City St George's, University of London MSc in Data Science

Predicting Cryptocurrency Volatility

Project Report 2025

Sultana Mohamed Saleh

Supervised by: Dr. Kevin Ryan

1st October 2025

By submitting this work, I declare that this work is entirely my own except those parts duly identified and referenced in my submission. It complies with any specified word limits and the requirements and regulations detailed in the assessment instructions and any other relevant programme and module documentation. In submitting this work I acknowledge that I have read and understood the regulations and code regarding academic misconduct, including that relating to plagiarism, as specified in the Programme Handbook. I also acknowledge that this work will be subject to a variety of checks for academic misconduct.

Signed: Sultana Mohamed Saleh

Abstract

This research forecasts daily prices for BTC, ETH, LTC, and XRP (from 2018–2023) using three architectures: an LSTM baseline, a CNN-LSTM with GAF/MTF representations, and an LSTM-HMM that encodes latent regimes. Datasets were chronologically split 0.65:0.1:0.25; models tuned via Bayesian optimisation, and evaluated with MSE, RMSE, R^2 and MAPE. A rule-based volatility–direction classifier assessed regime fidelity. Results show the LSTM-HMM delivers the lowest errors and clearest regime alignment—especially for BTC/ETH—while CNN-LSTM underperforms. Findings indicate regime-aware recurrent models improve robustness under market shifts, whereas image encodings offer limited benefit for daily crypto price prediction.

Keywords: cryptocurrency, LSTM-HMM, regime switching, time-series forecasting, deep learning.

GitHub Repository can be accessed here: https://github.com/Sultana2024/Cryptocurrency-Price-Prediction

Use of Gen AI

The Generative AI used is ChatGPT. I have included in Appendix H the cases in which it was used, the prompts, responses and actions taken.

1 Table of Contents

1. Introduction and Objectives
1.1. Background and Motivations

Cryptocurrencies—decentralised digital assets—have expanded rapidly, with total market value reaching approximately $3.99 trillion (CRIX index). Their growing integration into financial infrastructure positions crypto assets as a global investment class with real economic relevance. Yet price prediction remains challenging due to pronounced volatility, frequent regime shifts, and extreme events (e.g., COVID-19). Between 2018–2023, markets cycled through distinct phases: the post-2017 boom correction (2018), the pandemic-era bull run (2020–2021) driven by liquidity and stimulus, mid-2021 regulatory tightening (e.g., China) triggering drawdowns, and the 2022 FTX collapse, which further destabilised markets. While digital transformation accelerates and policy interest grows (e.g., stablecoins, central-bank digital currency initiatives), high volatility remains a barrier to wider adoption, increasing the value of robust predictive models that perform under stress.

Methodologically, a gap persists in daily, multi-year crypto forecasting across full regime cycles. For example, García-Medina (2023) focuses on hourly prices concentrated around 2020, and Agakishiev et al. (2024) utilise hourly indices (2015–2022); such designs may under-represent key daily dynamics and the full sequence of market shocks across 2018–2023. This study addresses that gap by evaluating models on daily data spanning multiple regimes, enabling a clearer assessment of generalisation across turbulent periods.

Aim of the study. To predict the daily price of four cryptocurrencies (BTC, ETH, LTC, XRP) across market regimes from 2018–2023 using deep learning models (LSTM, CNN-LSTM, LSTM-HMM), to evaluate their robustness and applicability in realistic settings, and to assess performance primarily by MSE and R^2 through volatile periods (e.g., COVID-19, FTX).

1.2. Problem statement and Scope

Problem statement – Existing studies report inconsistent accuracy for crypto price prediction and often focus on narrow time windows (or single assets), limiting conclusions about generalisation across regimes and assets with heterogeneous statistical properties. A comparative evaluation of multiple hybrid LSTM-based models across four major cryptocurrencies and a multi-year daily horizon is therefore needed to determine whether such models can deliver robust, regime-resilient performance.

Research question – Can hybrid LSTM-based models (CNN-LSTM, LSTM-HMM) accurately predict daily cryptocurrency prices and generalise across market regimes relative to an LSTM model?

Scope – The scope of this study is to predict the daily price for BTC, ETH, LTC and XRP for period spanning 2018-2023. The regression will determine the next day price y_{t+1} using chronological train/validation/test sets for all three models – LSTM, CNN-LSTM, LSTM-HMM. The evaluation relies on MSE (primary) and R^2 (secondary) as error metrics and regime-wise analyses (e.g., bull/bear/neutral).

1.3. Objectives

To address the problem posed, this study must satisfy the following key objectives:

Objective 1: To implement and train a Long Short-Term Memory (LSTM) model to predict the price for four major cryptocurrencies (BTC, ETH, LTC, XRP) using a historical dataset spanning 2018–2023. To evaluate using MSE (primary metric) and R^2. The success criteria is to obtain $MSE_{LSTM} < MSE_{Garcia\text{-}Medina}$ with Andrés García-Medina and Aguayo-Moreno's model used as a benchmark.

Objective 2: To implement a CNN-LSTM hybrid model based on the architecture used by Vidal and Kristjanpoller, 2020 for gold price volatility prediction, and evaluate whether it improves on the LSTM baseline for the same four cryptocurrencies and time window using identical error metrics. The success criteria is $MSE_{CNN\text{-}LSTM} \leq$ 17% reduction on MSE_{LSTM}.

Objective 3: To implement a LSTM-HMM hybrid model (per Zhang et al., 2022) to capture regime switching and improve the LSTM performance. The success criteria is ≥17% MSE reduction compared to MSE_{LSTM}.

Objective 4: To identify the best-performing model through hyperparameter tuning and demonstrate it outperforms literature benchmarks (ARIMA/GARCH) across volatile periods (e.g., COVID crash, FTX collapse). The success criteria is the lowest MSE across multiple seeds.

1.4. Research Questions

To answer the problem statement, the following are relevant research questions to address:

RQ1 – To what extent can an LSTM accurately forecast daily cryptocurrency prices during highly volatile periods (2018–2023), as assessed by MSE and R^2 on chronologically held-out data?
RQ2 – Do Gramian Angular Fields (GAF) and Markov Transition Fields (MTF) used as inputs to a CNN provide additional information that significantly reduces MSE relative to the LSTM baseline under the same splits?
RQ3 – Does incorporating a Gaussian HMM to infer latent market regimes improve volatility modelling and out-of-sample MSE compared with the LSTM baseline, particularly across volatile regimes?

1.5. Hypotheses

Building on the research questions, two testable hypotheses are formulated to be evaluated based on the empirical results presented in this study.

H1 – CNN-LSTM and LSTM-HMM will outperform LSTM with mean MSE at least 10% lower in 2 of the 3 seeds
H2 – CNN feature extraction on GAF/MTF representations will improve generalisation across the 2018–2023 timeframe (lower test MSE) relative to raw-series inputs.

1.6. Outcomes & beneficiary

The project outcomes are:

- Three reproducible, tuned forecasting pipelines (LSTM, CNN-LSTM, LSTM-HMM) with released code and data splits;
- a selected best model that surpasses baselines on test MSE;
- findings on representation learning (GAF/MTF) and regime modelling (HMM) relevant to volatile assets.

The beneficiaries:

- Quantitative researchers and finance practitioners (institutional and private) seeking robust crypto forecasting baselines for risk-aware decision making across market regimes.

