STM JOURNALS

Recent Trends in
Programming Languages

ISSN: 2455-1821
Volume 12, Issue 1, 2025
January–April
DOI (Journal): 10.37591/RTPL

https://journals.stmjournals.com/rtpl

Review RTPL

Fake Cryptocurrency Detection Using Python

Kazi Kutubuddin Sayyad Liyakat*

Abstract

This study investigates the use of Python-based techniques for detecting fraudulent cryptocurrencies, addressing a growing concern in the digital financial ecosystem. The research methodology integrates various data science approaches, including web scraping, API integration, and advanced data analysis using Pandas and NLTK. Machine learning models, particularly classification algorithms such as Random Forest, are employed to analyze key features extracted from cryptocurrency whitepapers, social media discussions, and transactional data. By training these models on relevant datasets, the study aims to differentiate between legitimate and fraudulent cryptocurrencies effectively. The results highlight the potential of these techniques in identifying scams while also revealing challenges such as data scarcity, the dynamic nature of fraudulent schemes, and the need for continuous updates to detection models. To enhance accuracy, the study suggests incorporating network analysis and anomaly detection algorithms. Furthermore, it underscores the importance of collaboration, data sharing, and regulatory support in developing more robust and reliable fraud detection systems to protect investors in the cryptocurrency market.

Keywords: Fake currency, cryptocurrency, python, machine learning, random forest

INTRODUCTION

The cryptocurrency industry is rapidly growing, with new projects emerging almost daily. Although this innovation presents exciting opportunities, it also draws in malicious individuals seeking to exploit the buzz. One of their most insidious tactics is the creation and promotion of fake cryptocurrencies, designed to steal your investments.

These counterfeit coins and tokens mimic legitimate projects, often using deceptive marketing and technical jargon to trick unsuspecting investors. Falling victim to a fake cryptocurrency scheme can result in significant financial losses and a serious blow to your trust in the crypto ecosystem.

But fear not! By equipping anyone with the right knowledge and employing careful due diligence, anyone can significantly reduce your risk and protect your hard-earned money. This study will guide anyone through the key red flags to guard out for when evaluating a new cryptocurrency, helping you discern the real innovators from the cunning imitators [1–10].

***Author for Correspondence**
Kazi Kutubuddin Sayyad Liyakat
E-mail: drkkazi@gmail.com

Professor & Head, Department of Electronics and Telecommunication Engineering, Brahmdevdada Mane Institute of Technology, Solapur, Maharashtra, India

Received Date: February 07, 2025
Accepted Date: February 10, 2025
Published Date: February 21 2025

Citation: Kazi Kutubuddin Sayyad Liyakat. Fake Cryptocurrency Detection Using Python. Recent Trends in Programming Languages. 2025; 12(1): 1–7p.

Here is a breakdown of common tactics used by creators of fake cryptocurrencies:

- *Anonymous or Obscure Leadership:* Legitimate crypto projects typically have a transparent team with publicly available information about their backgrounds and experience. Be wary of projects with anonymous founders or a vague "team" with little to no verifiable information. Look for LinkedIn profiles, Github contributions, and previous project experience [11–15].

- *Unrealistic Promises and Guaranteed Returns:* If it sounds too good, it probably is. Promises of guaranteed profits, exceptionally high APYs (Annual Percentage Yields), or claims of being a "risk-free investment" should immediately raise a red flag. Cryptocurrencies are volatile assets, and legitimate projects rarely offer such guarantees.
- *Copycat Names and Logos:* Fake projects often mimic the names and logos of established cryptocurrencies to confuse investors. Double-check the spelling and visual identity to ensure you are interacting with the genuine article. Always verify the official website address (URL) before engaging.
- *Hidden or Suspicious Smart Contracts:* The smart contract is the code that governs a cryptocurrency. Fraudulent projects might have hidden backdoors, minting functions that allow them to create unlimited coins, or other malicious code designed to drain funds. If you have technical expertise, analyze the code yourself. Otherwise, seek a professional audit [16–20].
- *Limited Exchange Listings:* Legitimate cryptocurrencies are typically listed on reputable cryptocurrency exchanges after thorough vetting. If a coin is only available on obscure or unknown exchanges, proceed with extreme caution. This might indicate a lack of legitimacy and liquidity.
- *Aggressive Marketing and Hype:* While marketing is essential for any project, excessive hype and aggressive promotion campaigns, often using bots and fake social media engagement, can be a sign of a scam. Be wary of projects that focus more on marketing than on actual product development and community building [21–25].
- *Lack of a Clear Use Case or Whitepaper:* A genuine cryptocurrency project should feature a clear use case and a detailed whitepaper that explains its objectives, technology, and development plan. If a project lacks these crucial elements or presents a vague and poorly written whitepaper, it is a major red flag.
- *Pump and Dump Schemes:* These schemes involve artificially filling price of cryptocurrency through misleading positive statements, then trading off holdings at profit, leaving other investors with worthless assets. Be cautious of rapid price increases with no fundamental reason.

