Creating a Convolutional Autoencoder for Dogecoin Anomaly Detection

By Matt Carswell

DSAN 6660

AGENDA

01

Background

What is Dogecoin?

- Type of Cryptocurrency
- Launched December 6, 2013
- Symbol: DOGE
- Based on the popular "Doge" meme featuring a Shiba Inu dog
- Originally created as a joke or parody cryptocurrency, Dogecoin has grown into a widely recognized and used digital currency.

Source: https://dogecoin.com/

Digging a little deeper…

- **Market Cap: $62.08B**
- Over **147B coins** in circulation
- **7th largest** cryptocurrency (in terms of market cap)
- Built on its own blockchain, Dogecoin is a decentralized, open-source cryptocurrency.
- Low transaction cost, fast transaction time
- Extremely prone to volatility from speculation and “extraneous events”

Source: https://coinmarketcap.com/currencies/dogecoin/

Musk's first Dogecoin-related tweet occurred on December 20, 2020. Musk tweeted "One Word: Doge". Shortly after, the value of Dogecoin rose by 20%.[79] This was followed by a series of Dogecoin-related tweets by Musk in early February 2021 captioned "Dogecoin is the people's crypto" and "no highs, no lows, only Doge". Following these tweets, the value of Dogecoin rose by roughly 40%.[79]

On April 15, 2021, the price of Dogecoin rose by more than 100% after Musk tweeted an image of Joan Miró's *Dog Barking at the Moon* painting captioned "Doge Barking at the Moon",[80] a message which was taken by some as a reference to the industry slang term "to the moon",[81] meaning a hoped-for increase in a cryptocurrency's value.[82]

On May 8, 2021, Dogecoin fell as much as 29.5%, dropping to US$0.49 during Musk's *Saturday Night Live* appearance.[83] It then rose by 11% on May 20, 2021, shortly after Musk tweeted a Doge-related meme.[84] In the same month, the price of Dogecoin was up 10% in the hours after Musk tweeted a Reddit link for users to submit proposals to improve the cryptocurrency.[85]

On December 14, 2021, Dogecoin spiked more than 20% after Musk said that Tesla will accept the currency as a means of payment for Tesla merchandise.[86][87]

On June 16, 2022, Elon Musk was named in a complaint seeking damages of $258 billion. The complaint was filed in federal court in Manhattan by plaintiff Keith Johnson. Johnson cited Musk's repeated use of his massive social influence to promote the altcoin, which he claims artificially inflated the price.[88]

It was reported in 2013 that Musk thinks Dogecoin could be used for Twitter transactions.[89] On October 27, 2022, Elon Musk completed a deal to take Twitter private. This led to a sustained rise in Dogecoin from October 25 to October 29, with Dogecoin increasing as much as 46%.[90]

Between April 3 and April 7,[91] 2023, Twitter's bird logo was replaced with an image of the Doge meme for desktop users, leading to a rise in Dogecoin prices.[92] No reason was given for the icon change, with some speculating that it was a late April Fool's joke,[93] or an attempt to troll investors over the Dogecoin lawsuit that Musk was seeking to end that week.[94]

On Noviembre 14, 2024, president-elect Donald Trump announced that Elon Musk and Vivek Ramaswamy would lead a new Department of Government Efficiency or DOGE for short, an acronym that shares the name of Dogecoin. The price of Dogecoin spiked soon after.[1]

DOGE's value is prone to anomalous jumps

Source: https://en.wikipedia.org/wiki/Dogecoin

Why did I choose Dogecoin and autoencoding?

There is no literature!

There is literature on using an autoencoder for anomaly detection on other cryptocurrencies, but not DOGE. This is a newly specific topic of research!

Dogecoin is volatile.

DOGE is inherently prone to anomalies. There is money to be made!

Continued Application.

Our workflow and autoencoder architecture can be applied to any cryptocurrency, financial asset or derivative.

Goals

Develop a robust convolutional autoencoder for anomaly detection at the hour level.

Explore DOGE market behavior.

Identify what drives anomalies.

02

Exploratory Data Analysis

Quick Data Overview

All data can be found on Kaggle.

"This dataset contains minute-level price data for 50 popular cryptocurrencies, obtained using the Binance API. It is a valuable resource for analyzing cryptocurrency markets, developing trading strategies, and creating financial models."

- *Website Description*

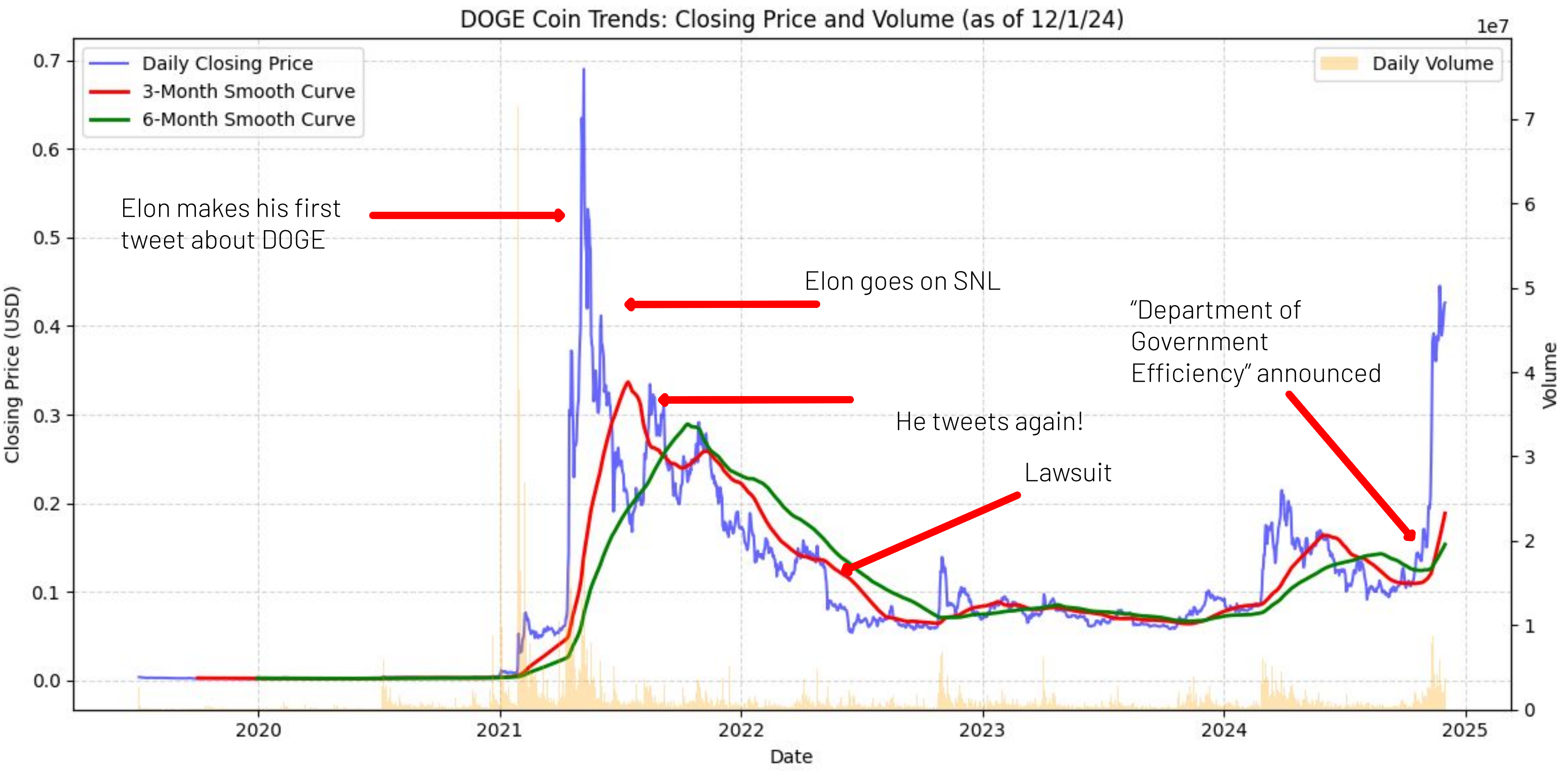
Dataset Summary

This dataset includes minute-level time series data with the following variables:

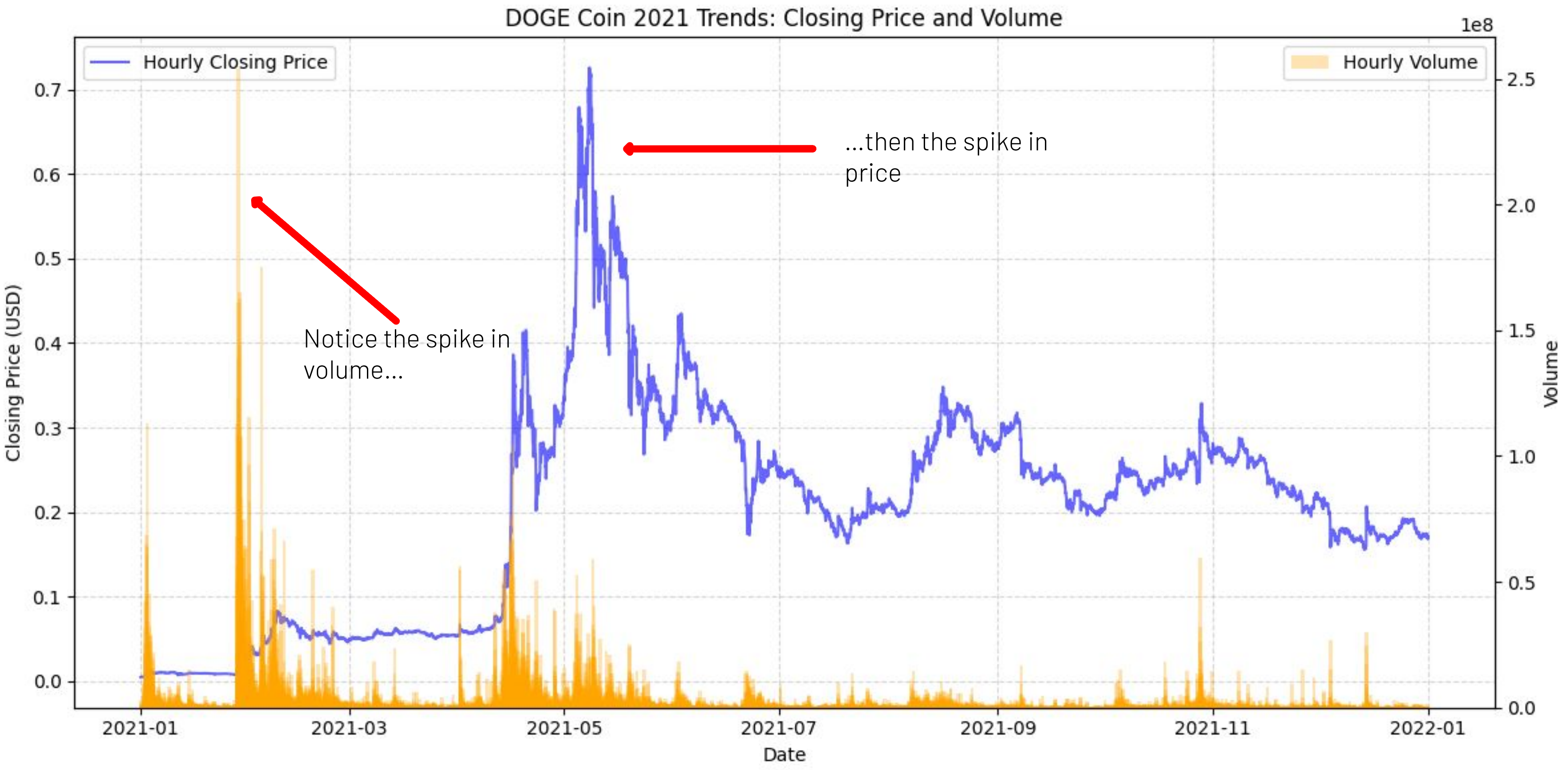
- timestamp: Date and minute (UTC)
- open: Price at the beginning of the minute
- high: Highest price within the minute
- low: Lowest price within the minute
- close: Price at the end of the minute
- volume: Trading volume (amount)
- close_time: Closing timestamp
- quote_asset_volume: Trading volume (value)
- number_of_trades: Number of trades
- taker_buy_base_asset_volume: Taker buy volume (amount)
- taker_buy_quote_asset_volume: Taker buy volume (value)
- ignore: Unused field

Source:https://www.kaggle.com/datasets/kaanxtr/btc-price-1m?resource=download

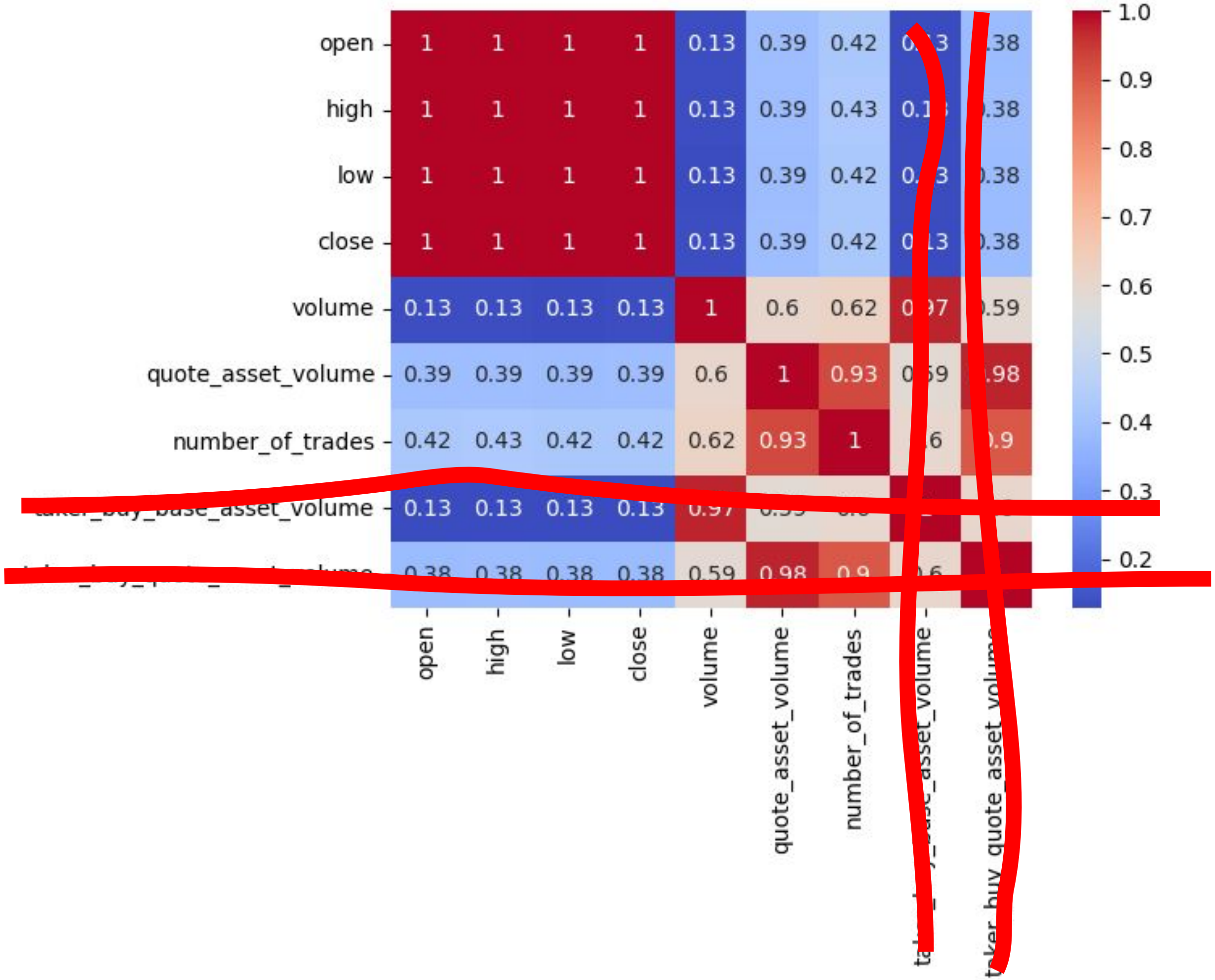
MARKET OVERVIEW



A look at 2021...