1.7. Overview of methods

- Models – Implement, tune, and evaluate three models (LSTM, CNN-LSTM, LSTM-HMM) with ordered train/validation/test splits; MSE (primary) and R^2 as evaluation metrics.
- Process – Conduct hyperparameter optimisation (e.g., Optuna) on Bitcoin (widely studied and highly volatile), then assess transferability by evaluating the selected configurations on ETH, LTC, and XRP using identical data processing and evaluation procedures

2. Context

Forecasting cryptocurrency prices is challenging due to non-linear, heteroskedastic, and regime-dependent dynamic. Classical ARIMA/GARCH models provide baselines but assume stationarity, short memory, and fixed parametric forms; they degrade under regime shifts common in 2018–2023. Aanandhi et al.'s study shows that these models are outperformed by deep learning approaches. LSTM due to its sequential nature and memory, was widely used in time-series forecast. García-Medina & Aguayo-Moreno (2023) analyse volatility forecasting for leading cryptocurrencies around the COVID-19 crisis using high-frequency (hourly) data. They compare GARCH variants, MLP, LSTM, and an LSTM–GARCH hybrid and evaluate errors. Deep networks outperform the GARCH family; an MLP is competitive with LSTM and the hybrid, at lower computational cost. For this project, the paper justifies LSTM as a credible baseline and underscores the need for regime-aware models. Vidal & Kristjanpoller (2020) propose a CNN-LSTM that fuses image encodings—Gramian Angular Fields (GAF) and Markov Transition Fields (MTF)—with sequence modelling to forecast gold volatility (daily, 1968–2017). A VGG-based CNN extracts spatial features from GAF/MTF images; these embeddings are concatenated with an LSTM over return lags. The hybrid achieves −37% MSE vs GARCH and −18% vs LSTM. The GAF exposes autocorrelation structure; MTF encodes transition probabilities; LSTM captures long memory. Although the model does not forecast cryptocurrencies and volatility, the architecture motivates testing CNN-LSTM (GAF/MTF) for price prediction in more volatile crypto markets across five years, a gap covered in this study.

Zhang, Wen & Yang (2022) develop an LSTM-HMM for GDP fluctuation states, comparing HMM, GMM-HMM, and LSTM-HMM across rolling 4–10-year windows with monthly/quarterly CPI inputs. LSTM-HMM generally achieves the highest accuracy and consistency, improves when inputs move from quarterly to monthly and longer windows (8–10 years). The HMM contributes state persistence. Daily cryptocurrency prices—where rapid switching and noisy emissions cause phase lag—should benefit from LSTM-HMM's regime tracking, directly supporting one of this dissertation's core models. HMM with LSTM were similarly used for CPI inflation by Sivakumar (2024); HMM-derived hidden states/means are added as LSTM features. This augmentation improves accuracy and interpretability and R^2; strengthening the case to include state information into crypto price models as it is only applied to economic indicators.

Agakishiev et al. focus on regime-switching by defining three regimes using return/volatility quantiles with buffer zones to reduce churning. An LSTM predicts regime probabilities from BTC metadata (2015–2022), and feeds these into reinforcement-learning policies that trade the CRIX index on hourly data. Regime features improve training performance; out-of-sample gains are less persistent—highlighting the generalisation challenge. Their design (quantile regimes + sequence model + downstream decision) informs this dissertation's regime-aware evaluation and motivates testing hybrids that stabilise regime classification (Agakishiev, Härdle, Becker & Zuo (2024).

The cryptocurrency literature has a strong focus on a hybrid model to outperform current model. Wu et al. (2024) review deep models for crypto price prediction (LSTM/GR, CNNs, Transformers) and report that Conv-LSTM with often attains the best accuracy across two experimental during COVID. To capture long- and short-range dependencies for BTC/ETH/LTC, Khaniki & Manthouri (2024) built a Performer transformer and Bi-LSTM; it improves scalability compared with standard self-attention; Benchmarks used indicate gains over baselines, reinforcing the trend toward hybrid stacks components for crypto time series. Sepehri et al. (2025*)* — CryptoMamba introduce a Mamba-based State Space Model (SSM) for BTC, arguing SSMs better capture long-range dependencies and regime shifts than RNNs. Early results suggest improved accuracy and generalisability versus LSTMs/Transformers. While outside our model set, CryptoMamba evidences a shift toward state-space/latent-state perspectives—conceptually aligned with LSTM-HMM and regime-aware models. Further, hybrid and ensemble directions continue to proliferate. Gautam (2025) combines LSTM and XGBoost, using sentiment/macro features. The hybrid outperforms standalone models on MAPE/normalised RMSE across several coins.

A review of the literature yields support to three propositions directly tied to this dissertation's aim and methods:
(1) LSTM is a strong deep baseline for volatile windows and an appropriate benchmark against classical models and other deep learners.
(2) CNN-LSTM can add autocorrelation and transition structure potentially benefitting forecasting, but its transfer to crypto price-level prediction across 2018–2023 remains untested—justifying the comparative evaluation on daily BTC/ETH/LTC/XRP.
(3) LSTM-HMM embeds latent regimes resulting in reduced phase lag at turning points—critical for robust performance through the post-2017 correction, COVID rally, mid-2021 regulatory drawdown, and the 2022 FTX collapse.

As can be seen studies emphasise hourly horizons and short windows rather than daily price-level prediction across multiple market shifts. There is a gap in implementing CNN-LSTM (GAF/MTF) and LSTM-HMM on daily, multi-asset, multi-year crypto with explicit regime fidelity analysis. This dissertation fills that gap by comparing LSTM, CNN-LSTM (GAF/MTF), and LSTM-HMM on BTC/ETH/LTC/XRP (2018–2023) with MSE, R^2 and regime-alignment diagnostics. The review places the study's objectives in a field that is moving to which hybrid best generalises across regime shifts—and why.

3. Methods

3.1. Project Overview and Objectives

This project investigates the suitability of LSTM-based approaches to forecasting regime shifts and short-term price direction in highly volatile daily cryptocurrency data from 2018–2023. The baseline model from Garcia-Medina (2023) operates on hourly data. This chapter explores adaptations required for a daily dataset with different volatility characteristics, moving from classification (initial) to regression to optimise performance under these conditions.

3.2. Data Collection and Preprocessing

The dataset sourced from Kaggle (Mhaze, 2025) spans from 1st January 2018 until 31st May 2023 for four cryptocurrencies – Bitcoin (BTC), Litecoin (LTC), Ethereum(ETH) and XRP (XRP). Each currency has missing dates hence during pre-processing of data, a linear interpolation is applied for the missing dates based on the previous and subsequent datapoints. There are 1,977 datapoints with a daily price for each currency since cryptocurrencies trade during the weekends (in contrast with traditional commodities, equities etc.). The target value y_t is 'Close price', the last price at which the currency traded each day hence the most accurate price to capture the daily volatility of the currency. The dataset is normalised and scaled between [0,1] as currencies such as bitcoin have a large magnitude impacting the model performance and training speed.

3.3. LSTM Model

3.3.1. Initial Model (LSTM Classifier – Garcia-Medina Adaptation)

Initially, an LSTM model adapted from Andrés García-Medina and Aguayo-Moreno (2023) was implemented to predict price direction ({-1,0,1}) (decrease, neutral, increase) and to infer market regimes (bear, neutral, bull). Whereas the reference study uses hourly data, the present dataset comprises daily prices; the primary task is classification (direction), with subsequent comparison to a regression formulation. A rolling-window scheme partitions the data into training and test sets. Input window lengths $w \in \{5,7,30,60,90,120\}$ days are used to predict the next-day direction. Class labels are defined by a thresholded return sign:

$$Label \ = \ Sign\{ \left[{}^{y_{t+1}-y_t}/_{y_t} \right] - \text{threshold} \}$$

Each sample has shape X= [number of samples, number of timesteps, features]. Features include OHLC prices within the window; log returns augment the inputs to stabilise learning relative to raw prices. Monthly volatility (standard deviation of returns) is added to indicate whether transitions are transient or sustained, providing regime-relevant information beyond daily fluctuations.

The network, implemented in Keras, uses three LSTM layers with ({32,18,2}) units, ReLU activations in hidden layers, and a softmax output to produce class probabilities; Argmax selects the predicted class. RMSprop is adopted as the optimiser due to its suitability for non-stationary sequences, and batch learning is used to expedite convergence while capturing temporal patterns. Empirical performance did not match that reported by García-Medina and Aguayo-Moreno; to enable a like-for-like benchmark, additional diagnostic evaluations were subsequently conducted.

3.3.2. Diagnostic Testing and Benchmarking

To isolate whether the model architecture or data was responsible, the following tests were performed:

Benchmark A: The proposed LSTM model was applied to Garcia-Medina's hourly dataset resulting is a higher performance (higher accuracy, lower cross-entropy loss and higher f1score). The dataset is hourly with a lower volatility and smaller magnitudes (max. 160) for the bitcoin currency.

Benchmark B: The paper's model was tuned to Mhaze's dataset yielding a lower performance than the paper's original dataset. It nevertheless exceeded the performance of the proposed LSTM model although validation and test errors remain high suggesting difficulty generalising to unseen data. The paper's code was sourced from Garcia-Media's github[1] (agarciam, 2022).

