CHAPTER V
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusions

Based on the overall research stages, ranging from historical data collection, artificial neural network architecture design, and model training, to the evaluation process and implementation of the web-based system interface, several main conclusions can be drawn as follows:

1. Bivariate Model Performance: The LSTM algorithm in the bivariate model (BTC and Gold) is capable of predicting Bitcoin prices with a high accuracy of 99.77%. Nevertheless, this model still has a Mean Absolute Error (MAE) of 181.12 USD, indicating limited responsiveness to micro-fluctuations.
2. Linear Interpolation Performance: The use of linear interpolation techniques to align data frequencies proved effective in maintaining time-series continuity without causing significant price pattern distortion. This allows the LSTM model to stably learn precious metal macro trends within Bitcoin's micro-volatility scale.
3. Dashboard Implementation: The web-based prediction system dashboard prototype was successfully designed and implemented using the Flask framework. This system is capable of visualizing data interactively and in real-time by automating data retrieval via the Yahoo Finance API.
4. Multi-Precious Metal Influence: The addition of multi-precious metal variables (Silver, Platinum, Palladium, and Rhodium) proved to significantly increase the model's prediction accuracy to 99.84%. This confirms that industrial metals carry additional predictive feature weights not possessed by the gold variable alone.
5. Performance Gap: There is a measurable performance gap where the multivariate model is able to reduce the MAE value by 50.88 USD compared to the bivariate model. The RMSE reduction of 65.25 USD also proves that the multivariate model is more robust in minimizing prediction errors during extreme price spikes (outliers).

5.2 Implications

In the design and implementation of this computational system, there are technical and fundamental limitations that affect the scope and final results of the research:

1. **Absence of Sentiment Analysis (Purely Quantitative Approach):** The model is designed purely using historical numerical data (historical price action). The algorithm lacks Natural Language Processing capabilities to detect or analyze external sentiments, such as global crypto regulatory news, central bank (The Fed) interest rate announcements, or social media sentiment manipulation, which often trigger sudden price anomalies (Black Swan events).
2. **Data Retrieval Architecture Latency:** The backend system relies on the public REST API from Yahoo Finance, which has a request-response delay of several seconds to minutes compared to native crypto exchange data streams. This condition makes the system adequate as a human technical analysis tool but does not meet the ultra-low latency requirements to be integrated directly as an autonomous High-Frequency Trading (HFT) bot.
3. **Static Time Window Parameter (Static Sliding Window):** The input shape matrix architecture design is rigidly configured at 12 time steps (60 minutes backward). The model cannot dynamically expand or shrink its memory focus to detect long-term market trend cycles that form outside that 60-minute duration.
4. **Ignorance of Real Transaction Costs:** The Entry, Take Profit, and Stop Loss recommendation algorithms on the web system are based purely on mathematical regression price movements. This algorithm has not incorporated real-world crypto trading simulation variables such as maker/taker fees, slippage tolerance, and bid/ask spread differences.
5. **External API Ecosystem Dependency:** The operational capability of the built web system has a tight coupling dependency on the availability and data structure stability of the Yahoo Finance API endpoints. Changes in rate limiting policies or service discontinuation by the data provider can halt the prediction function instantly.

5.3 Suggestions for Future Development

To perfect this research and expand the scope of knowledge in the field of financial computing, there are several development recommendations for future research:

1. **NLP Integration and Cross-Modal Sentiment Analysis:** Developing a hybrid architecture combining the LSTM numerical regression model with Transformer models (such as BERT or FinBERT) to extract sentiment scores from financial news portals and social media platforms (such as X/Twitter), in order to improve the model's resilience against market sentiment spike phenomena.
2. **Exploration of Alternative Deep Learning Architectures:** Conducting comparative studies by implementing more modern or efficient sequential computational architectures, such as Gated Recurrent Unit (GRU), CNN-LSTM hybrids for spatial-temporal feature extraction, or Time-Series Transformer mechanisms that possess Self-Attention functions to capture long-distance correlations.
3. **Concurrent Network Infrastructure Optimization:** Migrating the data fetching architecture from a REST API to a WebSocket (WSS) protocol directly connected to major crypto exchange servers (e.g., Binance or Coinbase) to ensure a fully asynchronous, continuous, and ultra-low latency data stream.
4. **Implementation of Dynamic Hyperparameter Optimization Algorithms:** Replacing the manual hyperparameter search process with evolutionary computational methods such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), or Bayesian Optimization to automatically discover the most optimal combination of neuron count, learning rate, and sliding window size.
5. **Expansion of Macroeconomic Variable Matrices:** Enriching the model's input dimensions by injecting traditional macroeconomic indicators, such as the US Dollar Index (DXY), S&P 500 Index, Treasury Yields, and the overall cryptocurrency market trading volume.
6. **Paper Trading Simulation Module:** Developing an additional module in

the web system that facilitates virtual transaction simulations, where Profit/Loss calculations are recorded automatically, inclusive of exchange fee simulations (maker/taker fees and slippage).

7. **Multi-Resolution Testing (Multi-Timeframe Analysis):** Implementing separate model training for various time resolutions other than 5 minutes (e.g., 15 minutes, 1 hour, and daily) using an ensemble learning architecture so the system can provide predictions based on trend confirmation intersections between short-term and medium-term movements.
8. **Deployment to Cloud Computing and Containerization Ecosystems:** Packaging the backend architecture and Deep Learning computational dependencies using containerization technology (Docker), to be subsequently deployed into a distributed computing-based Cloud Computing ecosystem (such as Kubernetes), to ensure the system does not experience downtime during spikes in user inference loads.