible to predict how the market might react. If most news is positive, people may be more likely to buy, making the price of a coin grow. If most news is negative, people may be more likely to sell, making a price of a coin fall. So, by measuring sentiment, it is possible to better understand market psychology. This psychology is what drives buying and selling decisions, and those decisions affect the price.

However, Seeing the news everytime in multiple crypto article sites one by one and analyzing a sentiment can be time consuming and tiring. With this innovation, it is expected for those continuous processes to be handled real-time and automated through sentiment calculations and analysis with string matching algorithms.

In this project, the news is collected automatically through an API from a source called CryptoPanic, which collects and aggregates the latest news related to cryptocurrencies. Each news article is then analyzed to detect whether it is positive, negative, or neutral. This helps turn news into measurable numbers. With these numbers, the system can track overall mood or trend in the market.

## III. METODOLOGY

This chapter details the methodology used to design and build the ChainPulse application, from its system architecture to the specific implementation of its core algorithms. ChainPulse is developed as a full-stack application to provide a complete, demonstrable proof of concept. The demo of this project can be accesed through the youtube link on this paper, and the source code can be accesed on https://github.com/andrewtedja/chain-pulse.

### *A. System Architecture*

ChainPulse is designed with a modern, decoupled architecture, separating the backend logic from the frontend presentation layer. This separation enhances modularity, scalability, and maintainability.

For the backend, Python and FastAPI framework is used, responsible for making most of the core logic and for all heavy lifting including fetching data from the external CryptoPanic API, identifying relevant coins in articles, performing sentiment analysis, calculating sentiment correlation with price, and storing the results in a database.

The frontend of this app is built with Next.js (a React framework), Typescript, TailwindCSS, and d3.js (a library for bubble chart visualization). The frontend's role is to request data from the backend's API endpoints, manage the user interface, and render the interactive data visualizations.

For the Database, SQLite is used during development for its simplicity, with the system designed to be compatible with PostgreSQL for a more robust production environment.

The general data flow of the system can be visualized as follows:

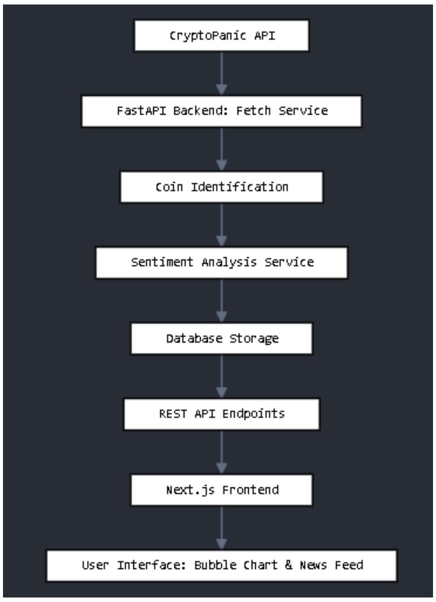


Fig 3.1 Flowchart visualization of general data flow in ChainPulse's system

### *B. Data Fetching and Processing*

The data pipeline begins with fetching news articles from the CryptoPanic API. This process, managed by the services.py module on the backend, is triggered either upon application startup or through a manual user request to refresh the data. The main processes done for fetching:

1. API Call
   A GET request is sent to the CryptoPanic endpoint, retrieving the latest news articles in JSON format.

```python
# ================= Fetch API =================
def fetch_crypto_news():
    try:
        response = requests.get(BASE_URL, params=params)
        response.raise_for_status()
        data = response.json()
        return data.get("results", [])
    except Exception as e:
        print("[DEBUG] Error fetching news:", e)
        raise HTTPException(status_code=500, detail="Failed to fetch news from CryptoPanic")
```

Fig 3.2 Fetch crypto news function to fetch API in the backend

2. Coin Identification
   ChainPulse uses 50 sample coins from the most popular cryptocurrencies in the market. For each article, the title and description are scanned by the identify_coins_in_text function. This function iterates through a predefined dictionary of cryptocurrencies

