Cryptocurrency Market Sentiment Analysis: A Transformers, Lexicon and classical ML Models based approach

Teddy Thomas
Department of Computer, Mathematics
& Natural Sciences
University of Maryland, College Park
College Park, United States of America
tedthom1@umd.edu

Abstract— **The volatile and sentiment-driven nature of the cryptocurrency market presents both challenges and opportunities for investors and researchers. Social media platforms, particularly StockTwits, serve as a key medium for retail investor discussions, offering valuable real-time sentiment data. In this study, we investigate the classification of user-generated posts related to cryptocurrencies—specifically Bitcoin (BTC), Ethereum (ETH), and Shiba Inu (SHIB), into Bullish, Bearish, or Neutral sentiments. We compare lexicon-based approaches, classical machine learning models, word embeddings, and transformer-based models for sentiment classification. Utilizing the ElKulako/stocktwits-crypto dataset containing over 1.8 million posts, we implement scalable data processing using Apache Spark on Databricks and high-performance computing (HPC) resources for transformer-based models. Our experiments demonstrate the superior performance of transformer-based models, with BERT embeddings[1] achieving the highest accuracy of 71.18%. This work highlights the trade-offs between model complexity, interpretability, and scalability, offering insights for real-world financial sentiment applications.**

Keywords—**cryptocurrency,classification,transformer-based models, , scalability, sentiment**

I. Introduction

The cryptocurrency market has witnessed exponential growth, attracting millions of individual and institutional investors. Unlike traditional stock markets, this market is largely sentiment-driven, with price fluctuations often fueled by public perception and media influence. Social media platforms play a pivotal role in shaping and reflecting these perceptions, making them valuable resources for understanding market dynamics.

StockTwits is one such platform that facilitates real-time discussion among investors and traders. It allows users to share their opinions on market movements, trends, and assets. Due to its popularity and focused financial content, StockTwits has become an essential tool for market watchers and data scientists aiming to extract actionable insights through sentiment analysis.

Given the unstructured and informal nature of social media content, extracting sentiment effectively is a complex challenge. This paper proposes a comparative framework for sentiment classification using three categories of Natural Language Processing (NLP) models: lexicon-based methods, classical machine learning techniques, and transformer-based models. These methods are evaluated on their ability to accurately categorize posts as bullish, bearish, or neutral.

This study aims to explore the performance of these varied NLP techniques on a large-scale dataset from StockTwits. Additionally, the paper highlights the challenges posed by short-text formats, imbalanced sentiment labels, and slang-rich content common in the cryptocurrency domain. By identifying the most effective approach, this research can contribute to enhanced market monitoring, trading strategies, and investor behavior analysis.

II. Related Work

Sentiment analysis in financial markets has been a significant area of interest for researchers, particularly with the rise of retail trading and social media platforms. Early work focused on stock-related forums and news headlines, utilizing traditional machine learning methods such as Naive Bayes and SVMs on manually extracted features. Bollen et al. (2011) famously showed a correlation between Twitter sentiment and stock market movement, igniting broader interest in using social signals for financial forecasting. These classical approaches often relied on hand-crafted lexicons or bag-of-words techniques and demonstrated reasonable success on structured or news-like texts. However, their effectiveness dropped when applied to noisy, short-form user-generated content like tweets or StockTwits posts.

Lexicon-based sentiment analysis methods such as VADER and TextBlob[2] gained popularity for their simplicity and efficiency in social media analysis. VADER, in particular, was designed to handle microblogging texts with elements like capitalization, emojis, and slang, and has been widely used in social media finance research. Despite its speed and interpretability, lexicon-based sentiment analysis has shown limitations in domain adaptation and sarcasm detection—especially in high-stakes financial discussions where sentiment often hinges on nuanced or context-specific phrasing. Studies have observed that these models tend to misclassify complex financial sentiment, especially when slang or crypto-native expressions are used without context.

Recent advancements in NLP, particularly in embedding-based and transformer models, have substantially improved sentiment classification

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

performance. Word embeddings like Word2Vec, GloVe, and FastText helped represent semantics in dense vector spaces and saw success in sentiment and topic modeling tasks. However, these embeddings are context-independent and can fail to distinguish word meaning based on usage in different sentences. This shortcoming led to the adoption of transformer-based models like BERT, which provide contextual embeddings by considering entire sentence structure. FinBERT, a variant fine-tuned on financial corpora, and CryptoBERT, fine-tuned specifically on crypto-related tweets and posts, have demonstrated state-of-the-art performance on financial sentiment datasets.