In conclusion, the dataset's volatility is likely causing performance degradation in classification models.

3.3.3. Reformulating to Regression Model

Classification proved unreliable under volatility and label noise, so the LSTM was reformulated as a regressor. The hypothesis is that regression supports continuous learning and improves robustness in high-volatility settings. Benchmarks confirm improved learning on daily data (2018–2023 with multiple regime shifts) relative to earlier, lower-volatility horizons. A Kaggle dataset covering this period was used to meet the objectives. The model was tuned via Bayesian optimisation (Optuna) to select layer sizes, activations, and optimisers (Adam/RMSprop), with pruning to accelerate convergence. Data were split 65/10/25; three seeds, dropout, and improved windowing enhanced robustness and mitigated overfitting. Mean squared error was chosen as the loss (differentiable and penalising large errors); evaluation used RMSE, MAE, and MAPE.

3.3.4. Experimental Setup

The final adopted model is an LSTM regressor trained on the Kaggle 2018–2023 dataset using a 65% train, 10% validation, 10% test split, with dropout, window size 120, and features limited to the Close price. Hyperparameters were tuned using Bayesian Optimisation with the bitcoin dataset and applied to LTC, ETH, XRP dataset. Evaluation metrics include MSE, RMSE, MAE, and MAPE, as described in Chapter 4. This pipeline was selected based on the exploratory work in Sections 3.3–3.5.

The model was implemented using VS Code. The Keras library was used along with scikeras, a wrapper to use scikit-learn Bayesian optimisation.

3.4. CNN-LSTM Model

3.4.1. Initial Model (CNN-LSTM adapted from Gold price paper, Vidal & Kristjanpoller, 2020)

Adapted from the model proposed by Vidal & Kristjanpoller (2020) for gold price forecasting

This section presents the initial hybrid model architecture used to predict Bitcoin prices, based on a fusion of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. The model adapts and extends the work of Vidal & Kristjanpoller (2020), originally applied to the gold market, to address the unique dynamics of the cryptocurrency market—particularly Bitcoin, which exhibits higher volatility, more frequent regime shifts, and greater susceptibility to sentiment-driven changes. It is also applied to Litecoin (LTC), Ethereum (ETH) and XRP.

<u>Data Preparation and Train/Test Split</u>

The model uses the same train/validation/test split described in Section 3.3 to maintain consistency across experiments. The historical Bitcoin price series comprises 1,977 daily closing prices with 65% used during training ~ 1,139 samples and 10% validation, which serve as the input for the feature engineering process and subsequent model training.

<u>Encoding Time-Series as Images: GASF, GADF, and MTF</u>

To enhance the model's ability to capture temporal dependencies and market dynamics beyond what traditional LSTM windows offer, the time series is transformed into image-based representations using the following three methods:

[1] https://github.com/agarciam/lstmgarch_project

- Gramian Angular Summation Field (GASF)
- Gramian Angular Difference Field (GADF)
- Markov Transition Field (MTF)

These transformations encode the time-series data into 2D matrices, capturing both correlation patterns (GASF, GADF) and state transition probabilities (MTF). Each matrix is of size 224x224x3 channels, reflecting the size of the image encoded with 1,139 training samples (the same image size is maintained for validation, testing).

GASF and GADF:

These methods map the normalised time series to angular space, permitting computation of pairwise relationships within [−1, 1]. The resulting 224×224 matrices encode dependence in complementary forms: the Gramian Angular Summation Field (GASF) captures additive correlations via cosine similarities, whereas the Gramian Angular Difference Field (GADF) emphasises contrasts through sine components. In cryptocurrency markets—where strong autocorrelation often accompanies directional phases—these representations expose coherent trend structures (e.g., momentum, mean reversion) that may be under-represented by local sliding-window LSTMs. For example, a 15% surge followed by reversion is reflected as a build-up and subsequent unwinding of angular correlations; a sharp attenuation or rotation in these relationships can signal turning points, enabling a CNN to detect likely reversals.

$$\begin{bmatrix} cos(\phi_0+\phi_0) & \cdots & cos(\phi_0+\phi_{29}) \\ \vdots & \ddots & \vdots \\ cos(\phi_{29}+\phi_0) & \cdots & cos(\phi_{29}+\phi_{29}) \end{bmatrix} \qquad \begin{bmatrix} sin(\phi_0-\phi_0) & \cdots & sin(\phi_0-\phi_{29}) \\ \vdots & \ddots & \vdots \\ sin(\phi_{29}-\phi_0) & \cdots & sin(\phi_{29}-\phi_{29}) \end{bmatrix}$$

MTF:

The Markov Transition Field (MTF) discretises a time series into quantile-based states and encodes, for each pair of times, the empirical probability of transitioning from one state to another. The resulting matrix captures second-order dynamics—how behaviour evolves—rather than only absolute levels (e.g., transitions from rising to falling states). In highly volatile cryptocurrencies, where regimes can switch rapidly, such transition structure is informative. Unlike LSTMs, which operate within a fixed input window, the MTF provides a global view of state evolution over the entire sequence. With an input length of 224, it can represent extended regime shifts (bullish, bearish, stagnant); for instance, during a bear phase the MTF concentrates probability mass on transitions between declining states.

$$T_{ij}=P(\text{transition from state } i \to j) \qquad \begin{bmatrix} P_{0\to0} & \cdots & P_{0\to223} \\ \vdots & \ddots & \vdots \\ P_{223\to0} & \cdots & P_{223\to223} \end{bmatrix}$$

Image-Based Encoding:

Stacking the GASF, GADF, and MTF matrices as RGB channels creates a spatial representation of temporal dynamics, allowing a convolutional neural network (CNN) to extract high-level spatial features such as periodicity or correlation hotspots. In effect, this hybrid representation enables the model to learn from both local and global temporal patterns—a critical advantage in financial time series characterized by non-stationarity, noise, and heavy-tailed behaviour.

<u>CNN</u>

VGG16 Module: Feature Extraction from Time-Series Images

The GASF, GADF, and MTF matrices are stacked as RGB channels to form a single image (size: 224x224× 3), encoding correlation and transition dynamics in the cryptocurrency time series. This image is processed by a VGG16 convolutional neural network, pre-trained on ImageNet, with its fully connected layers removed.

Although VGG16 was developed for natural image classification, its early convolutional layers are known to extract generalisable spatial features such as edges, contours, and textures. When applied to time-series images, these features help identify structural patterns such as trend consistency, regime boundaries, and temporal anomalies. The convolutional base is fine-tuned on the financial data, and its final output is a dense feature vector summarizing the time-series structure.

LSTM Module

In parallel, the raw normalized time series is fed into an LSTM network, which models temporal dependencies using a sliding window of size $w = 224$ (same input as CNN). The LSTM captures short to medium term dynamics such as momentum, volatility, which are essential for forecasting price direction in rapidly shifting crypto markets.

Fusion via Concatenation

The output vectors from the CNN and LSTM modules are concatenated to form a joint representation that combines spatial (correlation/state-transition) and temporal (sequence-based) features.

This hybrid design leverages the strengths of both architectures:

- LSTM captures sequential behaviour (e.g., momentum within recent days),
- CNN identifies global structural patterns (e.g., regime shifts, reversals) from the image-transformed data.

This concatenated vector is then passed through one or more fully connected layers, culminating in a regression output layer which predicts the next-day closing price of Bitcoin. The model is trained to minimise Mean Squared Error (MSE) between the predicted and actual values.

3.4.2. Diagnostic Testing and Benchmarking

Given the weak performance of the initial CNN-LSTM, the CNN component was tuned independently to identify an architecture suitable for Bitcoin price prediction. A VGG16 backbone with ImageNet pre-training was adopted to leverage robust, transferable convolutional features commonly effective for structured time-series encodings such as GAF/MTF. The optimised CNN comprises two convolutional blocks (128 and 64 filters) to capture multi-scale variability associated with cryptocurrency volatility, followed by dropout to mitigate overfitting. The CNN produces a 128-dimensional feature vector that is concatenated with the LSTM output. The LSTM configuration from the baseline model is retained; the input window is set to 224 days, matching the CNN window, to prevent leakage and ensure temporal alignment (i.e., predictions at (y_{225}) use observations (y_{1})–(y_{224}) only). The concatenated representation is passed to a regression layer to generate the target price.