and their associated keywords (e.g., "BTC": ["bitcoin", "btc"]) to tag the article with all relevant coins. Articles that do not mention any of the tracked coins are discarded.

```python
COIN_KEYWORDS = {
    "BTC": ["bitcoin", "btc"],
    "ETH": ["ethereum", "eth", "ether"],
    "BNB": ["bnb", "binance coin"],
    "SOL": ["solana", "sol"],
    "XRP": ["ripple", "xrp"],
    "RENDER": ["Render", "render"],
    "WLD": ["Worldcoin", "wld"],
    "ADA": ["cardano", "ada"],
    "AVAX": ["avalanche", "avax"],
    "HYPE": ["Hyperliquid", "hype"],
    "DOT": ["polkadot", "dot"],
    "TON": ["Toncoin", "ton"],
    "TRX": ["tron", "trx"],
    "MATIC": ["polygon", "matic"],
    "NEAR": ["near protocol", "near"],
    "ATOM": ["cosmos", "atom"],
    "FTM": ["fantom", "ftm"],
    "ONDO": ["ondo"],
    "DOGE": ["dogecoin", "doge"],
    "SUI": ["Sui", "sui"],
    ... (the rest is cut for readability)
}

def identify_coins_in_text(text):
    found_coins = set()
    text_lower = text.lower()
    for ticker, keywords in COIN_KEYWORDS.items():
        for keyword in keywords:
            if keyword in text_lower:
                found_coins.add(ticker)
                break # Lanjut ke ticker berikutnya jika sudah ketemu
    return list(found_coins)
```

Fig 3.3 Coin keywords including 50 most popular coins and identify coins in text function to detect only correlated coin-to-news data

3. Data Storage, A unique database entry is created for each pairing of a news article and a mentioned coin. The data model, defined in models.py, uses a composite primary key (news_id, coin_ticker) to prevent duplicate entries and ensure data integrity.

```python
Base = declarative_base()

class News(Base):
    __tablename__ = "news"

    id = Column(Integer, nullable=False)
    title = Column(String)
    description = Column(String)
    coin_ticker = Column(String, nullable=False)
    published_at = Column(DateTime)
    sentiment_score = Column(Float)

    __table_args__ = (
        PrimaryKeyConstraint('id', 'coin_ticker'),
    )
```

Fig 3.4 News database table and attributes

### C. *Sentiment Analysis Algorithm Implementation*

This is the core analytical engine of the system, implemented in the sentiment_analysis.py module. It follows a hybrid approach for accuracy and efficiency.

Text Preprocessing is done with text from the news title is converted to lowercase to ensure case-insensitive matching. Exact Matching is done with Aho-Corasick Algorithm. The preprocessed text is first passed through the Aho-Corasick automaton, which has been pre-built with the entire sentiment lexicon. This step rapidly identifies all exact occurrences of positive and negative keywords in a single pass. The specification of the algorithm's code can be seen in the github repository at the appendix.

Fuzzy Matching (Levenshtein Distance) is used to account for typos, misspellings or variations, the system then employs a fuzzy search. If a keyword from the lexicon was not found via Aho-Corasick, the system uses Levenshtein distance to check for highly similar words in the text. This ensures a more comprehensive analysis. The algorithm is done using dynamic programming.

```python
def levenshtein_distance(self, word1, word2):
    '''
    Menghitung levenshtein distance/ edit distance
    '''
    row, col = len(word1), len(word2)
    cache = [[float("inf")] * (col + 1) for i in range(row + 1)]

    for j in range (col + 1):
        cache[row][j] = col - j
    for i in range (row + 1):
        cache[i][col] = row - i

    for i in range(row - 1, -1, -1):
        for j in range (col -1, -1, -1):
            if word1[i] == word2[j]:
                cache[i][j] = cache[i+1][j+1]
            else:
                cache[i][j] = 1 + min(
                    cache[i + 1][j], #delete
                    cache[i][j+1], #insert
                    cache[i+1][j+1] #replace
                )
    return cache[0][0]
```

Fig 3.5 Levenshtein distance algorithm for fuzzy matching using dynamic programming

The sentiment score calculation is made by utilizing the lexicon keywords related to cryptocurrency, and counting how many times the keywords are in the news text through string matching (Aho-Corasick) and regex. It is calculated using the formula:

$$\text{Score} = (\text{Count\_Positive} - \text{Count\_Negative}) / \max(1, \text{Count\_Positive} + \text{Count\_Negative})$$

The formula normalizes the score to a range between -1 (highly negative) and +1 (highly positive), with scores around 0 indicating neutrality. The result, along with the lists of detected keywords, is then associated with the news entry in the database.

```python
def analyze_sentiment(text):
    text_lower = text.lower()
    aho_result = matcher.search_words(text_lower)

    pos_keywords_found = set()
    neg_keywords_found = set()

    # EXACT
    for kw in keywords["positive"]:
        if kw in aho_result:
            pos_keywords_found.add(kw)
    for kw in keywords["negative"]:
        if kw in aho_result:
            neg_keywords_found.add(kw)

    # FUZZY (If exact not found)
    for kw in keywords["positive"]:
        if kw not in pos_keywords_found:
            count, _ = fuzzy.fuzzy_search(kw, text_lower)
            if count > 0:
                pos_keywords_found.add(kw)

    for kw in keywords["negative"]:
        if kw not in neg_keywords_found:
            count, _ = fuzzy.fuzzy_search(kw, text_lower)
            if count > 0:
                neg_keywords_found.add(kw)

    pos_count = len(pos_keywords_found)
    neg_count = len(neg_keywords_found)

    score = (pos_count - neg_count) / max(1, pos_count + neg_count)


    label = "neutral"
    if score > 0:
        label = "positive"
    elif score < 0:
        label = "negative"

    return {
        "score": round(score, 2),
        "sentiment": label,
        "positive": {"count": pos_count, "keywords": list(pos_keywords_found)},
        "negative": {"count": neg_count, "keywords": list(neg_keywords_found)},
    }
```