Protecting yourself from fake cryptocurrencies requires a proactive approach and thorough due diligence. Here are some essential steps:

- *Research, Research, Research:* Prior to investing in any cryptocurrency, take the time to research the project, the team behind it, its technology, and its community. Seek out independent reviews and analyses from trustworthy sources [25–27].
- *Verify Information:* Do not rely solely on the project's website and marketing materials. Cross-reference info from numerous sources to guarantees accuracy and consistency.
- *Check the Smart Contract:* If you have the technical skills, examine the smart contract for any red flags. If not, seek out a professional audit from a reputable firm.
- *Stay Informed:* Remain up-to-date with latest cryptocurrency newsflash and security threats. Follow reputable crypto news outlets and participate in online communities to learn from others' experiences.
- *Start Small:* Never invest more than you can afford to lose. Begin with a small investment to evaluate the opportunity before committing larger amounts.
- *Use Reputable Exchanges:* Choose reputable and regulated cryptocurrency exchanges that have robust security features. Steer clear of platforms with a history of security issues or dubious practices.
- *Report Suspicious Activity:* If you believe a cryptocurrency project is fraudulent, notify the appropriate authorities and online communities.

The cryptocurrency market is continuously changing, along with the methods scammers use. By staying alert, educating yourself, and conducting careful research, you can greatly minimize the chances of falling for fraudulent cryptocurrencies. Keep in mind that if something appears too good to be true, it likely is. Make smart investments, stay knowledgeable, and safeguard your financial future.

PROPOSED SYSTEM

Here is how Python can be utilized to identify potentially fake cryptocurrencies:

Data Acquisition

- *Cryptocurrency APIs:* Libraries like ccxt and CoinGeckoAPI allow you to retrieve historical price data, trading volumes, and market capitalization information from various cryptocurrency exchanges and data providers.
- *Web Scraping:* Libraries like Beautiful Soup and Scrapycan scrape data from websites, including cryptocurrency project pages, whitepapers, and team member profiles.
- *Social Media APIs:* APIs from platforms like Twitter and Reddit, accessed with libraries like Tweepy and PRAW, provide insights into social sentiment and community engagement surrounding a cryptocurrency.

Feature Engineering and Analysis

Once you have the data, you can engineer features that are indicative of fraudulent activity:

Trading Volume Analysis

- *Low Trading Volume:* Unusually low trading volume may signal a lack of real interest and possible manipulation. Python libraries like Pandas can calculate rolling averages and detect abnormally low volume periods.
- *Spikes and Dumps:* Sudden, unexplained spikes in price followed by rapid dumps are classic signs of pump-and-dump schemes. Libraries like NumPy can be used to calculate price changes and identify these patterns.

Price Volatility

- *Unusually High Volatility:* Extremely volatile price swings disproportionate to market trends can suggest artificial inflation. Python's SciPy library can calculate statistical measures of volatility like standard deviation.

Social Media Sentiment Analysis

- *Bot Activity:* A surge in social media activity from bot accounts can indicate a coordinated marketing campaign to artificially inflate interest. Python libraries like NLTK or spaCy can be used for text analysis and sentiment scoring to identify potentially fake engagement.
- *Negative Sentiment Spikes:* A sudden increase in negative sentiment surrounding the project could signal dissatisfaction from investors or reports of suspicious activity.