3.4.3. Experimental Setup

The model was implemented using the Keras and TensorFlow. Due to the computational demands of processing large time-series image inputs (1139, 224, 224, 3) and training the CNN-LSTM hybrid model, all experiments were conducted on Google Colab Pro using Nvidia A100 GPU acceleration.

The train/test split followed the configuration outlined in Section 3.3 for consistency across models.

Model optimisation was performed using the RMSprop optimiser, with performance evaluated using four standard regression metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

3.5. LSTM-HMM Model

3.5.1. Proposed model

The initial task was classification which was re-written as a regression task as per section 3.3. The reference paper by Zhang, Wen and Yang - Prediction of fluctuation China GDP, 2022, was a classification task with a Long Short Term Memory (LSTM) followed by Hidden Markov Model (HMM). However, HMM is suitable for a classification as it detects the state rather than predict a time-series datapoint.

Hence, the model was modified to suit a regression task whilst detecting the hidden states to detect regime changes. The model to predict four cryptocurrencies' prices (BTC, ETH, LTC, XRP) is a hybrid architecture that combines a Gaussian Hidden Markov Model (HMM) for regime detection with a stacked LSTM for nonlinear sequence modelling. The HMM captures persistent latent market regimes (e.g., calm vs high-volatility phases), while the LSTM learns temporal patterns in recent observations. Their learned representations are concatenated and mapped to a one-step-ahead price prediction via a regression head.

<u>GaussianHMM</u>

The input of the GaussianHMM is the closing price for each cryptocurrency (1D) to determine the emission probability for each state. To predict a price at y_{t+1} for a high volatility cryptocurrency, the transition to different regimes is assessed by 3 hidden states according to the Markov's principle. The transition from each s_0 to s_1 or s_2 is causal i.e. depends on the price from time $1 \rightarrow t$.

Posterior probability: $p(s_t / y_{1:t}) \rightarrow p(s_{t+1} / y_{t+1})$

For a time-series with the properties as in 3.1., a Hidden Markov Model with Gaussian emissions is suitable as it models latent states with persistent transitions. Each state's mean and covariance captures the regime changes due to drastic changes (from bull to bear). The EM algorithm is based on e-step to obtain state posteriors and m-step, to update the parameters such as probability, means and covariance.

Posterior probabilities for each y_{t+1} (target) in the set is computed such that the causality as above is maintained and no future datapoints (as opposed to Viterbi) are included to predict transition for price y_t to y_{t+1}. Finally, a dense layer is added to build 64-dimension feature vector as an output of the GaussianHMM using ReLU activation function as the input is from [0,1].

$y=ReLU(w_x+b)$

<u>LSTM</u>

The window input is a 120-days datapoints consistent with prior models and empirically the strongest window for the LSTM. The model consists of a 3 layer stacked LSTM with hidden sizes 64, 8, 4 (tanh activations), with a dropout layer after the first LSTM to mitigate overfitting under high volatility (i.e. avoid impacting validation/test performance, generalisability of the model).

<u>Concatenated Output</u>

The learned feature vector is then concatenated with the output of GaussianHMM in a dense layer producing a feature vector of size 64 with activation function 'Tanh' with domain [-1,1]. The final regression layer (activation='Linear') results in the target price at $t+1$.

3.5.2. Experimental Setup

The dataset is min–max normalised to [0,1] and partitioned into 65/10/25 train, validation, and test splits. For the LSTM–HMM, the HMM inputs are aligned with the LSTM 120-day sliding window to prevent leakage; the same windowing is used across all models. Hyperparameters are tuned via Bayesian optimisation (Optuna) on Bitcoin, then applied to ETH, LTC, and XRP to assess transferability. Robustness is assessed with three seeds (42, 45, 91). In the HMM component, expectation–maximisation is run with a 200-iteration cap and multiple initialisations to mitigate local optima. A ReLU nonlinearity is used in the neural layers, and RMSprop is adopted for stable optimisation on non-stationary sequences. Final metrics (MSE, RMSE, MAE, MAPE) are reported as mean ± standard deviation across seeds. Test-set predictions are inverse-scaled and compared to original closing prices to evaluate performance in real magnitudes.

3.6. Regime Classification

3.6.1. Rule-based classification

To address objective 4, the forecasting models are assessed on their ability to reproduce market regimes in the test period. Rather than relying on internal model states, a rule-based classifier is applied to both actual and predicted series (March 2022–May 2023) to label volatility levels and return directions. This yields a like-for-like regime map for the ground truth and for each model, enabling direct comparison of regime fidelity. In many asset classes returns and volatility are negatively correlated; in cryptocurrencies this relationship is mixed or weak, necessitating the joint use of volatility and returns to characterise trend and direction.

Prior studies often infer regimes via latent-state methods (e.g., HMM/Markov-switching, MS-GARCH) and then incorporate those states downstream. The hybrids evaluated here adopt different internal mechanisms focused on improving price prediction and regime alignment. The volatility–direction rule functions as a model-agnostic benchmark: it is transparent, captures volatility clustering and directional phases, and avoids circularity by mapping each model's predictions to regimes under a common external procedure.

3.6.2. Implementation

Let P_t denote the closing price. The daily log-returns – $r_t = \ln P_t - \ln P_{t-1}$ – are computed on (i) the actual test prices and (ii) one-step prediction on $\overline{P}_t$. Volatility is the rolling standard deviation of returns over short period with *window* ∈ {5,7,10} days to reflect the cryptocurrency's fast regime shifts. From the tested values, 10-day rolling std. was selected for accuracy. Ilyas Agakishiev et al.'s 30-day window was excluded used as the dataset is only 1,977 datapoints of which 25% are used for testing. Additionally, the dataset selected covers a highly volatile period which means faster switching hence shorter horizons to be tested.

Volatility thresholds are determined based on test set quantile distribution similar to Ilyas Agakishiev et al. The 33rd and 67th percentiles' are computed $\{q_{0.33}, q_{0.67}\}$ as low and high volatility respectively. The data is then labelled as follows – for each test day (t), let σ_t be the 10-day rolling volatility and μ_t the 3-day average of returns: $\mu_t = \frac{1}{3}\sum_{i=0}^{2}(r_{t-i})$. The volatility quantiles and sign of the average returns informs the classification as follows:

$$
Regime_t = \begin{cases} \text{Neutral}, & \sigma_t \le q_{0.33} \\ \text{Bull}, & \sigma_t \ge q_{0.67} \text{ and } \mu_t > 0 \\ \text{Bear}, & \sigma_t \ge q_{0.67} \text{ and } \mu_t < 0 \\ \text{Uncertain}, & \text{otherwise} \end{cases}
$$

Regime change labelling changes only after 3-days to ensure no transient peaks/troughs are capture as a shift. The windows and parameters are retained across all models and currencies for consistent comparison.

4. Results
4.1. Dataset
4.1.1. Pre-processing Datasets
The dataset is sourced from Kaggle (Mhaske, 2025) and spans January 2018–May 2023, encompassing key regime shifts. Each asset (BTC, ETH, LTC, XRP) includes date, open, high, low, and close. Preprocessing: removed missing/duplicate records; linearly interpolated isolated gaps using adjacent values to limit distortion of local mean/variance; scaled to [0,1] with Min–Max for consistent inputs and efficient training across models.

4.1.2. Bitcoin, Litecoin, Ethereum and XRP statistics
Key statistics are summarised in Table. Pre-scaling, Bitcoin exhibits the largest USD magnitudes; Ethereum and Litecoin are smaller, with XRP lowest. ETH's mean daily return is 0.16% with 4.78% standard deviation; BTC averages 0.11% with 3.76% volatility. Annualised volatility indicates substantial risk: XRP ≈110% (highest), BTC ≈71.8% (lowest).

MACD characterises momentum: positive values indicate bullish pressure, negative bearish. BTC's small mean MACD relative to price, combined with a large dispersion (≈1,192), suggests short boom–bust phases. ETH shows stronger bullish tendency with larger swings (≈86). XRP's mean near zero implies neutrality and relative stability, consistent with higher regression accuracy.