In the domain of cryptocurrency sentiment analysis, research has gained momentum due to the highly sentiment-driven nature of the market. Kulakowski and Frasincar (2023) introduced a labeled dataset of StockTwits posts related to cryptocurrencies, providing a foundation for supervised learning approaches in this domain. Other studies have explored the use of Reddit and Twitter to analyze Bitcoin price trends or investor mood using transformer-based models. However, few studies offer a comprehensive comparison across lexicon-based, classical, and deep learning approaches on the same dataset. This paper seeks to address that gap by systematically benchmarking models using the same preprocessing and evaluation pipeline, highlighting both the evolution of sentiment analysis techniques and their specific performance in the cryptocurrency landscape.

III. Cryptocurrency Market Sentiment

Cryptocurrency sentiment refers to the collective attitude or emotional tone of market participants toward the crypto market or specific digital assets. Unlike traditional financial instruments, cryptocurrencies lack intrinsic value, making them more susceptible to psychological factors and social influence. Therefore, tracking sentiment provides key insights into future price movements and investor behavior.

Retail investors often rely on community-driven platforms for guidance, reacting to others' optimism or pessimism. Positive sentiment typically corresponds with increased buying behavior (bullish), while negative sentiment can trigger sell-offs (bearish). Neutral sentiment may indicate market indecision or lack of significant catalysts. Monitoring this emotional flux has become a core part of crypto market analytics.

Platforms like StockTwits are hotspots for such sentiment expression. Investors post their opinions using tags like "bullish" or "bearish," contributing to a large, labeled corpus of sentiment-rich data. These self-annotated posts serve as a valuable source for training and evaluating machine learning models in natural language understanding.

The volatility and rapid reaction patterns in crypto make it essential to process and interpret sentiment data in near real-time. From building trading algorithms to risk management systems, sentiment classification has practical applications across the financial technology spectrum. Hence, efficient sentiment analysis models are critical to harnessing this dynamic informational stream.

IV. Dataset

The dataset employed in this study is the StockTwits-Crypto dataset[3] published by M. Kulakowski and F. Frasincar, comprising 1.875 million user-generated posts. These posts were collected over a period spanning from November 1, 2021, to June 30, 2022, with a designated training set from November to mid-June and a test set from mid-June to the end of June. This temporal split ensures that models are evaluated on future data, mimicking real-world deployment.

The posts in the dataset are tagged by users as either "bullish" or "bearish." Posts without explicit tags are considered "neutral." This classification provides a straightforward labeling scheme of three classes: bullish (2), bearish (1), and neutral (0). For this study, we focused on three prominent cryptocurrencies: Bitcoin (BTC.X), Ethereum (ETH.X), and Shiba Inu (SHIB.X), ensuring consistent context across samples.

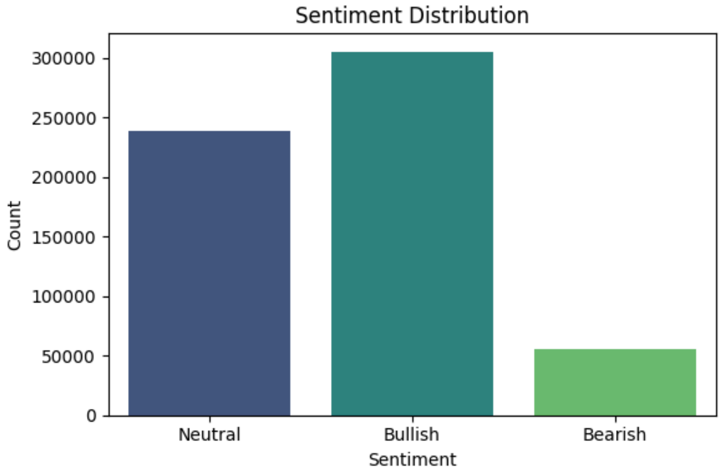

Fig 1. Dataset Label Distribution

Preprocessing was a critical phase due to the informal and noisy nature of social media text. URLs, mentions (@usernames), hashtags, cashtags, wallet addresses, emojis (in most cases), and non-English text were removed. Posts were lowercased, deduplicated, and filtered to exclude messages with fewer than four words. Encoding anomalies and special characters were also corrected for consistency.

