**Preethi  
Hello everyone, I'm Preethu, and I am excited to introduce Group 5's project titled "** **Cryptocurrency Prediction Using Fin-BERT and Time-Series Methods."**  
As we all know, accurate crypto market forecasting is critical for investors and analysts alike. With the volatility of modern markets, the ability to predict crypto currency movements can directly impact investment strategies and risk management.  
Through this project, our aim is to harness cutting-edge deep-learning models to deliver precise and actionable insights.

Our goal is to explore the potential of LSTM and FinBERT, a transformer-based model, to improve stock prediction accuracy by leveraging the latest advancements in natural language processing and deep learning.

**Slide 2: Motivations  
Coming to the motivation of our project we would like to mention 3 main points**

**1)Navigating Market Volatility:**  
The crypto market is driven by a wide range of factors—both predictable and unpredictable. From economic indicators to geopolitical events, market volatility presents constant challenges for investors. Markets are inherently **uncertain**, and this unpredictability makes effective **risk management** essential.

**2)Risk Reduction and Profit Maximization:**  
Managing risk strategically and at the same time gaining maximum profits is the main goal of the investors. By tapping into LSTM and more advance models like **Fin-BERT**, which is specifically fine-tuned for financial data, we aim to bring a new level of sophistication to crypto price forecasting leveraging financial news.

**3)Limitations of Traditional Models (Holt-Winters & ARIMA):**  
Traditional statistical models, such as **Holt-Winters** and **ARIMA**, have long been used in financial forecasting. They primarily rely on **historical data** and **linear relationships**, capturing trend, seasonality etc. However, these models have significant limitations.   
For instance, traditional models may perform well in stable environments but often **fail to respond** quickly to sudden changes, such as shifts in public sentiment, market disruptions, or geopolitical shocks. These external factors, which heavily influence crypto prices, are hard to predict using conventional approaches.

**Next coming to the research questions, throughout this project, we will address the following:**

**First Question: How can news data be incorporated into predictive models for accurate forecasting?**To do this, we rely on word embeddings—they capture the meaning and context of words in financial news articles. By transforming text into numerical vectors, we can incorporate sentiment and context into our models, allowing us to forecast crypto movements based on relevant news**.**

**How RECUURENT NEURAL NETWORKS LIKE LSTM IS USED IS USED TO PREDICT CRYPTO PRICES FROM FINANCIAL NEWS?**

LSTMs can maintain information over time, making them ideal for capturing the sequential nature of financial news. However, they have limitations—specifically, they struggle with long-term dependencies and use static vector embeddings, which means the context of words doesn’t change with different meanings.

**While LSTM is a powerful model for our project, are there any models that overcome its drawbacks?  
Yes, this brings us to the next question.**

**How can transformer models LIKE BERT SUITABLE for text generation be adopted for numerical CRYPTO PRICE forecast?**

Transformer models introduce attention mechanisms, which allow them to focus on relevant parts of the input, addressing the long-term dependency issue. Unlike LSTMs, transformers can capture relationships in the data without needing sequential order, making them much more efficient for understanding complex text like financial news.

**Lastly, How SEMANTIC SEARCH improves the model predictions?**

Using vector search algorithms like IndexFlat and Hierarchical Navigable small world, from the Facebook AI semantic search library enables us to efficiently retrieve relevant news articles based on similarity. This helps us find the most relevant news for specific currencies, even from large datasets.

These are the questions we’ll explore throughout the project. Now, Sandeep will take over the next slides.

**Sandeep: Hi everyone**

**Now, coming to the data sources we are working with:**

1. Our news data related to cryptos is sourced from a database available on public github repository, spanning from Sep-2017 to July-2024. This dataset includes news headlines and articles that can provide sentiment and context for currency movements.
2. For currency prices, we fetch data for over 400+ coins from the coincodex website.
3. We performed extensive data preprocessing, which includes cleaning, tokenizing, and aligning the news data with the corresponding price data. This allowed us to merge these two distinct datasets into a single, unified dataset that is suitable for modeling.

Now coming to the workflow of LSTM model.

Before feeding the news data into the LSTM layer, we convert the text into **vectors** using **GloVe embeddings**. GloVe, which stands for **Global Vectors for Word Representation**, provides pre-trained embeddings derived from a large corpus (like 100 billion words). These embeddings capture semantic relationships between words, allowing us to represent textual data numerically.