Stationarity was assessed using ADF and KPSS. ADF tests the null of a unit root (non-stationarity) against stationarity; KPSS tests the null of stationarity against non-stationarity. Given pronounced fluctuations, prices are typically non-stationary, while returns tend to be closer to stationary, influencing modelling performance.

Cryptocurrency	Mean	Std. Price	Min.	Max.	Mean Return	Std. Return	Annualised Volatility %	Mean MACD	Std. MACD
BTC	20,667.66	16,399.60	3,188.00	67,802.00	0.0011	0.0376	71.8	49.36	1,192.89
LTC	98.42	58.74	22.58	373.64	0.0009	0.0520	99.4	-0.49	7.04
ETH	1,188.20	1,163.20	81.72	4,800.00	0.0016	0.0478	91.4	3.85	86.07
XRP	0.51	0.332	0.14	2.78	0.0009	0.0577	110.2	-0.01	0.05

Table. Cryptocurrencies statistics

In Bitcoin, the ADF p-value for price is 0.55, so the unit-root null is not rejected (price non-stationary); for returns, (p=0), so (H_0) is rejected (returns stationary). Overall, ADF and KPSS indicate stationary returns and non-stationary prices; isolated contradictions for LTC/XRP are weak, so the conclusion holds. ETH exhibits the highest p-values, followed by BTC, LTC, and XRP, consistent with greater complexity and volatility.

Cryptocurrency	ADF Test		KPSS Test	
	Price p-value	Return p-value	Price p-value	Return p-value
BTC	0.550 → price non-stationary	0.00 → returns stationary	≤ 0.01 → price non-stationary	0.10 → returns stationary
LTC	0.017 → price stationary (weak)	0.00 → returns stationary	0.022 → price non-stationary	0.10 → returns stationary
ETH	0.628 → price non-stationary	0.00 → returns stationary	0.010 → price non-stationary	0.10 → returns stationary

XRP	0.007 → price stationary (weak)	0.00 → returns stationary	0.017 → price non-stationary (weak)	0.10 → returns stationary

Table. Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests

ACF and PACF analyses show rapid decay after the first lag for all four assets, with coefficients near zero thereafter. This “white-noise-like” behaviour implies limited autocorrelation beyond short horizons and is consistent with a mixing process (dependence declining with time). Such dynamics support sequence models that emphasise short-to-medium temporal structure, aligning with the LSTM backbone used across models.

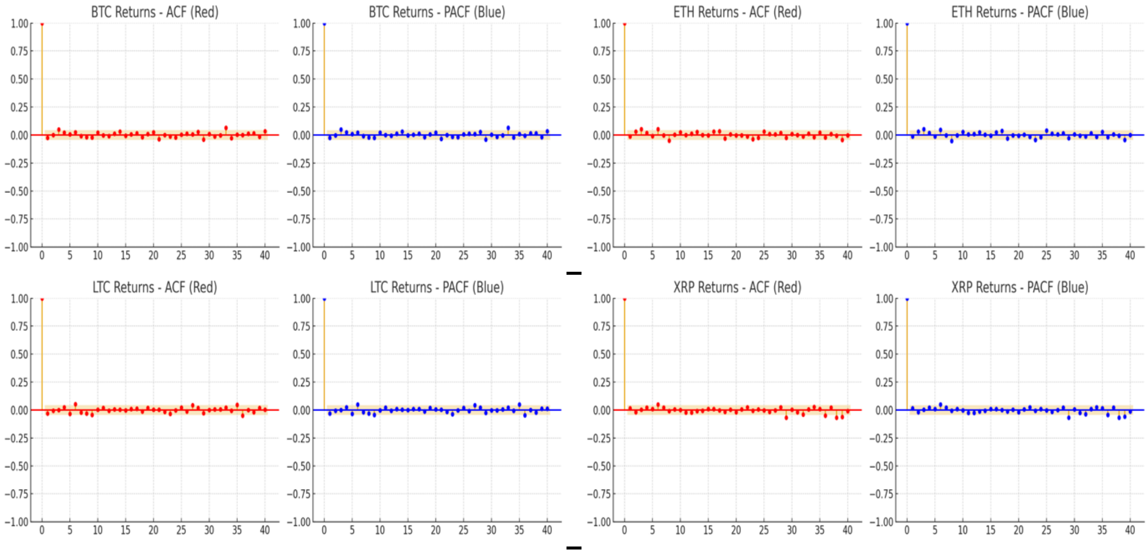


Figure. ACF and PACF Plot for Bitcoin, Ethereum, Litecoin and XRP

4.1.3. Cryptocurrency properties

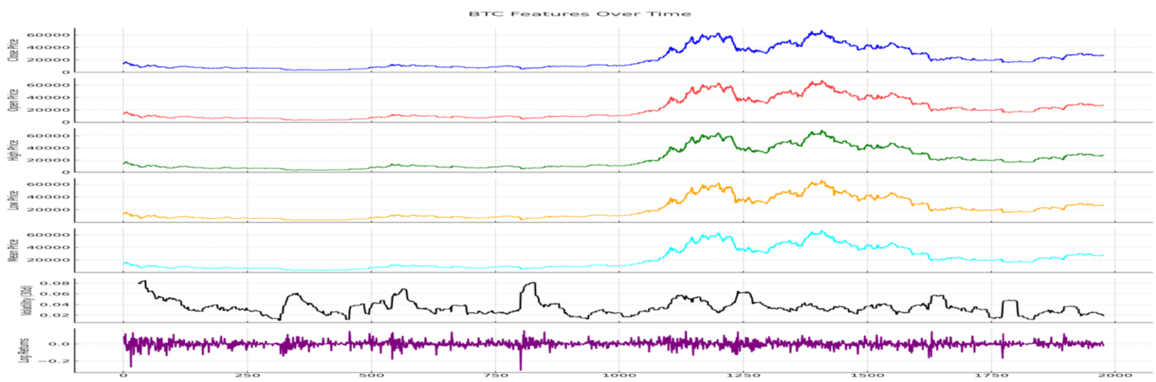


Figure. Bitcoin Features from 2018 to 2023

The figure shows the open, high, low, and close series in the original Bitcoin dataset. Log returns and volatility were computed for diagnostics but excluded as input features because preliminary tests indicated degraded predictive accuracy. The final design uses a single-feature input (close price) for the regression task. Empirical inspection indicates that open, high, low, and close have near-identical first and second moments; after min–max scaling, their differences are negligible and do not contribute additional signal, warranting removal. Volatility and log returns—though widely used—did not improve performance in this setting and were therefore omitted.

Feature plots for LTC, ETH, and XRP are provided in Appendix E. ETH exhibits frequent regime shifts (steep rises followed by steep declines) and relatively large peak magnitudes, indicating greater modelling difficulty due to transient states. XRP shows the lowest volatility and fewer shifts, suggesting greater stability. The same input specification (close price only) is applied across all assets.

4.1.4. Train, validation, test split

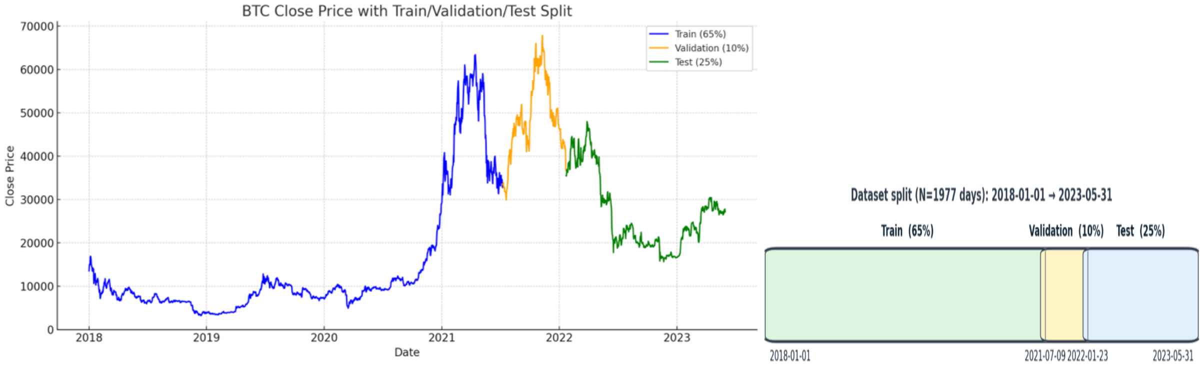


Figure. Line plot of train, validation, test split by dates for Bitcoin & diagram of sample split by dates

The figure depicts the chronological train, validation, and test partitions for Bitcoin. The same proportions and dates are used for ETH, LTC, and XRP. Training ≈3.5 years, validation ≈0.5 year, and test ≈1.5 years. Training spans multiple regime shifts; validation is shorter and more volatile with larger peaks; the test period differs markedly—a common challenge in time-series generalisation. This partitioning enables rigorous assessment of generalisability and overfitting. Splits were chosen to include pronounced Bitcoin shifts, as models are tuned on BTC and then evaluated on the other assets under the same scheme. Although longer training horizons can reduce statistical mixing, the selected windows proved suitable. Consistent with ACF/PACF diagnostics, mixing is retained for all four currencies.