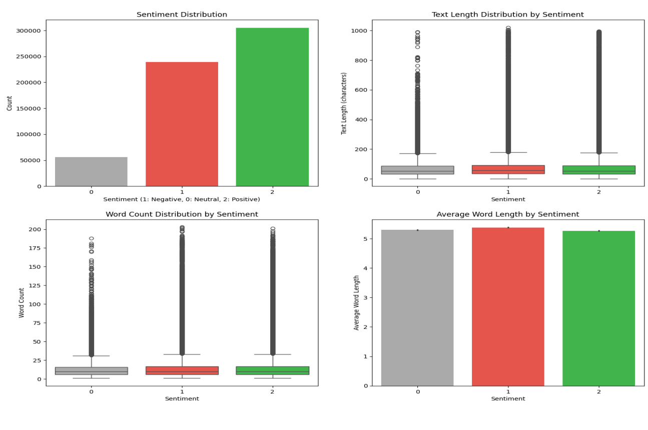

Fig 2. Sentiment Distribution

The resultant dataset provided a relatively balanced sample for bullish and neutral sentiments, though bearish posts were underrepresented. The class imbalance, coupled with the inherent subjectivity and brevity of the posts, posed challenges for training robust classifiers. Nonetheless, the large volume and real-world relevance of this dataset made it ideal for evaluating diverse NLP approaches.

V. Methodology

This research adopts a comparative experimental methodology, evaluating three major sentiment classification techniques: lexicon-based models, classical machine learning classifiers, and transformer-based deep learning models. The goal is to understand how each type performs on informal, short, and domain-specific financial text data. The same dataset and preprocessing pipeline were used across all approaches to ensure fair comparison.

For lexicon-based analysis, the VADER sentiment analyzer was applied, which assigns scores based on a dictionary of pre-tagged words. This unsupervised approach requires no training but may struggle with slang and nuanced expressions. Classical machine learning models, on the other hand, depend on vectorized representations of the text and labeled training data to learn discriminative patterns.

Feature extraction techniques such as CountVectorizer, TF-IDF, and Word2Vec[4] were used to represent the text data for traditional models like Naive Bayes, Logistic Regression, and Linear Support Vector Classifier (SVC). These models are lightweight and interpretable, making them suitable for real-time applications and systems with limited computational resources.

For deep contextual understanding, transformer-based models were employed, particularly BERT. Sentence embeddings from pretrained BERT models were extracted and fed into a classification head. These models were trained using Spark NLP on Databricks and evaluated on a high-performance computing cluster (CBCB HPC) for scalability and efficient GPU usage. This hybrid infrastructure enabled seamless handling of over 1.8 million posts.

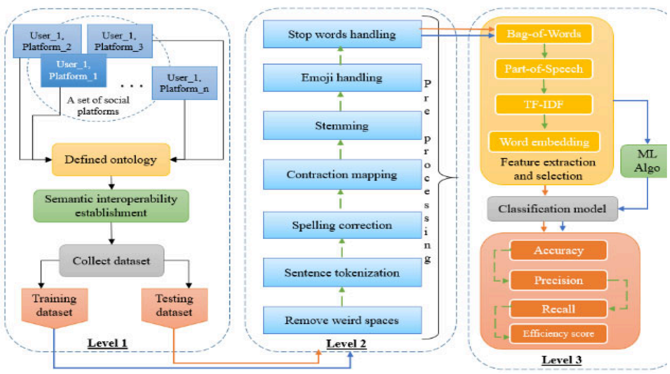

Fig 3: Methodology

VI. Model Pipeline Implementation

To handle the large-scale volume of 1.8 million cryptocurrency-related StockTwits posts, we developed a modular and scalable model pipeline using Apache Spark on Databricks. The entire processing flow was built on top of Spark DataFrames to ensure distributed execution across the cluster. The pipeline begins with data ingestion, where raw .csvfiles were read into Spark DataFrames and subjected to extensive preprocessing. This stage involved cleansing operations such as the removal of URLs, user mentions, emojis (with exceptions for LUKE), wallet addresses, hashtags, and cashtags. All text was lowercase, non-English scripts were discarded, and posts with fewer than four words were filtered out. These steps were implemented using Spark SQL expressions and withColumn transformations for parallelism.