Once we have our vectors, they are passed to the **LSTM layer**. The LSTM uses hidden cells to maintain and update the cell state as it processes the sequence of data. This capability allows it to capture long-term dependencies within the text, making it effective for analyzing the sequential nature of news articles.

Finally, the model predicts the **percentage change in crypto price** at the output layer, which is a linear layer designed to produce continuous values. This predicted value is then compared to the actual stock price change using a **loss function**, which is typically **Mean Squared Error (MSE)**. The MSE loss is calculated based on the squared differences between the predicted and actual values, and this loss is backpropagated through the model to adjust the weights, enabling the model to improve its predictions over time.

Coming to the performance of LSTM the model is trained for 50 epochs on GPU-p100 and the best epoch came out to be 20th epoch with 0.29 loss and 0.4 rmse from the test loss we can see that the model is overfitting after 20th epoch. This is also evident from the test-RMSE plot.  
The main problem with LSTM is it has the **vanishing gradient problem**, where gradients diminish as they are backpropagated through many layers and also **GloVe embeddings** are static, meaning that each word is represented by a fixed vector regardless of its context. To address this we use transformer based models like BERT which will be explained by azmat in the coming slides.

**Azmat  
Coming to Workflow Using FinBERT**

**TF-IDF Vectorizer** known as term frequency and inverse document frequency evaluates the importance of each word in the headlines relative to the entire dataset, highlighting terms that are significant for understanding crypto movements.

Next FinBERT model is applied to enhance the initial embeddings generated from the TF-IDF vectorizer into attention enhanced embeddings. Is is the most important step. It generates dynamic embeddings which changes based on context.

**The Linear** layer outputs predictions related to crypto price movements based on the enriched information provided by the news.

**Finally a mean squared error is employed as loss function which is** crucial for backpropagation, allowing the model to adjust its weights and improve future predictions.

This is the workflow using Finbert and we can further improve the performance using semantic search algorithms from the facebook AI semantic search library which is the basis for latest RAG based techniques. Where we can use a query vand get top k news articles which are close to the query vector.

Coming to the performance of Fin-Bert the model is trained for 50 epochs on GPU-p100 and the best epoch came out to be 50th epoch with 0.19 loss and 0.05rmse which is improvement from the previous LSTM model. Now, Harsha will explain the performance of finbert when integrated with semantic search in the next slides.

**Harsha**Coming to the performance of Finbert using semantic search. The vector search algorithm which we used here is IndexFlatL2 which is a brute force algorithm. It basically calculates the Euclidean distance between query vector and all other vectors and returns the nearest k vectors. For very large datasets its very time consuming and not optimal. The performance of using this algorithm before feeding the vectors into finbert is shown here. The model is trained for 50 epochs on GPU-p100 and the best epoch came out to be 48th epoch with 0.17 loss and 0.04rmse. This is a slight improvement from the previous model but the improvement can be very high if we use even a more large dataset and this is how the modern RAG applications work. We are basically improving the model response by enriching it with relevant latest information.

So coming to the next slide the figure on the left shows how the best model is learning from the training data. The training data which we used spans from sep-2017 to July-2024 and we saved this model into a pickle file and used it for predicting on Bitcoin test data which spans from Aug-2024 to Nov-2024. Even though the predictions are not as accurate, it efficiently captures the direction of the currency movement i.e. positive or negative. This suggests that the model is effectively caputuring market sentiment even though the magnitude of the effect is slightly wrong. And this how crypto market works its highly volatile and modelling events like market shocks is extremely difficult but nevertheless our model is pretty much good and we can improve it by training it on larger datasets.

Coming to the conclusion the table here shows the improvements from each stages of our methodology. The loss improvement from lstm to Finbert is approximately 34% and rmse improvement is 87.5% which shows the exceptional power of transformer based models in text analysis. And coming to future works we can integrate API’s like polygon API for crypto prices and Yahoo finance API for news for continuous data flow. We can automate the process of data ingestion and preprocessing by building data pipelines. We can build a end-to-end RAG application and a user interface using flask or streamlit enabling users to get real-time insights. Coming to vector search algorithms if we are working with large datasets it’s better to use IVF (Inverted File Index) algorithm, which organizes vectors into clusters using techniques like K-means, so the search is only performed within clusters close to the query. Alternatively, we could use the HNSW (Hierarchical Navigable Small World) algorithm, a graph-based method that uses depth-first or breadth-first search techniques to speed up the search.

So this is our presentation and we are open to any questions. Thank you!