4.2. LSTM Results

This section presents the results of the LSTM models which was reformulated into a regression framework. The results are structured according to model training performance, a comparative analysis with the initial classification approach, hyperparameter tuning, predictive accuracy and four cryptocurrencies' performance.

4.2.1.Model Training Performance

The final LSTM regression model was trained using the daily Kaggle Bitcoin dataset (2018–2023) with a window size of 120 timesteps (days) and only the close price as the input feature as per 4.1. The model predicts the next day's closing price y_{t+1} with a training, validation and test set of 1,165; 77 and 375 prices respectively. The rolling window size is maintained across the three sets. The loss function minimised is Mean Squared Error (MSE) in this regression model with batch learning (size 64) to improve learning speed and generalisability. The training performance shown in *figure below* indicates that the model converges after 20 epochs with a of MSE ~ 0.002. The normalised predicted price is thus close to the target value, indicating a high accuracy over a period from 3.5 year period. Across 120 epochs, the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) converge steadily around the 20th epoch as well followed by a plateau indicating stable learning, RMSE~0.025 i.e. error between target and prediction is 2.5%. The Mean Absolute Percentage Error (MAPE) is quite high as it hovers around 20,000 (very high) however it is unstable. As for the validation performance, it is unstable across all four metrics (incl. MAPE but not visible due to magnitude difference). This is due to the high volatility of the validation set with substantial transient peaks, small sample size (only 77 points). This train, validation and test split was selected from three iterations where a larger validation sample yields improved performance however both training and testing performance reduces drastically. The volatility of the currency modelled requires more inputs. The MAPE training value suggests that the model error is 20,000% i.e. it is not learning. However, as the input is scaled between [0,1] and BTC has high maxima relative to mean, there are values close to 0 in the denominator leading to an exploding MAPE. Also, the MAPE training vs. validation differ greatly as it hovers between 4-7% suggesting a very accurate model. The validation set has high values which would be further from 0 confirming the analysis. As a result, this metric was proven to be less effective to evaluate training for this BTC set.

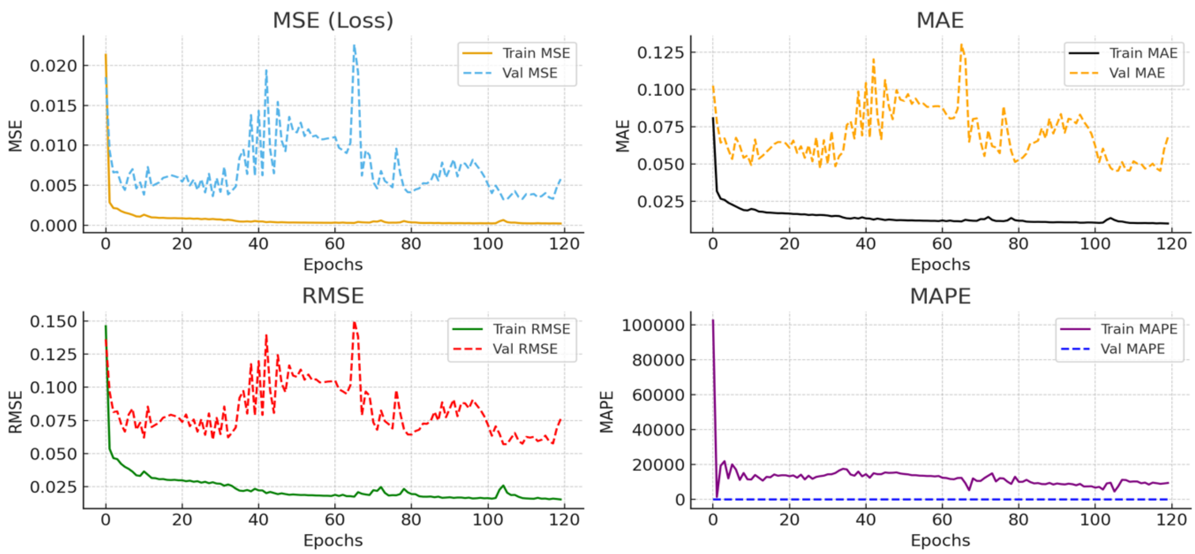


Figure. Bitcoin optimised model – training/validation curve

As outlined in 3.3.1., each model is run on three seeds 42,45,91 and the results are recorded. The above best training performance is at seed=45. In the table below, the train/val/test error for MSE, RMSE, MAE, MAPE and R^2 is reported as the mean± std. deviation of the three runs. The full results are included in Appendix D. The lowest MSE error is the training error at 2.3e-3 with a validation and test error x10 higher at 2.3e-2 and 2.1e-2 respectively, with a larger deviation for the test error. This means that the model is performing better on training data and is generalising poorly. The coefficient of determination R^2 is a useful method to measure how well the model explains the data, however since cryptocurrency prices are noisy and non-stationary, this can lead to errors. In this model, the model explains only 49.39% of the variance in the price however the deviation is large at ±39.03%. Hence, it is important to account for the other four metrics.

BTC - LSTM	**MSE**			**RMSE**			**MAE**			**MAPE (%)**			**R² Score**
	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
Mean ± std. deviation	0.000236 ± 0.000026	0.002320 ± 0.000971	0.002137 ± 0.001648	0.015338 ± 0.000839	0.047406 ± 0.010446	0.043395 ± 0.019506	0.00970 ± 0.00031	0.03728 ± 0.01036	0.03236 ± 0.01494	7,300.6217 ± 3,605.9623	4.98179 ± 1.29678	0.104505 ± 0.047281	**0.4939** ± **0.3903**

Table. BTC – MSE, RMSE, MAE, MAPE and R^2performance

Overall, the low training MSE supports the hypothesis from Section 3.3.3. that reformulating the problem as a regression task enables the model to learn more robust representations, particularly under volatile conditions.

4.2.2.Comparison to Initial Approach

In the initial phase, a classification-based LSTM was implemented using an adaptation Garcia-Medina's architecture (refer to section 3.3.). This model aimed to predict price direction as one of three discrete classes {-1,0,1} for {bear, neutral, bull market} respectively. However due to the volatility and noisiness of cryptocurrencies, the classifier severely underperformed with a test accuracy around 0.38-0.45 across multiple configurations. The cross-entropy loss remained high as per the training performance figure below, at 0.80 around epoch 100 with an increasing training accuracy at 0.65. The confusion matrices consistently mis-classified 'decreasing market condition' i.e. (-1). The reproduction of the Garcia-Medina model yields higher learning with training and validation loss converging around 0.001. Overall, classification was deemed unsuitable for this task due to the model's poor learning on this dataset.

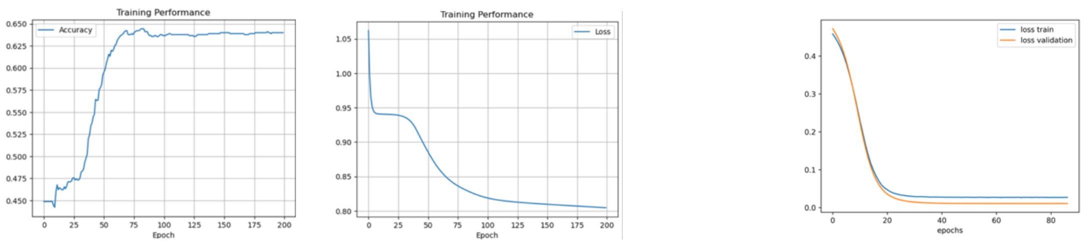


Figure. Training accuracy, cross-entropy LSTM classifier & (right) Reproduction of Garcia-Medina model using paper's original dataset

To further confirm this, tests with the following configurations were conducted as a benchmark (the error here is MSE for comparison):

	Test Configuration	Train MSE	Comment
A	Garcia-Medina model on Garcia-Medina data	0.001	Performs well due to lower volatility and hourly input resolution
B	Garcia-Medina model on daily Kaggle dataset (study dataset)	0.17(train) 0.19 (val.)	Outperforms initial model, but poor generalization due to volatility of daily prices vs. hourly prices.
C	LSTM regressor on Kaggle dataset (final model)	~0.00024	Most stable and accurate model. Improved accuracy compared to paper's result.