Let’s do some feature engineering!

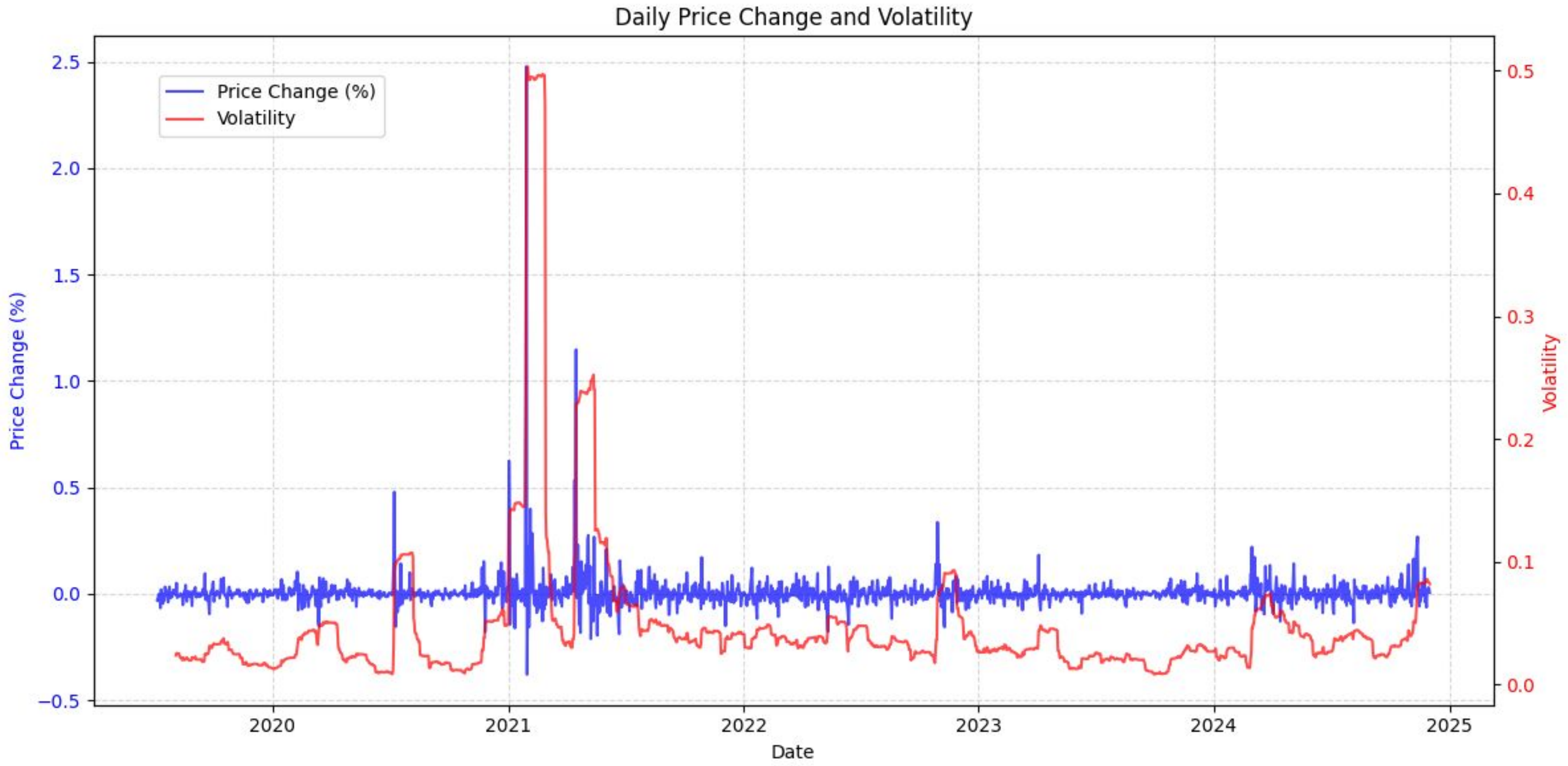

open
high
low
close
volume
quote_asset_volume
number_of_trades
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2

Let's add some variables!

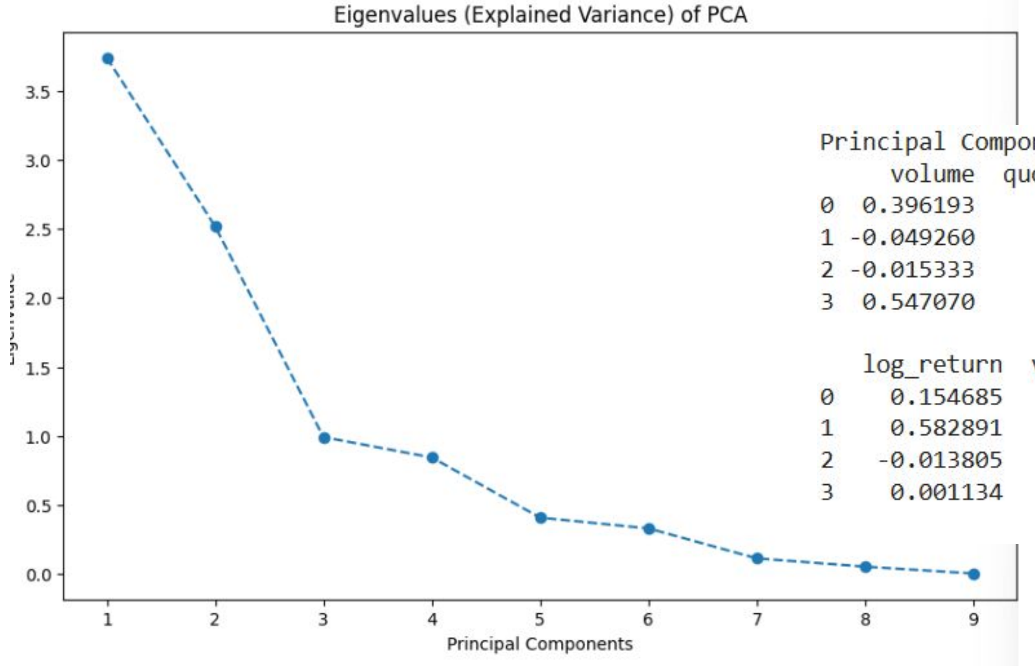
```
# Relative Volume: Volume relative to the moving average 24 hours
# A 24-hour MA adds some stability to volume measure
df['relative_volume'] = df['volume'] / df['volume'].rolling(window=60*24).mean()

# Log Return: Logarithmic difference between current close and previous close
df['log_return'] = np.log(df['close'] / df['close'].shift(1))

# Volatility: Rolling standard deviation of log returns (over a window of 24 hours)
df['volatility'] = df['log_return'].rolling(window=60*24).std()

# Range: Difference between the high and low price for each minute
df['range'] = df['high'] - df['low']

# Change: Difference between current close and previous close
df['change'] = df['close'] - df['close'].shift(1)

# Close to Open Ratio: Ratio of close price to open price
df['close_open_ratio'] = df['close'] / df['open']

# Optional: Drop rows with NaN values caused by rolling window calculations
df.dropna(inplace=True)
```

Behavior, momentum, and relative price movement

The Volatility of DOGE



PCA Time!