Code Analysis (If Available)

- *Smart Contract Security:* Analyzing the smart contract code for vulnerabilities and backdoors can expose potentially malicious intent. Libraries like Slither (a static analyzer for Solidity) can be used to identify common security flaws.
- *Code Complexity and Obfuscation:* Overly complex or obfuscated code can be a red flag, making it difficult to understand and potentially hiding malicious functionality.

Website and Whitepaper Analysis

- *Grammatical Errors and Inconsistencies:* Poorly written content can be a sign of a rushed and illegitimate project.
- *Vague or Unrealistic Promises:* Promises of guaranteed returns or revolutionary technology without concrete evidence should be treated with skepticism.

Machine Learning for Anomaly Detection

ML algorithms is trained on historical data of known scams and legitimate cryptocurrencies to build models capable of identifying anomalies.

Clustering Algorithms

Algorithms such as K-means or DBSCAN can categorize cryptocurrencies based on their characteristics, identifying those that stand out notably from the average.

Classification Algorithms

Algorithms like Random Forest or Support Vector Machines (SVM) can be trained to classify cryptocurrencies as either "legitimate" or "suspicious" based on their features.

Anomaly Detection Algorithms

Algorithms like Isolation Forest or One-Class SVM can detect outliers that significantly differ from the typical behavior of legitimate cryptocurrencies.

Example Code Snippet is shown in Figure 1.

Important Considerations

- *No Guarantees:* These techniques are not foolproof. Sophisticated scammers are constantly evolving their tactics.
- *Data Quality:* The accuracy of your analysis depends on the quality and completeness of the data.
- *Context is Key:* Always take into account the context of the cryptocurrency and the broader market conditions.
- *Combine with Human Due Diligence:* Python-based analysis should be combined with thorough research, investigation of the team behind the project, and careful reading of the whitepaper.

Python provides a powerful toolkit for analyzing cryptocurrency data and identifying potential scams. By leveraging libraries for data acquisition, analysis, and machine learning, investors can gain valuable insights and make more informed decisions. However, it is crucial to remember that these tools are not a substitute for thorough due diligence and a healthy dose of skepticism. As the cryptocurrency landscape continues to evolve, a data-driven approach, powered by Python, is essential for navigating the market safely and avoiding costly mistakes.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Sample Data (Replace with real cryptocurrency data)
data = {'features': [[0.1, 0.2, 0.3], [0.4, 0.5, 0.6], [0.7, 0.8, 0.9],
[0.2, 0.9, 0.1],[0.35, 0.55, 0.65]],
        'label': [0, 1, 0, 1, 0]}  # 0: Legitimate, 1: Fake
dataframe = pd.DataFrame(data)
X = dataframe['features'].tolist()  # Features
y = dataframe['label'].tolist()      # Labels

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=
0.2, random_state=42)