The LSTM regressor achieves a training loss of 2.4×10^{-3}, outperforming the reproduced García-Medina baseline (1.0×10^{-2}), indicating greater capacity to capture market shifts. Notably, García-Medina employs hourly data up to 2022, whereas the present study spans 2018–2023 to encompass multiple steep regime changes. These results underscore that dataset characteristics—scale, volatility, and sampling frequency—materially influence model performance. Reformulating the task as a regression improved training convergence and test-set accuracy.

4.2.3. Hyperparameter Tuning Results

To optimise the regressor's hyperparameters, Bayesian optimisation was employed. Using pruning at 10 and 20 trials across multiple runs, the configuration yielding the lowest MSE was selected. Relative to alternative setups—including shorter input windows (60, 90) and different optimisers (e.g., RMSprop)—the chosen specification provided the most favourable balance between learning stability and predictive accuracy on the test set.

Hyperparameter	Range of values tested	Optimised Parameter Value
Window Size	[5, 10, 30, 60, 90, 120]	120
Optimizer	[Adam, RMSprop]	Adam
Batch Size	[16, 32]	32
Epochs	[50, 200]	120
Hidden Layers	[2, 3, 4]	3
Units per Layer	[18, 32, 64] [18, 32, 64] [2,4]	[64, 16, 2]
Activation Function	[relu, tanh, sigmoid]	'tanh'

4.2.4. Predictive performance

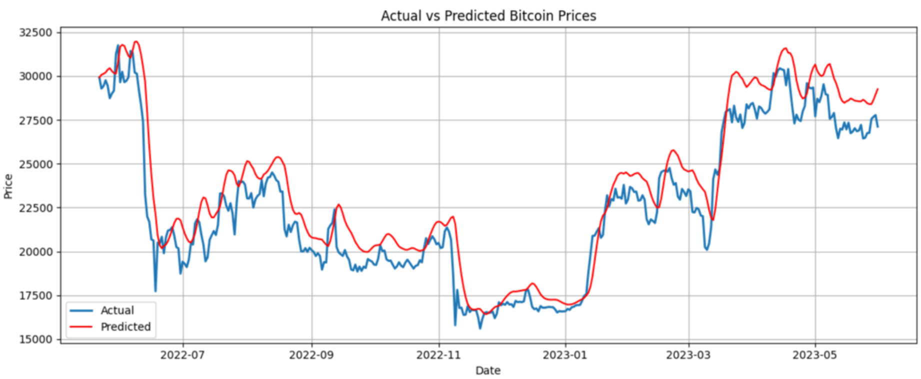


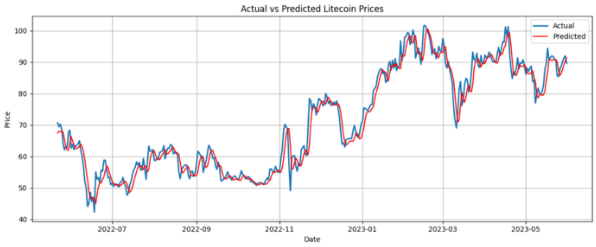
Figure. Bitcoin – Model price prediction vs. actual target value (seed:42)

The prediction graphs corroborate the error metrics, indicating high accuracy in value prediction and effective capture of market shifts, with only minor phase lags for LTC and XRP. XRP performs less reliably at lower price magnitudes (≈0.5), where deviations from the ground truth are more pronounced. By contrast, Bitcoin attains the strongest performance under LSTM-HMM, underscoring the benefit of GaussianHMM in modelling volatility via latent state transitions. This regime-based representation enables adaptation to sharp fluctuations characteristic of highly volatile assets such as BTC and ETH, while providing limited gains for more stable, low-magnitude series like XRP, where regime dynamics are weaker.

4.2.5. Comparative analysis of BTC, ETH, LTC and XRP performance

Mean ± std. deviation	**MSE**			**RMSE**			**MAE**			**MAPE (%)**			**R^2 Score**
	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
BTC	0.000236 ± 0.000026	0.002320 ± 0.000971	0.002137 ± 0.001648	0.015338 ± 0.000839	0.047406 ± 0.010446	0.043395 ± 0.019506	0.00970 ± 0.00031	0.03728 ± 0.01036	0.03236 ± 0.01494	7,300.6217 ± 3,605.9623	4.98179 ± 1.29678	0.104505 ± 0.047281	**0.4939 ± 0.3903**
LTC	0.000366 ± 0.00002	0.001364 ± 0.00002	0.000096 ± 0.000005	0.019118 ± 0.00060	0.036925 ± 0.00031	0.009791 ± 0.00027	0.01149 ± 0.0004	0.02523 ± 0.0001	0.00721 ± 0.0003	4,738.80 ± 3,980.97	5.6463 ± 0.0101	0.05635 ± 0.0023	**0.9545 ± 0.003**
ETH	0.000169 ± 0.000007	0.009362 ± 0.004591	0.000529 ± 0.000127	0.013008 ± 0.000273	0.094795 ± 0.023749	0.022885 ± 0.002690	0.00720 ± 0.00013	0.07960 ± 0.02000	0.01661 ± 0.00186	1351.5019 ± 726.7923	9.36406 ± 2.13846	0.05647 ± 0.00702	**0.8326 ± 0.0403**
XRP	0.000218 ± 0.000014	0.000269 ± 0.000024	0.000052 ± 0.000001	0.014763 ± 0.000466	0.016397 ± 0.000721	0.007233 ± 0.000091	0.00831 ± 0.00017	0.01244 ± 0.00083	0.00492 ± 0.00017	6331.2942 ± 4527.8605	4.33479 ± 0.27293	0.05052 ± 0.00159	**0.8892 ± 0.0028**

The predictive performance varied across currencies. The highest R^2 was achieved for LTC at 95.45% ± 0.3%, followed by XRP at 88.92% ± 0.28% and ETH at 83.26% ± 4.03%, while BTC was the lowest at 49.39% ± 39.03%. The small standard deviations for LTC, XRP, and ETH indicate greater robustness compared to BTC. Despite ETH being the most volatile asset, it recorded strong training and test losses (1.7e-3 and 5.3e-3) but performed poorly on the validation set (9.4e-2), highlighting limited generalisation to unseen data. LTC and ETH both exhibited validation losses approximately ten times higher than their training losses, similar to BTC, but LTC maintained a visibly smaller test loss than ETH, reinforcing its superior predictive stability. XRP delivered the most consistent results, with training and validation losses of similar scale and a test loss almost an order of magnitude smaller. This strong generalisation aligns with earlier findings (Section 4.1) where XRP's dataset was shown to be comparatively stable with low volatility (MACD~0).



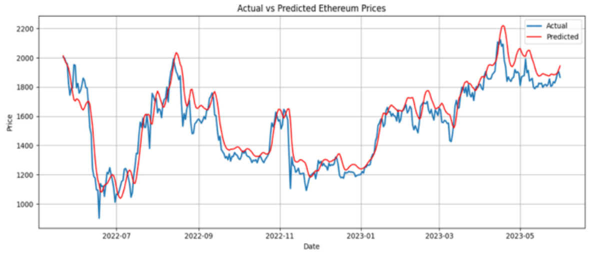


Figure. Litecoin –Price prediction vs. actual (seed:45) Figure. Ethereum–Price prediction vs. actual (seed:45)

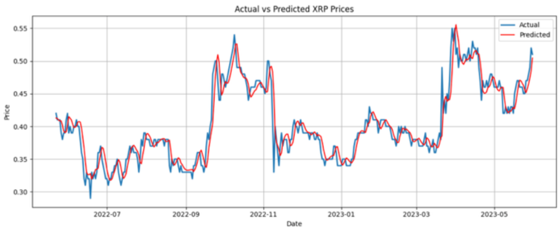


Figure. XRP –Price prediction vs. actual (seed:91)

The comparison between predicted and target price graphs confirms these patterns. XRP aligns most closely with actual prices, followed by LTC, then ETH. ETH shows a mild phase shift, smaller than BTC. Full training and validation curves appear in Appendix B. The final regression model effectively learns long-term Bitcoin patterns—especially with Bayesian optimisation and early stopping—and outperforms García-Medina's benchmark. XRP, LTC, and ETH surpass BTC and exhibit more robust results (smaller standard deviations) across error metrics.