Principal Components (First 4):

	volume	quote_asset_volume	number_of_trades	relative_volume
0	0.396193	0.467910	0.475684	0.146174
1	-0.049260	-0.129758	-0.125669	-0.000694
2	-0.015333	-0.015978	0.025469	0.929103
3	0.547070	-0.325104	-0.265265	0.229987

	log_return	volatility	range	change	close_open_ratio
0	0.154685	0.341737	0.437159	0.127841	0.168185
1	0.582891	-0.104122	-0.172922	0.503218	0.575508
2	-0.013805	-0.365234	0.000417	-0.043507	-0.012163
3	0.001134	0.572158	-0.355476	-0.131640	0.026542

PC1: number_of_trades, quote_asset_volume, range, volume

PC2: log_return, close_open_ratio, change

PC3: relative_volume

PC4: volatility, volume, quote_asset_volume

Keep everything!

03

The Autoencoder

A Quick Refresher

Definition: An autoencoder is a type of artificial neural network used for unsupervised learning, primarily for dimensionality reduction and feature learning. It learns to encode input data into a compressed representation and then decode it back to the original input.
Structure: Consists of two main parts:

- **Encoder**: Compresses the input into a lower-dimensional latent space (bottleneck).
- **Decoder**: Reconstructs the original input from the compressed representation.

Objective: Minimize the difference between the input and the reconstructed output (usually via mean squared error).

Types of Autoencoders

1. **Vanilla Autoencoder**: Basic form used for simple data compression and reconstruction.
2. **Convolutional Autoencoder**: Uses convolutional layers, ideal for image data where spatial relationships are important.
3. **Variational Autoencoder (VAE)**: A generative model that learns a distribution over the latent space, enabling sampling of new data.
4. **Denoising Autoencoder**: Trains to reconstruct clean data from noisy input, helping in data noise reduction.
5. **Sparse Autoencoder**: Enforces sparsity constraints on the activations to encourage learning of more useful, efficient features.

Applications of Autoencoders

- **Dimensionality Reduction**: Reducing the number of features for faster and more efficient machine learning.
- **Anomaly Detection**: Identifying outliers by measuring reconstruction error.
- **Image Denoising**: Removing noise from images by learning clean representations.
- **Data Compression**: Compressing data (e.g., images, audio) while retaining essential information.
- **Generative Models**: Creating new data similar to the input data (e.g., VAE generating new images or texts).

Data Structure and Split

Remember! 2021 and 2024 seem like they contain “anomalous” market behavior

Train Data: 2019 through 2020
Validation Data: 2021
Test Data: 2024

```
Train shape: (30553, 60, 9)
Validation shape: (8743, 60, 9)
Test shape: (8043, 60, 9)
```

We will also normalize all dat

```
def normalize_data(data, mean, std):
    return (data - mean) / std
```

Model Architecture

Input Layer: Accepts sequences of shape `(seq_length=60, n_features=9)`.

Encoder:

- **LSTM Layer**: Single LSTM with 64 units and `tanh` activation. It processes the input sequence into a single fixed-size vector representation, capturing temporal dependencies.

Bottleneck:

- **RepeatVector**: Replicates the encoded representation to match the original sequence length (60 timesteps).

Decoder:

- **LSTM Layer**: Single LSTM with 64 units and `tanh` activation, reconstructing sequences from the bottleneck representation.
- **TimeDistributed Dense Layer**: Outputs predictions for each timestep, matching the feature dimensions of the input (`n_features=9`).

Loss Function: Mean squared error (MSE), optimized using Adam.

Sequential Context: The architecture processes the sequential order of data, preserving relationships across time steps.

Noise Resilience: LSTMs are robust to noisy data and can focus on meaningful patterns, which is valuable in volatile cryptocurrency markets.

The LSTM autoencoder compresses the input sequence of shape `(60, 9)` into a latent vector of size `(64)` via the encoder, repeats it across 60 timesteps in the bottleneck, and reconstructs it back to the original shape `(60, 9)` through the decoder.

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 60, 9)	0
lstm (LSTM)	(None, 64)	18,944
repeat_vector (RepeatVector)	(None, 60, 64)	0
lstm_1 (LSTM)	(None, 60, 64)	33,024
time_distributed (TimeDistributed)	(None, 60, 9)	585

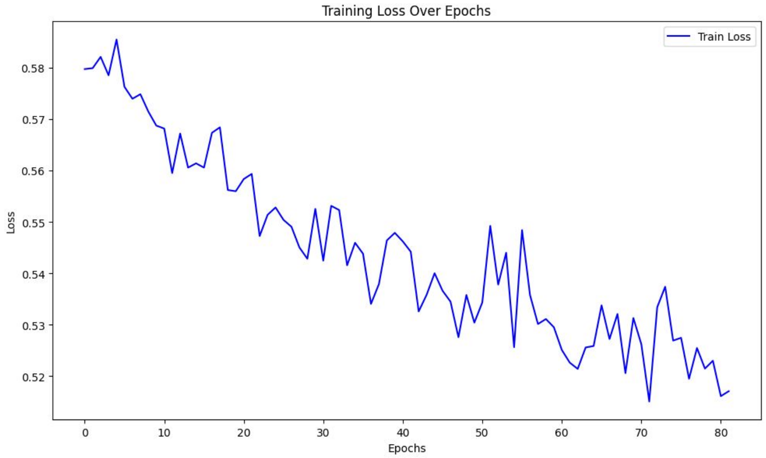
Total params: 52,553 (205.29 KB)
Trainable params: 52,553 (205.29 KB)
Non-trainable params: 0 (0.00 B)

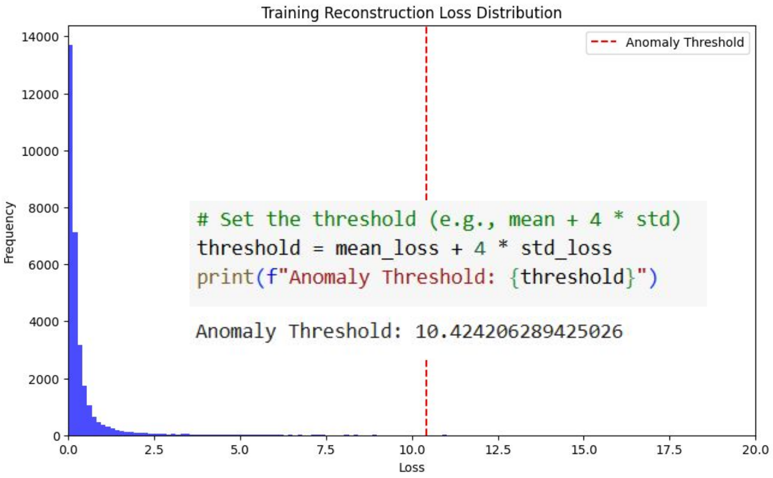
Training Loop

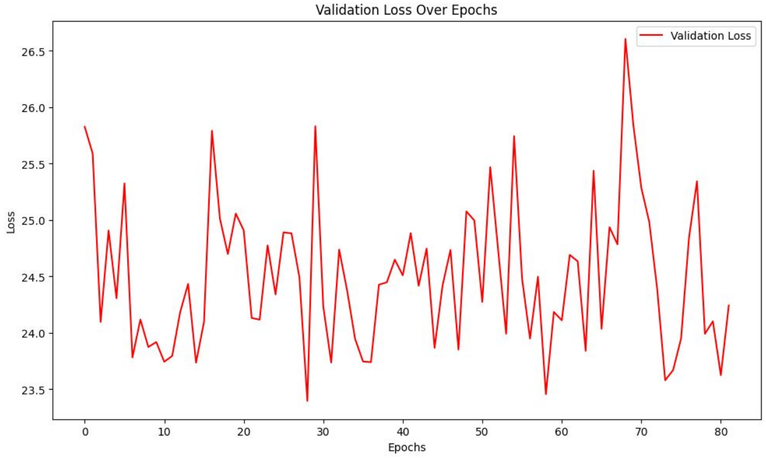
```
# Early stopping based on training loss with a minimum delta of 0.001
early_stopping = EarlyStopping(monitor='loss',  # Monitor the training loss
                               patience=10,     # Patience of 10 epochs
                               min_delta=0.001,  # Stop if the loss doesn't improve by 0.001
                               restore_best_weights=True)  # Restore the best weights when stopping



# Train the model
history = model.fit(X_train_normalized, X_train_normalized, epochs=200, batch_size=64,
                    validation_data=(X_val_normalized, X_val_normalized), callbacks=[early_stopping])

# Save the model
model.save("lstm_autoencoder.h5")
```