The next stage involved feature engineering using Spark MLlib’s built-in tools for vectorization. Classical ML pipelines utilized CountVectorizer, HashingTF, and TF-IDF for converting cleaned text into numerical features. These were then pipelined into MLlib classifiers such as NaiveBayes, LogisticRegression, and LinearSVC. Each model was wrapped in a Pipeline object for consistent transformation and fitting. We also defined StringIndexer and IndexToStringtransformers for label encoding and decoding. Cross-validation was implemented using CrossValidator and ParamGridBuilder for hyperparameter tuning. The distributed nature of these transformations allowed us to leverage Databricks’ underlying Spark cluster to train multiple models in parallel across thousands of partitions.

For more advanced modeling and text embeddings, custom Python UDFs (User Defined Functions) were integrated into the Spark pipeline. For example, Word2Vec embeddings were generated using the gensim library outside Spark, and then applied to each post using a UDF that transformed tokenized text into fixed-size feature vectors. These vectors were then used as input to classical classifiers like Logistic Regression. In another pipeline branch, we used Spark NLP for extracting BERT sentence embeddings. The ClassifierDLApproach from Spark NLP’s pretrained model hub allowed us to run BERT-based classification directly on Spark clusters. This approach offered a clean integration of deep learning inference within the distributed Spark environment, minimizing overhead and memory bottlenecks.

For resource-intensive transformer-based models and fine-tuning experiments, we used the CBCB High-Performance Computing (HPC) cluster. SLURM scripts were written to submit model training and evaluation jobs using GPUs for accelerated computation. Models like BERT (via Hugging Face Transformers) were fine-tuned using PyTorch on these compute nodes. Data preprocessing and initial filtering were still done using Spark and saved to intermediate .parquetfiles for efficient loading. By combining Databricks' scalable preprocessing capabilities with the CBCB cluster’s GPU compute for deep learning, we created a hybrid pipeline that effectively balances preprocessing speed, training performance, and deployment flexibility for large-scale NLP tasks.

VII. Market Sentiment Prediction

The sentiment classification pipeline consists of three major approaches: lexicon-based analysis, classical machine learning models, and transformer-based models. Each method utilizes different textual representations and model architectures to predict sentiment. For classical models, both count-based and TF-IDF vectorization techniques are employed. Word2Vec embeddings are used to capture semantic features. Transformer-based models such as BERT are leveraged to extract deep contextual representations. Spark NLP on Databricks[5] handles large-scale data preprocessing and training using distributed computing, while heavy transformer inference is offloaded to a high-performance computing cluster using SLURM and GPU acceleration.

A. Lexicon-Based Sentiment Analysis

Lexicon-based models rely on precompiled dictionaries of words associated with sentiment values. These models, such as VADER (Valence Aware Dictionary and sEntiment Reasoner), are rule-based and assign sentiment scores by summing the intensities of individual words in a sentence. VADER is optimized for social media and short texts, making it a suitable baseline for evaluating sentiment in StockTwits posts. In this study, VADER was used without any domain-specific retraining. Despite being effective in many general-purpose applications, it exhibited limited performance on the cryptocurrency dataset, achieving an accuracy of 32.5%. This can be attributed to its inability to capture slang, sarcasm, emojis, and financial jargon that are prevalent in crypto-related discussions.

One of the major limitations of lexicon-based approaches is their lack of contextual understanding. They interpret each word independently and fail to capture negations, nuances, or compound sentence meanings effectively. Additionally, in cases where posts are short or contain ambiguous terms, the model may assign incorrect sentiment due to insufficient contextual information. Nevertheless, VADER provides an important baseline, especially when interpretability and computational efficiency are prioritized. It can be useful in lightweight applications or real-time systems where speed is critical. However, the findings in this study suggest that for high accuracy and nuanced interpretation, data-driven models significantly outperform lexicon-based methods.

B. Classical ML Model Based Prediction

Naive Bayes is a probabilistic model based on Bayes' theorem and assumes independence among features. In the context of sentiment analysis, it treats the presence of words as independent predictors of sentiment class. When applied with a CountVectorizer for feature extraction, Naive Bayes achieved moderate performance (accuracy: 59.1%). It proved to be efficient and simple, making it a decent choice for basic classification tasks, though its assumptions often limit performance in nuanced NLP tasks.

