🚀 CRYPTO PUMP DETECTION SYSTEM 🚀

From Signals to Machine Learning

The Detection Pipeline

Goal: Detecting coordinated crypto pump-and-dump events by cross-referencing Telegram "Signals" with Binance "Market Reality."

📱 **Telegram Intent** + 💹 **Market Execution** = 🤖 **ML Detection**

📱 Source #1: The Telegram Registry

Concept: Ground Truth (Intent)

This is where pump organizers announce their plans. It tells us **WHERE** and **WHEN** a pump was planned.

Data Template Example:

Symbol	Group	Date	Hour	Exchange
BNT	BPS	2021-01-13	16:00	binance
XRP	CryptoKings	2021-01-15	14:30	binance
DOGE	MoonShot	2021-01-18	19:00	binance

🎯 **Purpose: Establishes the PLANNED pump timeline**

Concept: Raw Market Data (Execution)

This shows what ACTUALLY happened in the market. We match this to the Telegram Registry to create a "Pump Index."

Data Template Example:

Timestamp	Side	Price	Amount	BTC Volume
2021-01-13 16:00:00.123	buy	0.00045	15000	6.75
2021-01-13 16:00:00.123	buy	0.00046	22000	10.12
2021-01-13 16:00:00.123	buy	0.00047	18500	8.70
2021-01-13 16:00:00.456	buy	0.00048	30000	14.40

🔍 **Key Insight: Multiple "buy" orders in the SAME MILLISECOND = Bot Activity!**

🔧 Feature Engineering Logic

⏱️ Time Chunking

Raw trades are grouped into **15-second "chunks"** to create manageable analysis windows.

```
chunk_duration = 15s
total_chunks = trade_history / 15s
```

📊 Rolling History

We compare the current chunk to the **last 900 chunks** to detect abnormal deviations.

```
lookback_window = 900 chunks
= 3.75 hours of history
```

Why This Matters:

Normalization: Converts chaotic tick data into structured features

Context: Establishes "normal" vs "abnormal" behavior baseline

Precision: 15-second windows catch rapid pump execution

🎯 **Result: Clean, time-series features ready for ML training**

Intelligence)

Feature Name	What It Detects	Why It Matters
std_rush_order	Bots buying at exact same millisecond	Coordinated automated attacks
std_volume	Whale activity & sudden money injections	Large capital movements = manipulation
avg_price_max	Price ceiling being pushed higher	Artificial price inflation signal
hour_sin / hour_cos	Mathematical time transformation	Helps model learn optimal pump timing

🎓 The Math Behind hour_sin/cos:

Instead of treating "23:59" and "00:01" as completely different (they're only 2 minutes apart!), we use sine and cosine to create a circular representation of time. This helps the model understand that pumps often happen at similar times of day.

💡 **These features transform raw chaos into mathematical patterns**

Concept: Labeled Dataset Creation

Every **15-second window** is labeled with:

gt=1 if a pump is happening (based on Telegram Registry match)

gt=0 if normal trading activity

Example Training Data:

Timestamp	std_rush_order	std_volume	avg_price_max	hour_sin	hour_cos	gt (Label)
16:00:00	0.89	12.5	0.00048	0.707	-0.707	1
16:00:15	0.92	15.8	0.00052	0.707	-0.707	1
12:30:00	0.12	2.3	0.00041	0.259	0.966	0
08:15:00	0.08	1.9	0.00040	-0.259	0.966	0

🎯 **Goal: This creates a clean dataset for the Random Forest classifier to learn patterns**

What We've Built:

✅ Cross-referenced Telegram signals with real market data

✅ Engineered features that detect bot behavior and whale movements

✅ Created labeled training data for ML classification

✅ Built a Random Forest model to predict pump events in real-time

🚀 Future Enhancements:

Twitter/X Sentiment Analysis: Detect social media hype before pumps

Order Book Depth: Analyze bid/ask walls for manipulation signals

Network Analysis: Map connections between pump groups

Real-Time Alerts: Deploy model for live monitoring and notifications

Multi-Exchange Support: Expand beyond Binance to catch cross-platform pumps

🎯 From Signals to Intelligence 🤖

Thank You! Questions? 💬