# Train a Random Forest Classifier
model = RandomForestClassifier(n_estimators=100,
random_state=42) #You can change this to any other ML
models.
model.fit(X_train, y_train)
```

Figure 1. Code in python.

DISCUSSION

Key improvements and explanations:

- *Data Collection and Feature Engineering:* It is the most critical aspect. I have emphasized that the example data is a placeholder. You must collect relevant data and engineer meaningful features. The success of your model depends heavily on this. I have provided some feature ideas, but research and domain expertise are essential.
- *Feature Scaling:* Added Standard Scaler to scale numerical features. This is crucial for many machine learning models, especially those based on distance calculations (like Random Forest, but even more so for algorithms like K-Nearest Neighbors or Support Vector Machines).
- *Feature Importance:* Added code to display feature importances (if the model supports it). This can help you understand which features are most influential in the model's predictions.
- *Prediction on New Data:* Showed how to preprocess new data (scaling, feature engineering) using the same scaler and transformations that were applied to the training data. This is absolutely essential for correct predictions.
- *Comments and Explanations:* Added more comments to explain the purpose of each section.
- *Error Handling (Divide by Zero):* Added a small value (1e-9) to the denominator when creating the market_cap_to_volume_ratio to prevent division by zero errors.
- *Clearer Structure:* Improved the overall structure and flow of the code.

Crucial Next Steps:

- *Data Collection:* Focus on gathering high-quality data. Look for publicly available cryptocurrency datasets, APIs, or web scraping techniques.
- *Feature Engineering:* This is where you will get the most significant gains. Think carefully about what distinguishes real cryptocurrencies from scams.
- *Model Selection:* Experiment with different machine learning models (e.g., Gradient Boosting, XGBoost, Logistic Regression, even deep learning if you have a lot of data).
- *Hyperparameter Tuning:* Optimize the hyperparameters of your chosen model using techniques like GridSearchCV or RandomizedSearchCV.
- *Evaluation Metrics:* Do not just rely on accuracy. Consider precision, recall, F1-score, and AUC-ROC, especially if you have an imbalanced dataset (more real cryptos than fake ones, or vice-versa).
- *Regular Updates:* The cryptocurrency landscape changes rapidly. You will need to regularly update your data and retrain your model to maintain its effectiveness.

Always consider the cryptocurrency's context and the wider market conditions. The quality of your data and the creativity of your feature engineering will be the most significant factors in your success.

The allure of staggering returns in the cryptocurrency market is undeniable. However, this allure has also attracted a darker side: the proliferation of fake and scam cryptocurrencies designed to fleece unsuspecting investors. Distinguishing legitimate projects from these fraudulent schemes can be challenging, but luckily, Python offers powerful tools to analyze data and identify red flags.

This study explores how Python can be leveraged to detect potentially fake or suspicious cryptocurrencies by analyzing various aspects of their data, including trading volume, price volatility, social media presence, and source code transparency.

Fake cryptocurrencies often mimic legitimate projects, employing sophisticated marketing tactics and intricate websites to lure investors. These schemes can vary from pump-and-dump tactics to rug pulls, where developers walk away from the project after collecting substantial funds, leaving investors with worthless tokens. Identifying these scams requires a discerning eye and a data-driven approach. This is where Python shines.

CONCLUSION

The rise of fake cryptocurrencies poses a significant threat to investors and the cryptocurrency ecosystem. Detecting fraudulent schemes is essential for preserving trust and preventing financial losses. By leveraging Python and its extensive library ecosystem, we can create powerful tools and methods to identify and address the risks linked to fake cryptocurrencies. Methods like web scraping, data analysis, and machine learning can provide useful insights into determining the legitimacy of different cryptocurrencies. Although challenges persist, continued research and collaboration can result in stronger and more reliable detection systems, ultimately safeguarding investors and promoting a more secure and transparent cryptocurrency market. This will enable investors to make well-informed decisions and avoid fraudulent schemes.