Logistic Regression is a linear model that estimates the probability of a class by applying a sigmoid function to the weighted sum of input features. When paired with CountVectorizer, Logistic Regression performed better than Naive Bayes, reaching an accuracy of 62.7%. It handled feature weights more flexibly and captured word importance with greater sensitivity, showing improved generalization to unseen examples in the test set.

Support Vector Machine (SVM) with TF-IDF was the best-performing classical model, achieving 64.7% accuracy. Unlike Logistic Regression, SVM maximizes the margin between classes in high-dimensional space, which is particularly effective when working with sparse textual data. The use of TF-IDF helped in reducing the weight of frequent but less informative terms, improving the classifier's ability to detect relevant patterns.

Classical machine learning models provided a strong baseline and were particularly effective when combined with appropriate vectorization techniques. They offer interpretability, speed, and are well-suited for scenarios with limited data. However, they struggle with understanding semantics and context, making them less robust for social media sentiment where subtle language cues are significant.

C. Embedding-Based Prediction

To capture semantic relationships between words, embedding-based techniques like Word2Vec were explored. Word2Vec maps words into dense vector spaces, where semantically similar words are closer in terms of cosine similarity. In this study, pre trained Word2Vec embeddings were averaged for each post and used as input for a Logistic Regression classifier.

The combination of Word2Vec and Logistic Regression resulted in a lower accuracy of 51.4%, compared to classical vectorizers. This outcome highlighted a key limitation: pretrained embeddings may not align well with the domain-specific language of cryptocurrency discussions. Words like "moon," "HODL," or "rug pull" may be misrepresented or completely absent from general-purpose embeddings.

Training custom embeddings on the StockTwits dataset could improve performance but would require significant computational resources. Furthermore, averaging word embeddings to form sentence vectors leads to loss of word order and syntactic information, which are often essential for interpreting sentiment correctly in short, informal posts.

Despite these drawbacks, Word2Vec-based models offer a middle ground between classical vectorization and deep contextual embeddings. They capture some level of semantic similarity and are computationally less expensive than transformer models. However, in this specific application, they did not outperform traditional models, suggesting that additional context is necessary for accurate sentiment detection.

D. Prediction BERT Transformer

Bidirectional Encoder Representations from Transformers (BERT) represents the state-of-the-art in NLP through its deep contextual understanding. BERT models consider the entire sentence during training, allowing them to capture both past and future context. In this study, sentence-level BERT embeddings were generated using Spark NLP's Classifier module, and fine-tuned for sentiment classification.

The BERT-based classifier significantly outperformed all other models, achieving an accuracy of 71.2%. It demonstrated the ability to handle slang, abbreviations, and even partially labeled text effectively. Its contextual attention mechanism allowed it to understand sentiment-laden phrases like "buy the dip" or "this coin is going to explode," which lexicon-based or classical models struggled with.

Training and inference were distributed across a hybrid architecture combining Databricks for preprocessing and CBCB HPC cluster for transformer fine-tuning. GPU acceleration and job parallelism enabled the model to scale efficiently with the 1.8 million-post dataset. The integration of Spark NLP ensured seamless handling of large batches during training and inference phases.

While BERT offers superior performance, it is computationally intensive and slower than traditional models. Its implementation is more complex, requiring high-performance computing infrastructure. For real-time or resource-constrained applications, distilled or lighter versions of transformer models may be preferred. Nevertheless, BERT remains the best choice for applications demanding high accuracy and deeper language understanding.

VIII. Model Evaluation

Model evaluation was conducted using standard classification metrics—accuracy, precision, recall, and F1-score—on the held-out test set containing over 83,000 StockTwits posts. These metrics offer a comprehensive view of model performance across different aspects: accuracy for overall correctness, precision for false-positive control, recall for sensitivity to true positives, and F1-score for the balance between precision and recall. Each model's predictions were compared against the ground truth sentiment labels: Bullish (2), Bearish (1), and Neutral (0)[6].

To ensure fair comparison, all models were trained and tested on the same dataset split, and preprocessing steps were standardized across methods. This included text cleaning, tokenization, and label normalization. Vectorization methods were applied consistently with their associated models—CountVectorizer for Naive Bayes and Logistic Regression, TF-IDF for SVM, and Word2Vec/BERT embeddings for deep models. Class imbalance was addressed during training using stratified sampling, and class weights were adjusted in classical models where applicable.