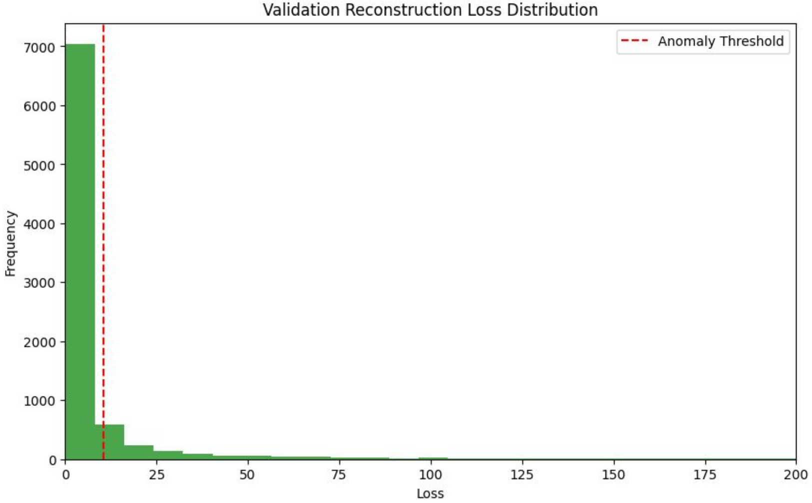
04

Results

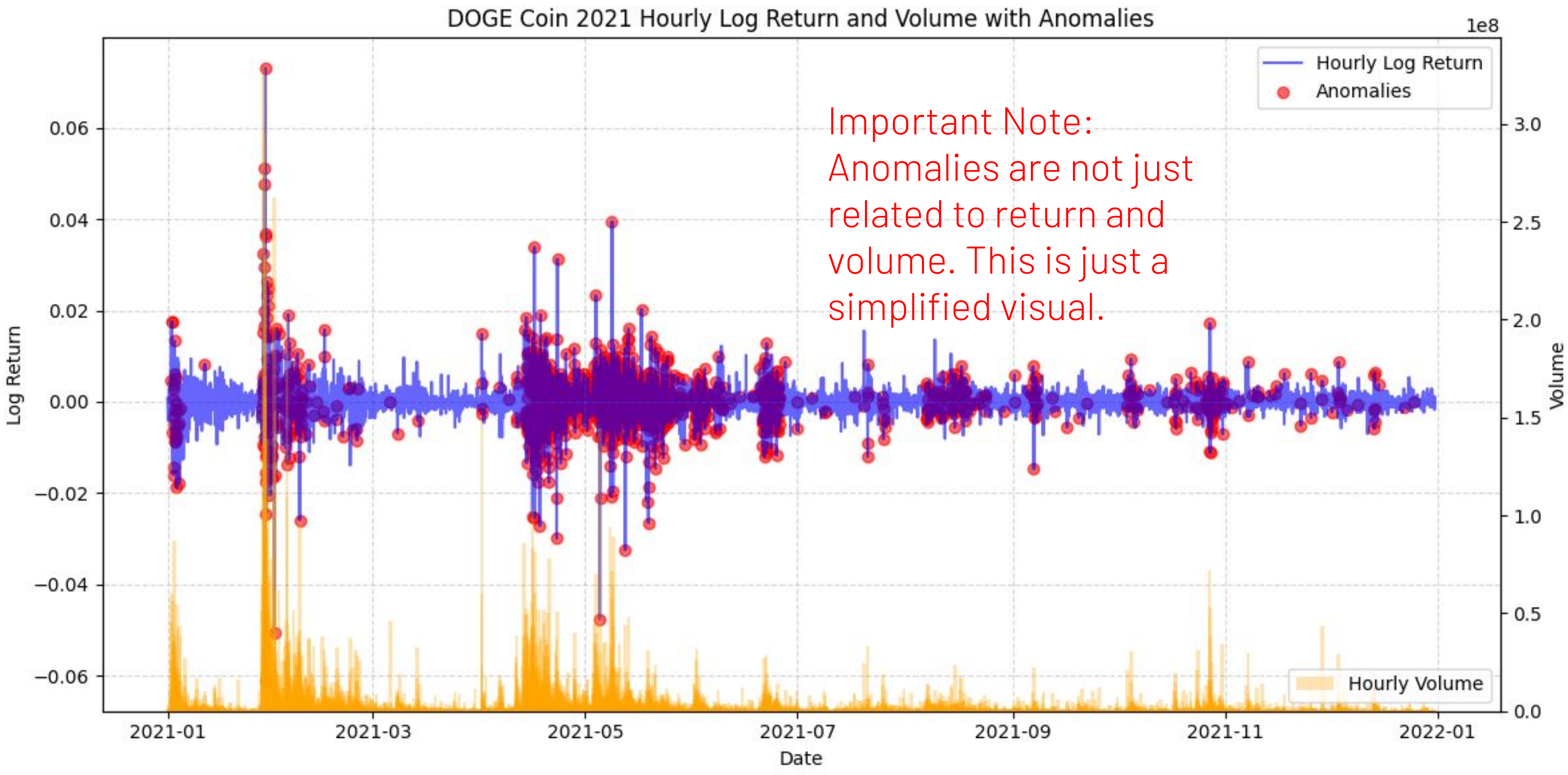

Training Loss Over Epochs
Train Loss
Loss
Epochs


Training Reconstruction Loss Distribution
Anomaly Threshold
Frequency
Loss
# Set the threshold (e.g., mean + 4 * std)
threshold = mean_loss + 4 * std_loss
print(f"Anomaly Threshold: {threshold}")
Anomaly Threshold: 10.424206289425026