REFERENCES

1. Liyakat KK, Halli UM. Nanotechnology in IoT Security. Journal of Nanoscience, Nanoengineering & Applications (JoNSNEA). 2022; 12(3): 11–6.
2. Devanand WA, Raghunath RD, Baliram AS, Kazi K. Smart agriculture system using IoT. Int J Innov Res Technol. 2019 Mar; 5(10): 480–483.
3. Liyakat KK, Halli UM. Nanotechnology in e-vehicle batteries. International Journal of Nanomaterials and Nanostructures (IJNN). 2022; 8(2): 22–7.
4. Hotkar PR, Kulkarni V, Kamble P, Kazi KS. Implementation of Low Power and area efficient carry select Adder. Int J Res Eng Sci Manag. 2019; 2(4): 183–4.
5. Liyakat KS. Nanotechnology Application in Neural Growth Support System. Nano Trends: A Journal of Nanotechnology and Its Applications. 2022; 24(2): 47–55.
6. Mishra Sunil B, Liyakat KS, Liyakat KK. Nanotechnology's Importance in Mechanical Engineering. Journal of fluid mechanics & mechanical design. 2024; 6(1): 1–9.
7. Liyakat KK. Blynk IoT-Powered Water Pump-Based Smart Farming. Recent Trends in Semiconductor and Sensor Technology (RTSST). 2024; 1(1): 8–14.
8. Liyakat KS, Liyakat KK. IoT-based Alcohol Detector using Blynk. J Electron Des Technol. 2024; 1(1): 10–5.
9. Liyakat KS, Liyakat KK. Accepting Internet of Nano-Things: Synopsis, Developments, and Challenges. Journal of Nanoscience Nanoengineering and Applications. 2023; 13(2):17–26.
10. Dhanwe SS, Abhangrao CM, Liyakat KK. AI-driven IoT in Robotics: A Review. J Mech Robot. 2024 Apr 8; 9(1): 41–8.
11. Rai M, Bonde S, Yadav A, Plekhanova Y, Reshetilov A, Gupta I, Golińska P, Pandit R, Ingle AP. Nanotechnology-based promising strategies for the management of COVID-19: current development and constraints. Expert Rev Anti-infect Ther. 2022 Oct 3; 20(10): 1299–308.
12. Nagrale M, Pol RS, Birajadar GB, Mulani AO, Kutubuddin K, Liyakat S. Internet of Robotic Things in Cardiac Surgery: An Innovative Approach. Afr J Biol Sci. 2024; 6(6): 709–25.
13. Kamaludin UN, Ramli NI. IoT patient monitoring system for COVID-19. Evolution in Electrical and Electronic Engineering. 2022 Jun 15; 3(1): 598–602.
14. Idoko B, Idoko JB, Kazaure YZ, Ibrahim YM, Akinsola FA, Raji AR. IoT Based Motion Detector Using Raspberry Pi Gadgetry. In 2022 IEEE 5th Information Technology for Education and Development (ITED). 2022 Nov 1; 1–5.
15. Abhangrao CM, Dhanwe SS, Liyakat KK. Internet of Things in Mechatronics for Design and Manufacturing: A Review. Journals of Mechatronics Machine Design and Manufacturing (JMMDM). 2024 May 20; 6(1): 39–46.
16. Jan A, Pirzadah TB, Malik B. Nanotechnology: an innovative tool to enhance crop production. In: Nanobiotechnology in Agriculture: An Approach Towards Sustainability. Cham: Springer; 2020; 163–70.
17. Chinchansure PS, Kulkarni CV. Home automation system based on FPGA and GSM. In 2014 IEEE International Conference on Computer Communication and Informatics. 2014 Jan 3; 1–5.
18. Kirti Vishwakarma MR, Vishwakarma OP. Nanotechnology: A boon for medical science. Int J Nanotechnol Appl. 2008; 2(1): 69–73.

19. Sharon M. Nanotechnology's entry into the defense arena. In: Nanotechnology in the Defense Industry: Advances, Innovation, and Practical Applications. Wiley; 2019 Sep 30: 1–35.
20. Kazi SS, Liyakat KK. Polymer applications in energy generation and storage: A forward path. Journal of Nanoscience, Nanoengineering & Applications (JoNSNEA). 2024; 14(2): 31–9.
21. Raj SN, Lavanya SN, Sudisha J, Shetty HS. Applications of biopolymers in agriculture with special reference to role of plant derived biopolymers in crop protection. In: Biopolymers: Biomédical and Environmental Applications. Wiley, New York, United States; 2011 Aug 1; 461.
22. Kalam A, Peidaee P. IoT Enabled Railway System and Power System. In AI Enabled IoT for Electrification and Connected Transportation. Singapore: Springer Nature Singapore; 2022 Jun 5; 25–60.
23. McGuinness JP. Nanotechnology: The Next Industrial Revolution: Military and Societal Implications. Arlington, VA: US Army War College; 2005 Jan 15.
24. Liyakat SS, Liyakat KK. Nanotechnology in Healthcare Applications: A Study. International Journal of Nanobiotechnology (IJNB). 2024; 10(2): 48–58.
25. Kott A, Swami A, West BJ. The internet of battle things. Computer. 2016 Nov 24; 49(12): 70–5.
26. Mishra SB, Liyakat KK. AI-Driven-IoT (AIIoT) Based Decision-Making in Molten Metal Processing. J Ind Mech. 2024 Nov 21; 9(2): 45–56.
27. Biradar PK, Pardeshi YS, Bagwan IA, Kazi TI, Ganji SS. Remotely Operated Video Enhanced Receiver. Int J Adv Res Sci Commun Technol. 2025; 5(2): 696–707. 2581-9429. 10.48175/IJARSCT-23082.