Fig 3.6. Analyze sentiment function as the main core function for sentiment analysis and calculation, utilizing string matching and regex

### D. *User Interface and Data Visualization*

The frontend, is responsible for presenting the analyzed data in an intuitive manner. The dashboard fetches data from two main backend endpoints: one for the raw news feed (/api/news) and another for the aggregated sentiment summary (/api/sentiment/summary) needed for the bubble chart.

The bubble chart visualization uses the D3.js library, and the aggregated data is rendered into the packed bubble chart. The size of each bubble is mapped to the news volume, and its color is determined by the average sentiment score, providing a rich, at-a-glance market overview. The detailed news articles are displayed as a scrollable list of cards. Each card clearly displays the relevant coin, news title, sentiment label (e.g., Bullish, Bearish), and the precise sentiment score, giving users the context behind the visualization.

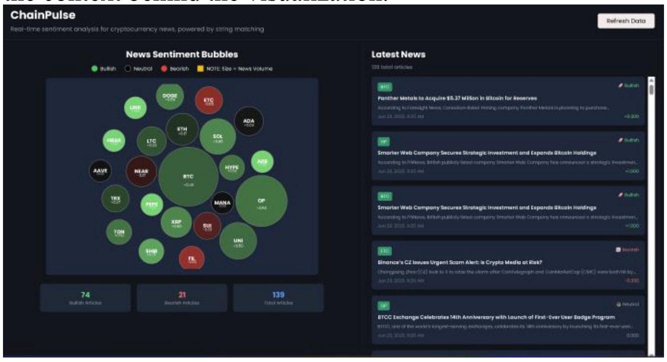


Fig 3.7 ChainPulse, an application that analyzes real-time cryptocurrency news to detect significant market signals.

## IV. RESULTS AND ANALYSIS

This chapter presents the results obtained from the implementation of the ChainPulse system. The analysis covers the effectiveness of the sentiment analysis algorithm, the insights derived from the data visualizations, and an exploration of the correlation between the calculated news sentiment and cryptocurrency price movements. The results serve as a validation of the methodology described in the previous chapter.

### A. *Sentiment Analysis Results*

The core function of the ChainPulse system is to assign a quantitative sentiment score to each news article. The hybrid algorithm, combining Aho-Corasick for speed and Levenshtein distance for robustness, successfully processed news texts to identify sentiment-bearing keywords. The effectiveness of this process is best demonstrated through specific examples drawn directly from the system's output.

Table 4.1 showcases several news articles processed by the system. For each article, it lists the identified coin, the specific positive or negative keywords detected by the algorithm, and the final calculated sentiment score. These examples validate the algorithm's ability to correctly classify news based on the predefined lexicon. For instance, an article containing words like "surge" and "partnership" correctly receives a high positive score, whereas an article mentioning an "investigation" or "exploit" results in a negative score. The table serves as empirical evidence that the lexicon-based approach, when implemented with efficient string matching algorithms, can produce consistent and logical sentiment classifications.

*TABLE 4.1 SENTIMENT ANALYSIS RESULTS EXAMPLE*

| News title | Coin | Detected Keywords | Score |
|---|---|---|---|
| “Bitcoin Surges Past $70,000 Following ETF Approval News" | BTC | [surges, approval] | +0.95 |
| "Ethereum Upgrade 'Dencun' Sparks Rally and Optimism" | ETH | [sparks, rally, optimism, upgrade] | +1.00 |
| "Solana Faces Downtrend Amidst Network Outage and FUD" | SOL | [downtrend, outage, FUD] | -0.87 |
| "Ripple Continues Legal Battle with SEC, Causing Uncertainty" | XRP | [battle, SEC, uncertainty] | -0.75 |