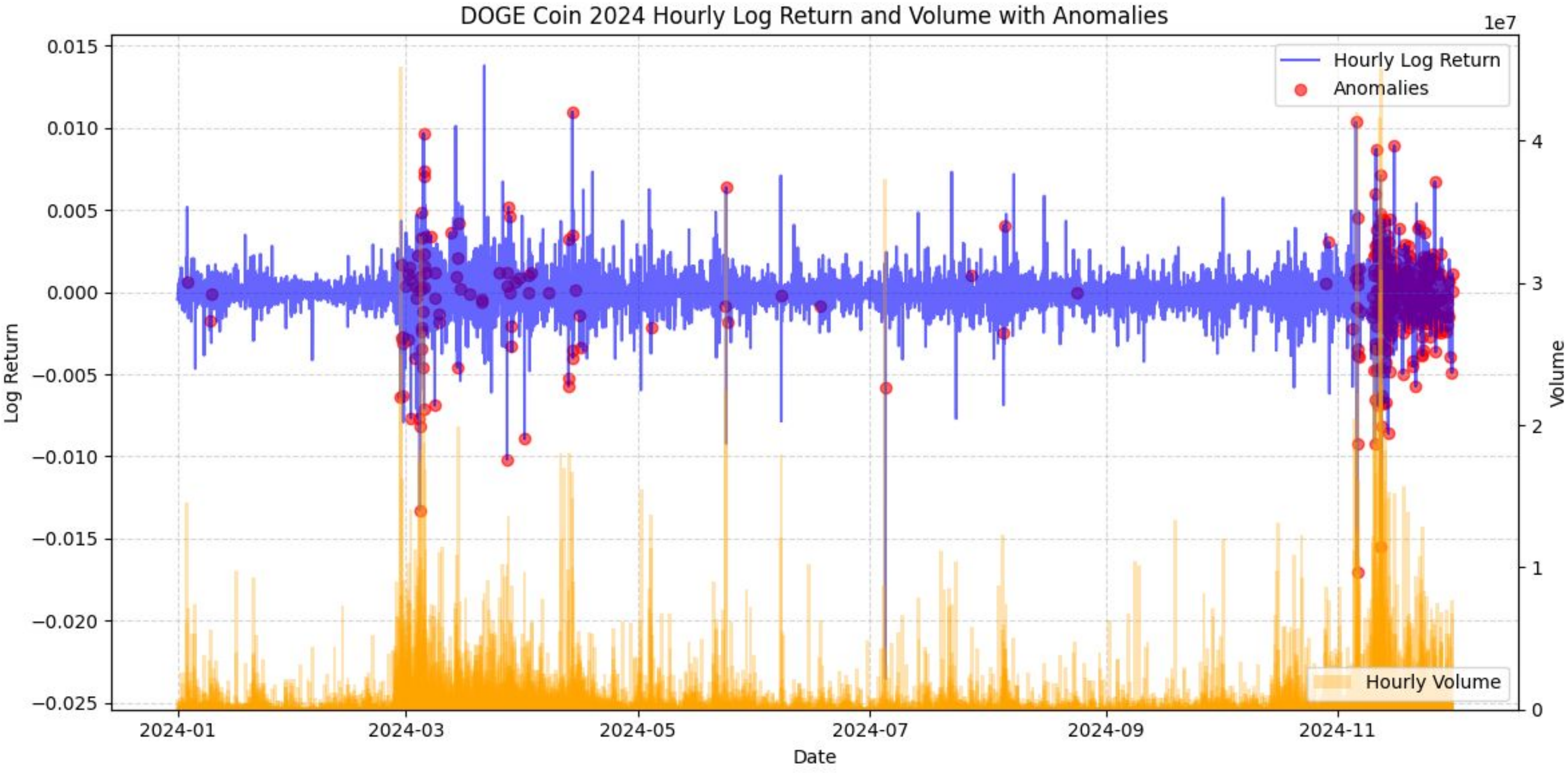

Validation Loss Over Epochs
Validation Loss
Loss
Epochs


Validation Reconstruction Loss Distribution
Anomaly Threshold
Frequency
Loss

Out of the 8,743 hours in 2021, we identified **1,1718** of those hours to present anomalous market activity.



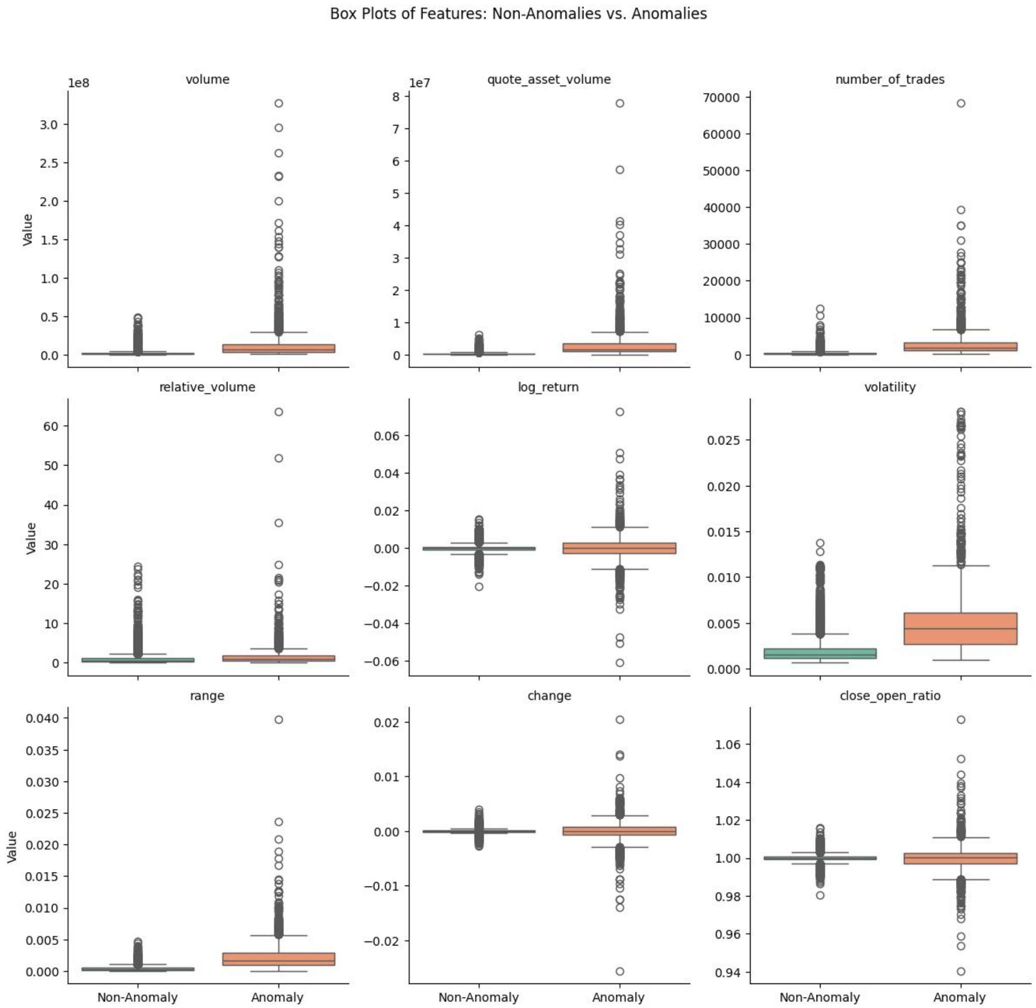
Out of the 8,043 hours in 2021, we identified **398** of those hours to present anomalous market activity.



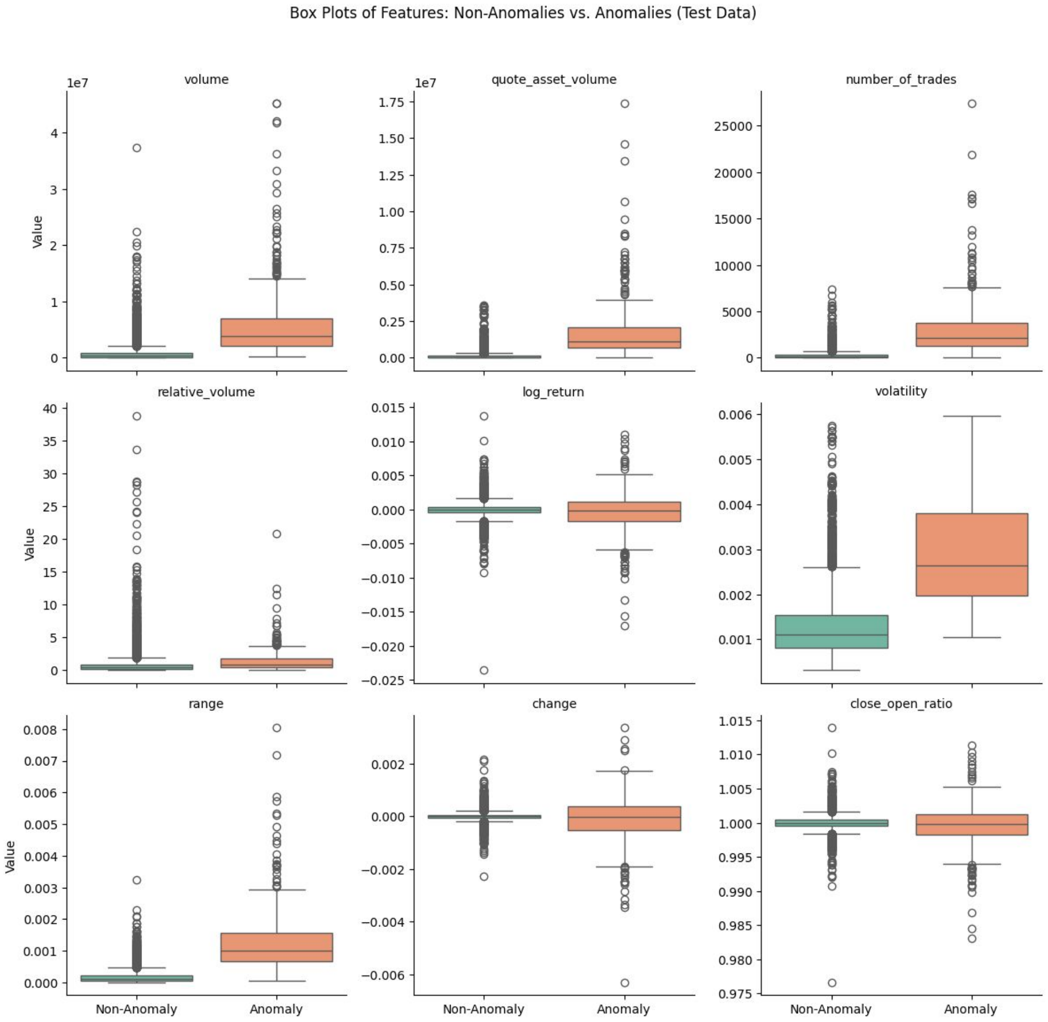
What is driving this anomalous activity?