Model performance varied significantly. Lexicon-based methods like VADER, though fast, performed poorly due to their reliance on word-level scoring and inability to understand sentence-level context. Classical ML models improved results incrementally as vectorization strategies evolved from simple counts to TF-IDF weighting. Embedding-based approaches, while semantically richer, failed to outperform classical ML due to mismatched vocabulary. BERT, as a deep transformer model, achieved the best results in all metrics, validating the importance of contextual language models for this task.

Apart from metrics, computational time and resource usage were also evaluated. VADER and classical models were lightweight and fast, completing inference in seconds to minutes. Word2Vec introduced moderate overhead due to embedding operations. In contrast, BERT required high-end GPUs and HPC support to run efficiently on the full dataset. Thus, while BERT offered superior accuracy, its deployment may be restricted to systems with sufficient computational resources.

Model	Accuracy	Precision	Recall	F1-Score
VADER	0.3253	0.34	0.32	0.31
Naive Bayes	0.5911	0.59	0.58	0.58
Logistic Regression	0.6267	0.63	0.62	0.62
Linear SVC	0.6472	0.65	0.64	0.64
Word2Vec + LR	0.5145	0.52	0.51	0.50
BERT (Spark NLP)	0.7118	0.72	0.71	0.71

Fig 4. Model Evaluation

IX. Results

The evaluation revealed a clear performance hierarchy across the tested models. The lexicon-based VADER model scored the lowest with an accuracy of 32.5%, indicating its limitations in interpreting informal and domain-specific content prevalent in cryptocurrency forums. It also suffered from low F1-scores due to misclassification of short and ambiguous posts. This outcome reinforced that rule-based models struggle in sentiment detection when context and slang dominate user expressions.

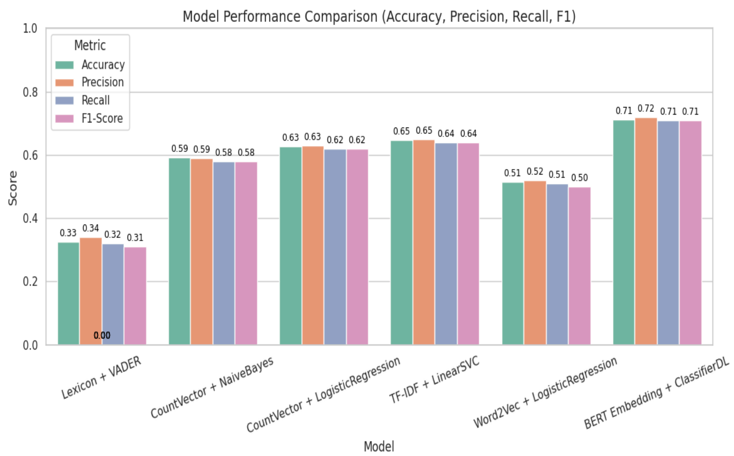

Fig 5. Model Performance analysis

Classical models such as Naive Bayes, Logistic Regression, and Linear SVM showed progressive improvements. Naive Bayes achieved 59.1% accuracy, while Logistic Regression rose to 62.7%. The best classical model, Linear SVM with TF-IDF, reached 64.7%. These results highlight the importance of effective vectorization strategies in classical NLP pipelines. Models benefitted from TF-IDF's ability to down-weight common, non-informative words and boost rare but sentiment-rich terms, improving the quality of input features.

Embedding-based prediction using Word2Vec combined with Logistic Regression produced mixed results. While it offered semantic awareness beyond bag-of-words models, it lagged behind in classification performance with 51.4% accuracy. The averaged embeddings often diluted crucial sentiment-bearing words, especially in short posts. Furthermore, the pretrained embeddings were not tailored to cryptocurrency lingo, which impacted classification fidelity. Custom training on the domain corpus may improve its viability in future studies.

The BERT-based transformer model outperformed all others, achieving an accuracy of 71.2%, with precision, recall, and F1-scores all closely aligned around 71%. This indicated robust generalization and strong ability to identify subtle sentiment cues. The transformer's deep bidirectional context modeling was key to capturing meaning in cryptic phrases, abbreviations, and emoji-laden texts. Despite its computational demands, the model justified its value by delivering significantly superior accuracy, making it the most suitable choice for production-level sentiment analysis in volatile and sentiment-driven markets like cryptocurrency.