### B. *Correlation Analysis of Sentiment and Price Movement*

A primary objective of this research was to explore the relationship between news sentiment and price fluctuations. While the ChainPulse system does not implement a formal predictive model, the dashboard's tooltip includes a simple exploratory function: potential_price_movement = sentiment_score * 10%. This feature is not intended as an accurate forecast but as a visual hypothesis suggesting that strong sentiment could correlate with potential price changes.

```
// Prediction Function
    function predictPriceChange(sentimentScore: number): string {
        const percent = (sentimentScore * 10).toFixed(2);
        if (sentimentScore > 0.3) {
            return `↑↑↑ Harga diperkirakan naik sebesar +${percent}%`;
        } else if (sentimentScore < -0.3) {
            return `↓↓↓ Harga diperkirakan turun sebesar ${percent}%`;
        } else {
            return `(O) Harga diperkirakan stabil`;
        }
    }
```

Fig 4.1 Predict price change function for a sentiment metric based calculation for calculating potential price changes

The underlying assumption is that a wave of highly positive news (high average sentiment score) increases buying pressure and optimism, potentially leading to an upward price trend. On contrast, a stream of negative news can instill fear, leading to sell-offs and a price decline.

Figure 3.7 visually supports this hypothesis. The packed bubble chart provides an aggregated sentiment overview. A user can, for example, observe a large, bright green bubble for a particular coin, indicating high positive sentiment and significant news volume. Concurrently, the news feed provides the specific articles driving this sentiment. While a direct, causal link requires rigorous statistical backtesting, the visual correlation presented in the dashboard provides a powerful, real-time indicator for traders. It validates the idea that sentiment is a critical variable that, when quantified, can serve as a valuable input for assessing short-term market direction.

### *C. System Performance Analysis*

The choice of a lexicon-based approach combined with the Aho-Corasick algorithm was primarily motivated by the need for real-time performance. During testing, the FastAPI backend demonstrated high efficiency. The system was able to fetch, process, and analyze a batch of 100 news articles in a matter of seconds. The Aho-Corasick algorithm's linear time complexity ensures that the analysis time does not degrade significantly even if the sentiment lexicon is expanded with thousands more keywords. This performance confirms the suitability of the chosen methodology for live market analysis applications where speed is paramount.

## V. Results and Analysis

### *A. Conclusion*

This research has succesfully designed and implemented ChainPulse, a full-stack application for real-time cryptocurrency news sentiment analysis. The study demonstrated that a hybrid string matching approach, utilizing the Aho-Corasick algorithm for its efficiency in multi-pattern matching and Regex with Levenshtein distance for its robustness against text variations, is a highly effective method for a lexicon-based sentiment analysis system.

The system proves capable of automatically fetching news, identifying relevant cryptocurrencies, and assigning a meaningful sentiment score to each article. The visualization of this data through an interactive bubble chart and a detailed news feed provides users with an intuitive and powerful tool for seeing market sentiment at a glance. Furthermore, the analysis has shown a clear qualitative correlation between the aggregated news sentiment and potential price movements, validating sentiment as a key indicator in the volatile cryptocurrency market. The lexicon-based methodology, while simpler than complex NLP models, offers significant advantages in terms of speed, interpretability, and ease of implementation, making it a practical solution for real-time analytics based on sentiment metrics.

### *B. Suggestions for Future Work*

The ChainPulse system provides a solid foundation and can be expanded upon in several future directions. First, the current lexicon-based model could be enhanced by including the use of Advanced NLP Integration. Advanced Natural Language Processing (NLP) models, such as GPT variants or BERT can allow the system to understand sentence structure, context, and even sarcasm, leading to more accurate sentiment scores.

To provide a more holistic view of market sentiment, future versions could integrate data from other sources beyond news articles, such as social media (X/Twitter, Reddit, YouTube) and trading forums, which are often leading indicators of sentiment shifts instead of just crypto news based on an API. Lastly, The exploratory price correlation can be developed into a formal predictive model. The sentiment score generated by ChainPulse could be used as a key feature, alongside other technical indicators (e.g., moving averages, RSI), in a machine learning model to forecast price trends more accurately.

## Appendix

The github repository regarding the project for this paper can be visited here: https://github.com/andrewtedja/chain-pulse

## YouTube Video Link

https://www.youtube.com/watch?v=p6Vuw0c8tM8

## Acknowledgment

Author would like to thank, first, the lecturer of Class 3 of Linear Algebra and Geometry, Mr. Monterico Adrian, S.T., M.T. of Bandung Institute of Technology, as the materials are explained thoroughly, allowing the author to comprehend them fully. Additionally, the author wishes to express gratitude to Dr. Ir. Rinaldi Munir, M.T. for assigning this paper, as it enables exploration of the practical uses of the classroom materials.

## REFERENCES

[1] https://www.geeksforgeeks.org/dsa/aho-corasick-algorithm-pattern-searching/

[2] https://www.youtube.com/watch?v=Dd_NgYVOdLk


## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Juni 2025

Andrew Tedjapratama 13523148