Looking at 2021...

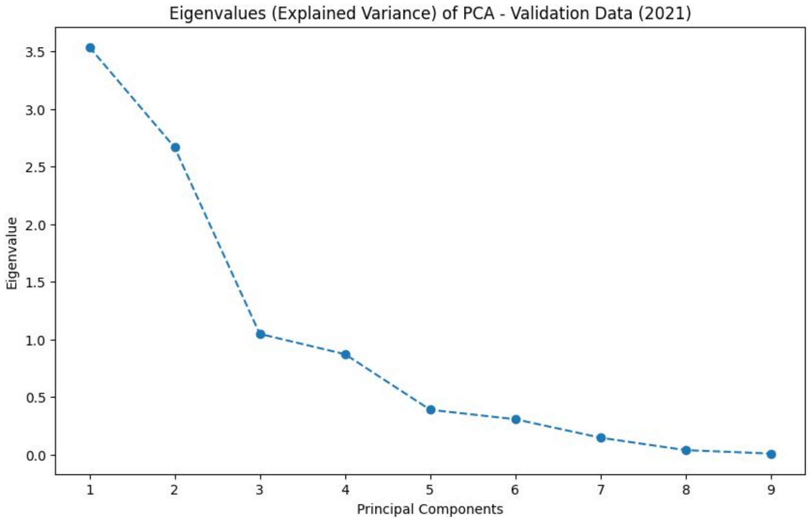
Notice the higher median and spreads for anomalies!

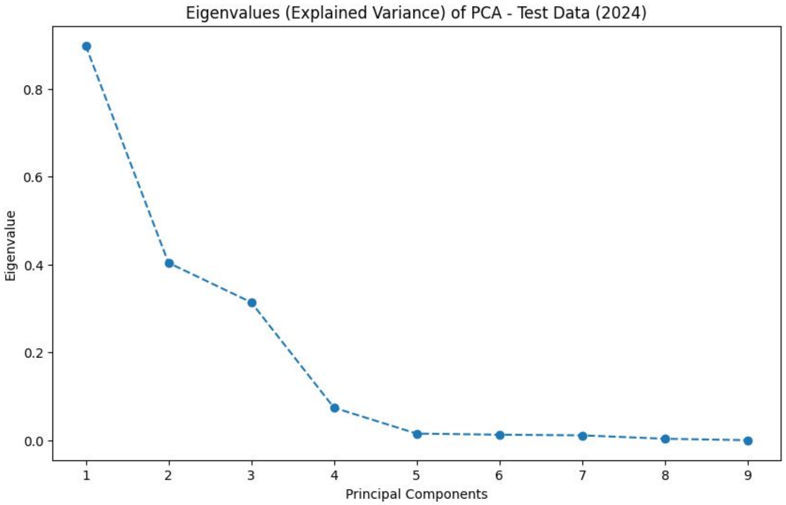


Looking at 2024...



You already know what time it is....PCA Time!






```
Principal Components (First 4) - Validation Data (2021):
     volume  quote_asset_volume  number_of_trades  relative_volume  \
0  0.389763            0.497636          0.503762         0.231370
1  0.079782            0.003690          0.006859         0.035485
2  0.217236           -0.102074         -0.111246        -0.632849
3  0.518184           -0.276444         -0.211792         0.615338

   log_return  volatility     range    change  close_open_ratio
0   -0.016210    0.280472  0.459861 -0.050501         -0.013617
1    0.596244    0.032091 -0.019740  0.529548          0.595791
2   -0.014957    0.723432 -0.075994 -0.004502         -0.011999
3   -0.038425    0.268318 -0.390804 -0.065699         -0.032440
Principal Components (First 4) - Test Data (2024):
     volume  quote_asset_volume  number_of_trades  relative_volume  \
0  0.163459            0.169264          0.301487         0.905203
1  0.163010            0.303487          0.549948        -0.388926
2  0.106306            0.188813          0.316697        -0.123530
3  0.012351           -0.144712         -0.373791         0.090645

   log_return  volatility     range    change  close_open_ratio
0   -0.052607    0.051222  0.156437 -0.041559         -0.052466
1   -0.293354    0.283239  0.335635 -0.249592         -0.293920
2    0.547328    0.205462  0.194924  0.391734          0.551596
3   -0.037396    0.909583  0.020585  0.016961         -0.038955
```

2021/Valdiation Data

3-4 principal components

PC1: number_of_trades, quote_asset_volume, range, *volume*

PC2: log_return, close_open_ratio, change

PC3: volatility, - relative volume

(**PC4**: relative volume, volume)

