**Report on GAN Implementation for Stock Prediction**

Introduction:

For the task B7 of Comprehensive Extension and Independent Research, I am seeking to propose a novel approach for stock prediction utilizing Generative Adversarial Networks (GAN) with components of Gated Recurrent Units (GRU) and Convolutional Neural Networks (CNN). I aim to address the limitations of traditional stock prediction models by leveraging the power of GANs to generate more accurate and stable predictions for multi-step ahead stock prices. The novelty lies in the integration of GAN architecture, which is primarily used in image generation tasks, into the realm of financial time series forecasting.

Background:

Traditionally, LSTM and GRU networks have been extensively utilized in stock price prediction due to their ability to capture long-term dependencies in sequential data. However, these models often suffer from issues such as vanishing gradients and difficulty in capturing complex patterns in financial time series data. GANs offer a promising alternative by introducing a competitive learning framework where a generator network generates synthetic data samples, and a discriminator network distinguishes between real and fake samples. This adversarial training process encourages the generator to produce more realistic data, which can be beneficial for improving prediction accuracy in financial markets. This approach also enroots ideally from Reinforcement Learning, one of the strongest most capable machine learning methods recently.

Methodology:

The methodology employed involves the construction of a GAN model with a GRU-based generator and a 1D-CNN discriminator. The data preprocessing steps involve standardization and normalization of the input features, followed by feature engineering to extract relevant information from the raw data. Hyperparameter tuning is conducted using Bayesian Optimization to optimize the performance of the GAN model. The generator aims to generate future stock price sequences, while the discriminator evaluates the authenticity of the generated sequences compared to real stock price data. The code implementation for this whole process of methodology is attached to the sbmission of this document.

**Tools clarification:**

technical\_generator.py:

Each method in this Generator class takes data as input, which would be a Pandas Series or DataFrame containing stock prices, and windows which is the period over which the calculations are performed. The methods return the calculated indicator values. Remember to use these indicators as part of a comprehensive trading strategy, as they are not foolproof predictions of market behavior.

VAE.py:

This class stands for Variational Autoencoder, using the PyTorch library. A Variational Autoencoder is a type of neural network that aims to learn a compressed representation (encoding) of its input data and then, from this representation, generate new data that is similar to the original input (decoding).

The VAE can be used for tasks like data denoising, anomaly detection, and generative modeling, where you want to create new data points that resemble your training data.

generator.py:

The Generator class attached is designed for sequential data processing, potentially for generating new sequences that mimic some training data, given the recurrent nature of the GRUs and the binary output from the sigmoid function. The exact use case is further specified in the gan\_demo part, for data generation.

discriminator.py:

The Discriminator is designed to distinguish between real and generated (fake) data. This used in the gan\_demo.py setup where it needs to output a probability that the input data is real. If self.sig is True, the output is in the form of a probability, suitable for binary classification. If self.sig is False, the raw score from the last linear layer is returned, which might be used in certain GAN setups where the sigmoid function is applied later or not at all.

TaskB7.ipynb: (data preprocessing and preparation for gan\_demo.ipynb)

The columns being added to the dataset are various technical indicators used in stock market analysis. These indicators are designed to help predict future price movements based on historical data. Here's why each type of indicator is added:

• Moving Averages (MA): The 7ma, 14ma, and 21ma columns represent the Exponential Moving Average (EMA) for 7, 14, and 21 days, respectively. EMAs are used to smooth out price data over a specified time period and are more responsive to recent price changes than simple moving averages.

• Moving Average Convergence Divergence (MACD): The 7macd and 14macd columns are MACD indicators, which are used to identify changes in the strength, direction, momentum, and duration of a trend in a stock's price.

• Relative Strength Index (RSI): The 7rsi, 14rsi, and 21rsi columns calculate the RSI for 7, 14, and 21 days. RSI is a momentum oscillator that measures the speed and change of price movements, often used to identify overbought or oversold conditions.

• Average True Range (ATR): The 7atr, 14atr, and 21atr columns represent the ATR for 7, 14, and 21 days. ATR measures market volatility by decomposing the entire range of an asset price for that period.

• Bollinger Bands: The 7upper, 7lower, 14upper, 14lower, 21upper, and 21lower columns are Bollinger Bands for 7, 14, and 21 days. These are volatility bands placed above and below a moving average, where the bandwidth expands and contracts based on volatility.

• Raw Stochastic Value (RSV): The 7rsv, 14rsv, and 21rsv columns calculate the RSV for 7, 14, and 21 days. RSV is used in the stochastic oscillator, which is a momentum indicator comparing a particular closing price of a security to a range of its prices over a certain period of time.

Adding these indicators to the dataset can provide valuable insights into market trends and help in building predictive models for stock prices. Each indicator offers a different perspective on the market's behavior, and when used together, they can form a comprehensive analysis toolset.

Results:

Experimental results demonstrate that the proposed GAN model outperforms traditional models like LSTM and GRU in terms of prediction accuracy. The Root Mean Squared Error (RMSE) values indicate that the GAN model produces more accurate predictions for multi-step ahead stock prices. Additionally, the GAN model exhibits greater stability in predictions, making it a promising approach for real-world stock market forecasting tasks.

Conclusion:

In conclusion, a novel GAN-based approach for stock prediction, it shows promising results compared to traditional models. The integration of GAN architecture with GRU and CNN components offers improvements in accuracy and stability for multi-step ahead predictions. Future research directions could focus on further optimizing hyperparameters and exploring reinforcement learning techniques to enhance the performance of the proposed model.

References:

Lin, H. C., Chen, C., Huang, G. F., & Jafari, A. (2021)1. Stock price prediction using Generative Adversarial Networks2. Journal of Computer Science, 17(3), 188-196. <https://doi.org/10.3844/jcssp.2021.188.196>