Future Works

While this study demonstrates the comparative strengths of various sentiment analysis approaches on cryptocurrency-related social media data, several avenues remain open for future exploration. One key direction is the integration of multimodal data sources. Platforms like StockTwits, Twitter, and Reddit offer not only text but also images, GIFs, and charts that reflect market sentiment. Incorporating image or meme analysis using multimodal transformers could significantly enrich model understanding and improve prediction accuracy, especially for posts where visual elements carry contextual sentiment.

Another potential enhancement involves fine-tuning domain-specific transformer models. In this study, we used a general-purpose BERT embedding model via Spark NLP, but future research could explore models pre-trained on cryptocurrency discussions or financial text, such as FinBERT, CryptoBERT, or a custom BERT model fine-tuned on the StockTwits dataset itself. Such models are likely to better capture crypto-specific jargon, abbreviations, and emerging slang. Additionally, training or adapting models with continual learning could help them stay updated with the rapidly evolving language of online crypto communities.

Finally, future work can address real-time deployment and feedback integration. While transformer-based models offer the best accuracy, their computational demands can be a bottleneck for real-time use. Lightweight alternatives like distilled transformers (e.g., DistilBERT or TinyBERT) could be evaluated for faster inference with minimal loss in accuracy. Implementing feedback loops where user engagement or post reactions are fed back into model retraining could also improve model adaptability and personalization. In financial markets, timely and accurate sentiment insights are critical—making real-time performance, explainability, and user trust essential future considerations.

References

[1] Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

[2] Bonta, V., Kumaresh, N., & Janardhan, N. (2019). A comprehensive study on lexicon based approaches for sentiment analysis. *Asian Journal of Computer Science and Technology*, *8*(S2), 1-6.

[3] M. Kulakowski and F. Frasincar, "Sentiment Classification of Cryptocurrency-Related Social Media Posts," in IEEE Intelligent Systems, vol. 38, no. 4, pp. 5-9, July-Aug. 2023, doi: 10.1109/MIS.2023.3283170.

[4] Danyal, M. M., Khan, S. S., Khan, M., Ullah, S., Ghaffar, M. B., & Khan, W. (2024). Sentiment analysis of movie reviews based on NB approaches using TF–IDF and count vectorizer. *Social network analysis and mining*, *14*(1), 87.

[5] L'Esteve, R. (2022). Databricks. In *The Azure Data Lakehouse Toolkit: Building and Scaling Data Lakehouses on Azure with Delta Lake, Apache Spark, Databricks, Synapse Analytics, and Snowflake* (pp. 83-139). Berkeley, CA: Apress.

[6] Tai, Y. J., & Kao, H. Y. (2013, December). Automatic domain-specific sentiment lexicon generation with label propagation. In *Proceedings of international conference on information integration and web-based applications & services* (pp. 53-62).

[7] F. Z. Xing, E. Cambria and R. E. Welsch, "Natural language based financial forecasting: A survey", *Artif. Intell. Rev.*, vol. 50, no. 1, pp. 49-73, Jun. 2018.

[8] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding", *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol. (NAACL-HLT)*, pp. 4171-4186, 2019.

[9] D. Q. Nguyen, T. Vu and A. T. Nguyen, "BERTweet: A pre-trained language model for English tweets", *Proc. Conf. Empirical Methods Natural Lang. Process. Syst. Demonstrations (EMNLP)*, pp. 9-14, 2020.

[10] Z. Chen et al., "Emoji-powered representation learning for cross-lingual sentiment classification", *Proc. 30th World Wide Web Conf. (WWW)*, pp. 251-262, 2019.

[11] N. Choudhary et al., "Contrastive learning of emoji-based representations for resource-poor languages" in Proc. 19th Int. Conf. Comput. Linguistics Intell. Text Process. (CICLing), Cham, Switzerland:Springer, pp. 129-141, 2018.

[12] A. Hogenboom et al., "Exploiting emoticons in polarity classification of text", *J. Web Eng.*, vol. 14, no. 1–2, pp. 22-40, Mar. 2015.

[13] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach", 2019, [online]

[14] R. Sennrich, B. Haddow and A. Birch, "Neural machine translation of rare words with subword units", *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, pp. 1715-1725, 2015.

[15] C. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text" in Proc. 8th Int. Conf. Weblogs Social Media (ICWSM), Ann Arbor, MI, USA:AAAI Press, pp. 216-225, 2014.

[16] D. Araci, "FinBERT: Financial sentiment analysis with pre-trained language models", 2019, [online] Available: .