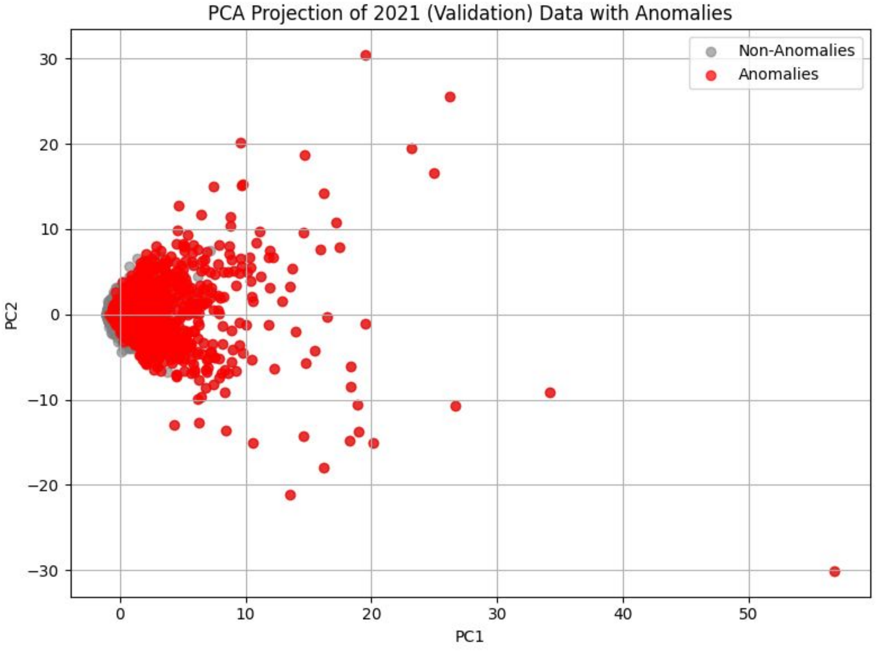
2024/Test Data

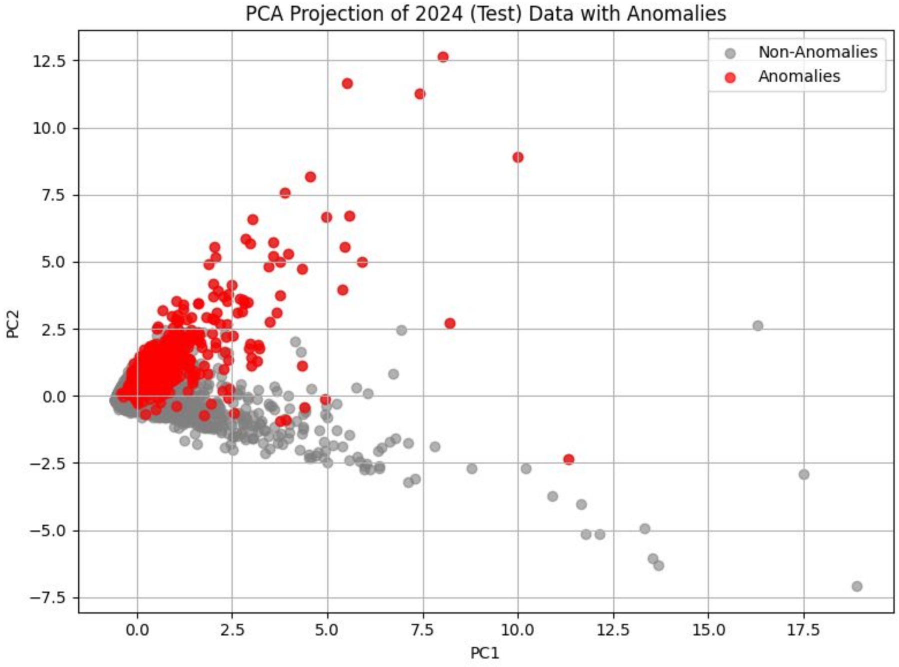
1 principal component

PC1: relative volume (explains most of it all), *number of trades*

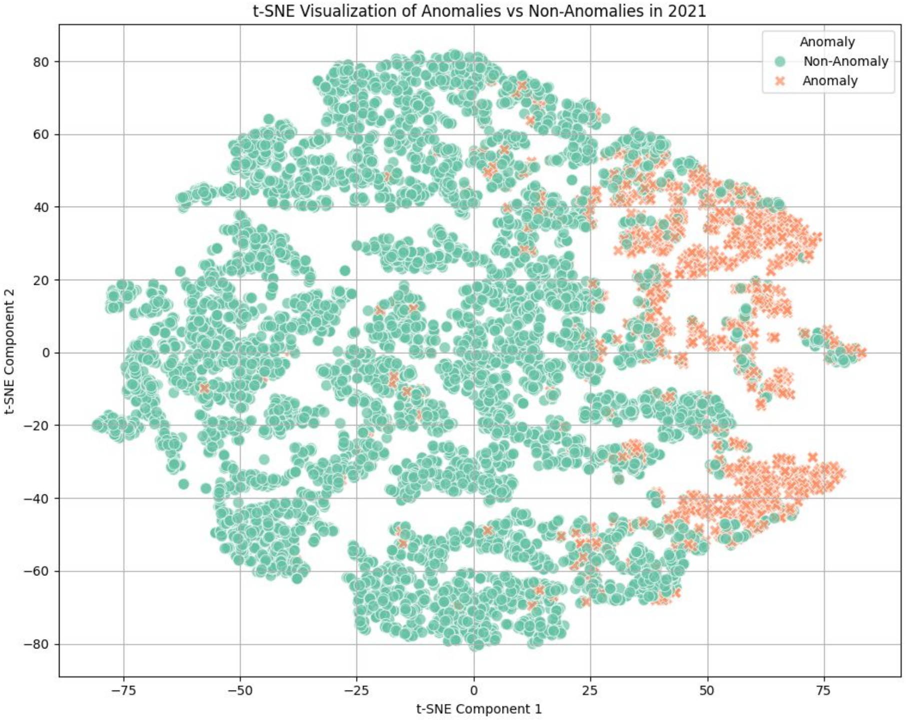
Visualizing our PCA

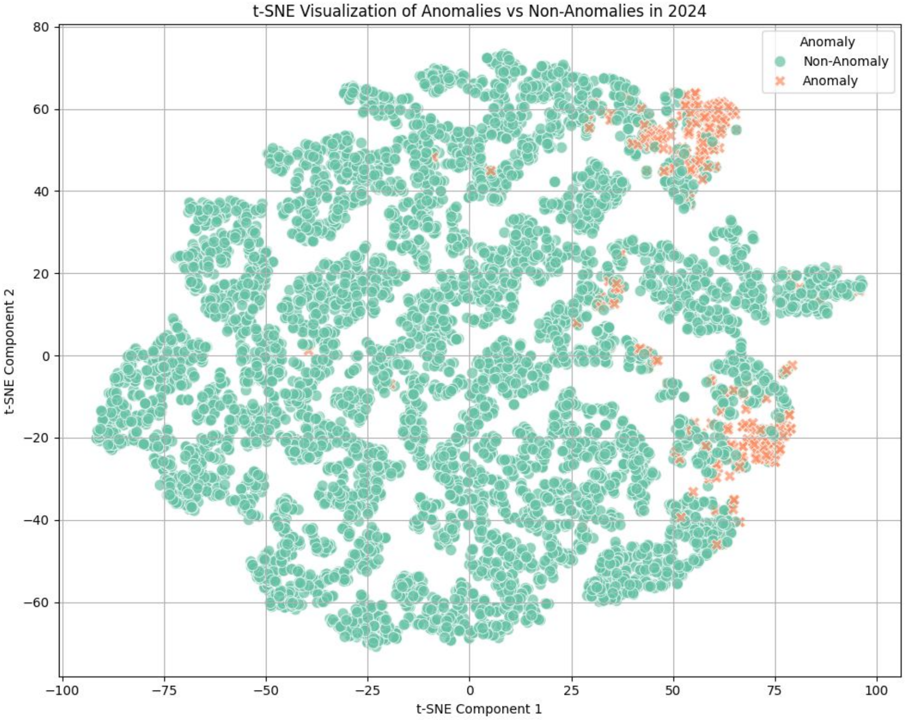
You can really see the first 2 PCs driving that separation between anomalies and non-anomalies in 2024





Why not some tSNE?





t-SNE has successfully captured meaningful differences in their high-dimensional representation!

05

Closing

Closing

Recap	• We developed a convolutional autoencoder wit LSTM in order to create an anomaly detection system for DOGE • We also explored various market behaviors for DOGE
Main Findings	• 2021 and 2024 indeed presented a lot of anomalous activity • 2021 presented way more anomalous volatility than 2024's bull rush • Volume is the most indicative, log return follows
Future Applications	• Immediate: Anomaly alert message system • Mid-term: Create a DOGE buy system • Long-term: Apply to other cryptocurrencies and financial assets
Improvements	• Develop a more complex thresholding system • Develop a more intent way of splitting data • Look more at price movements before and after anomaly detection

THANK YOU

Sources

Inzirillo, H., & De Villelongue, L. (2023, April 20). An attention free conditional Autoencoder for anomaly detection in cryptocurrencies. arXiv.org. https://arxiv.org/abs/2304.10614

Data Source:
https://www.kaggle.com/datasets/kaanxtr/btc-price-1m?resource=download

Relevant Market info:
https://coinmarketcap.com/currencies/dogecoin/

Questions?