

## # Kevin: AI-Powered Crypto Trading Bot Builder

### ## Overview

Kevin is an automated, modular, deep learning (DL) and neural network (NN)-powered system designed to create, backtest, optimize, and deploy cryptocurrency trading bots with a single click. Built for Ubuntu, Kevin leverages AI to streamline the development of profitable trading strategies, focusing on BTCUSDT perpetual futures from Binance. It uses historical data for backtesting and GPU acceleration for efficient training and inference, making it ideal for traders seeking passive income through automation.

### ## What Kevin Does

- **Designs Trading Strategies**: Uses deep neural networks (e.g., LSTM) to generate trading signals based on market data.
- **Generates Trading Bots**: Automatically writes Python code for bots tailored to the generated strategies.
- **Backtests Strategies**: Tests bot performance against historical BTCUSDT Futures data.
- **Optimizes Performance**: Employs deep reinforcement learning (RL) to refine strategies until profitable.
- **Deploys Bots**: Launches optimized bots for live or paper trading on Binance.
- **Automates Everything**: Executes all steps with one command, minimizing manual intervention.

### ## Target Audience

- Beginner traders wanting an easy, one-click solution.
- Advanced traders seeking customizable, AI-driven automation.
- Developers interested in a modular framework for crypto bot experimentation.

### ## Core Technologies

- **Python**: Primary language for scripting and bot logic.
- **PyTorch**: Deep learning framework for neural networks (LSTM).
- **Stable-Baselines3**: Reinforcement learning (PPO) for optimization.
- **CCXT**: Library for Binance API integration.
- **NVIDIA CUDA/cuDNN**: GPU acceleration for training and inference.
- **Pandas**: Data handling for historical BTC Futures data.

### ## Features

#### ### 1. Modular Architecture

- **Why**: Allows independent development, testing, and scaling of components.
- **Components**:
  - `dl_nn_strategy_generator.py`: Generates trading strategies using an LSTM neural network.
  - `dl_nn_code_generator.py`: Writes bot code with embedded AI logic.
  - `dl_nn_backtester.py`: Tests bots against historical data.
  - `dl_nn_optimizer.py`: Optimizes strategies with deep RL (PPO).
  - `dl_nn_deployer.py`: Deploys bots for live trading.
  - `dl_nn_trainer.py`: Trains the LSTM model on historical data.
  - `utils.py`: Shared utilities (data loading, logging).

#### ### 2. Deep Learning and Neural Networks

- **LSTM Model**: Predicts BTCUSDT price trends based on 50-candle sequences (OHLCV data).
- **GPU Support**: Accelerates training and inference using NVIDIA CUDA.
- **Prediction**: Outputs a threshold for buy signals (e.g., if prediction > threshold, buy).

### ### 3. Historical Data Integration

- **Source**: BTCUSDT Futures 1-minute Kline data from data.binance.vision.
- **Usage**: Backtesting and training use local CSV data (e.g., BTCUSDT-1m-2024-12.csv).
- **Automation**: Downloaded and extracted via setup script.

### ### 4. One-Click Automation

- **Execution**: Running `python3 main.py` triggers the full pipeline: train, generate, backtest, optimize, deploy.
- **No Manual Steps**: Fully automated from setup to deployment.

### ### 5. Risk Management

- **Risk Percent**: Configurable (default 0.5% of account balance per trade).
- **Profitability Check**: Only deploys bots if backtest profit > 0.

### ### 6. Logging

- **Location**: `logs/kevin.log`.
- **Purpose**: Tracks backtest results, deployment status, and errors.

### ### 7. Binance Integration

- **API**: Uses CCXT to connect to Binance for live trading.
- **Configurable**: API keys stored in `config.yaml`.

## ## Directory Structure

```
kevin/
├── main.py # One-click entry point
├── modules/ # Modular components
│   ├── dl_nn_strategy_generator.py
│   ├── dl_nn_code_generator.py
│   ├── dl_nn_backtester.py
│   ├── dl_nn_optimizer.py
│   ├── dl_nn_deployer.py
│   ├── dl_nn_trainer.py
│   ├── utils.py
│   └── init.py
├── bots/ # Generated bot scripts (e.g., dl_nn_trading_bot.py)
├── data/ # Historical BTCUSDT data (e.g., BTCUSDT-1m-2024-12.csv)
├── models/ # Trained models (e.g., dl_nn_predictor.pt)
├── logs/ # Log files (e.g., kevin.log)
├── config.yaml # Configuration settings
├── requirements.txt # Python dependencies
└── setup_kevin.sh # Setup script
```

## ## Prerequisites

- **OS**: Ubuntu (tested on 20.04/22.04).
- **Hardware**: NVIDIA GPU (e.g., GTX 1060 or better) for CUDA acceleration.

- **Software**:
  - Python 3.8+
  - NVIDIA Drivers, CUDA Toolkit (e.g., 11.8), cuDNN (e.g., 8.9)
  - wget, unzip (for data download)
- **Binance Account**: API keys required for live trading.