4.3. CNN Results

This section presents the results of the CNN-LSTM models in terms of GAF/MTF visualisations, training performance, comparison with the initial CNN model, hyperparameter tuning, predictive accuracy, and cross-currency performance.

4.3.1. Grammian Angular Field Markov Transition Field

(Please note the visualisation of GASF/GADF/MTF were generated by AI)

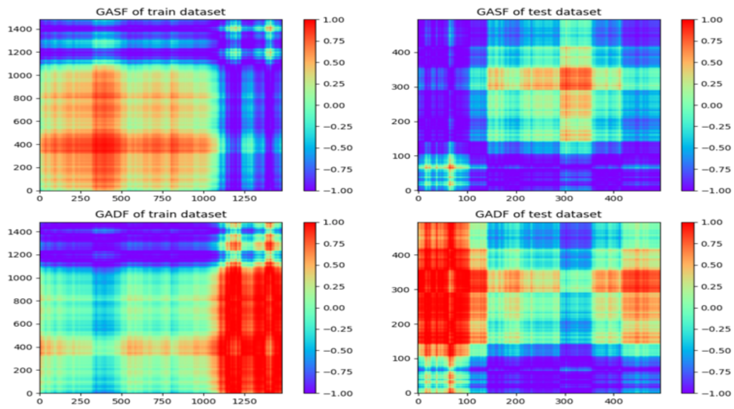


Figure. GADF/GASF Bitcoin matrices train set (left) and test set (right)

GAF matrix explains the spatial, temporal features using a GASF and GADF. It is important to visualise these matrices for bitcoin to grasp the correlation within the data. In GASF, the dark red sections indicates a ‘perfect alignment’ of point at index *i,j* ; i.e. these points have a strong positive correlation (~1). Whereas the dark blue/purple indicates a ‘perfect opposition’ of point at index *i,j* ; i.e. these points have a strong negative correlation (~-1). The matrix show a strong positive correlation between i ∈[300-500] and j ∈ [0-1,000]. This means that BTC prices at i indices move in the same direction as prices at j indices in this interval. Conversely, there is a negative correlation between i ∈ [1,125-1,400] and j ∈ [0-1,400] which suggests that as the price increases at i indices, it strongly decreases at j indices (opposite magnitude). As for the GADF, the dark red indicates a ‘strong positive temporal lead/lag relationship’ of point at index *i,j* ; i.e. at i, the angle leads index j (90° ahead of j) ~1. Whereas, the dark blue/purple indicates a ‘strong negative temporal lead/lag relationship’ of point at index *i,j* ; i.e. at i, the angle lags index j (90° behind j) ~-1. The matrix shows at index i ∈ [1,125-1,400] and j ∈ [0, 1,150]; i leads j meaning if the price of BTC increases at i, it follows at j. For i ∈ [0-1,125] and j ∈[1,150-1,400], i lags j hence if price at i is stagnant, it would have reached a spike at j (directional/temporal difference).

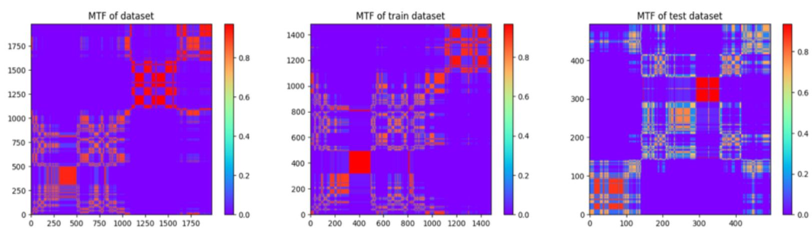


Figure. MTF for Bitcoin train, test set

In a MTF matrix, each MTF entry is the probability of transitioning from the state at time i to the state at time j. In the dataset matrix (left), at index i~300, the probability to transition to the state at time j~400 is 0.75 (high). Practically, this translates as : if BTC is high at time i then, there is a 75% probability it will remain high at time j. In the three matrices, the highest probabilities are observed in the full dataset, although all three's strongest probabilities are across the diagonal bins. The test data matrix exhibits the lowest transition probabilities with i ∈[200,300] and j ∈[175,200] have less than 30% transition probability i.e. if BTC is low at i than the chance it remains low at j is less than 30%. The matrices indicate that most state transitions cluster around zero, with strong diagonal probabilities more evident in the training set than in the test set. This suggests that the MTF captures richer structural information during training than during testing.

4.3.2. CNN-LSTM model performance

The final CNN-LSTM model was trained, validated and tested on the bitcoin dataset (the same split as above model) however the optimal window size for CNN is 224 (compared with 120 in 4.2.), resulting in lower MSE. The CNN captures more spatial features hence a window – 224 and image size of 224x224 provides sufficient information compared with tested windows of 120/200. The model; VGG was pre-trained with 'imagenet' weights with 2 convolution layers of size 128,64 with a dropout rate of 0.29 i.e. 29% of neurons 'turned off' prior to feeding into 128-dimension dense layer resulting in a feature vector concatenated with the LSTM output as in table below. The model's training converges around epoch 20 similar to the LSTM at ~0.002 train and 0.02 validation MSE. Whilst the training loss converges and stabilises, the validation loss remains high similar to section 4.2. The model is not generalise well in spite of the CNN. The GAF/MTF information relating to price directional/temporal changes, transition probability; the validation error is x10 higher than the training. The validation set starts around i~1,250th – it can be observed from the GASF that the strong negative correlation means that the price direction of the validation set is opposite most of the early training set, this is in line with the visualisation in 4.1. where the validation set was during a bull/bear market (sharp sustained increased and sharp sustained decrease in a short period of time). The GADF shows a strong positive temporal lead/lag relationship i.e. since i leads j, if price is increasing at i, it follows at j. However, in both gasf and gadf for i,j∈[1,250-1,400], there is a mix of no/negative correlation and no/strong positive temporal relationship which explains why the GAF may not providing sufficient directional/temporal information to predict the drastic market shifts. The MTF for i,j∈[1,250-1,400] shows high state transition probability (>0.7) which helps the model determine if the price is increasing, the state at at j is likely increasing however there are 'blind spots' where the probability is 0%. The high volatility and 'boom-bust' feature in the bitcoin currency yields a low validation performance in spite of CNN-LSTM architecture.

	Model	**Values**
CNN	VGG Input	(224, 224, 3)
	Activation Function	ReLU
	Convolution Layers	[128,64]
	Dropout rate	0.29145
	Dense layer	128
Hybrid	LSTM Input size	224
	Optimiser	RMSprop
	Learning Rate	0.00040
	Batch	64
	Epochs	150
Parameters	Window size ~ number of values per sequence	224
	Image size ~ sampled from sequence for DxD image input GAF/MTF	224x224
	Train sample size – 65% (*len-window*)	1,139
	Validation sample size – 10% (*len-window*)	175
	Test sample size – 25% (*len-window*)	439

Table. Model structure and hyperparameters

For visualisation purposes, the loss at epoch 0 was removed from the training/validation curves to avoid eclipsing the loss values (init. value x100 higher). The error metrics in the table below show the model R^2 is 46.71% ± 13.73% which is 3% lower than LSTM however the significant decrease in deviation signals more robustness. The train mse is 0.00016±0.00003, validation 0.01162±0.0043 and test mse 0.00694±0.0018. The train:validation:test ratio is 1:100:10 which is indicative of poor generalisability and low test accuracy. The model is overfitting on the training data however yielding low predictive accuracy compared with LSTM.

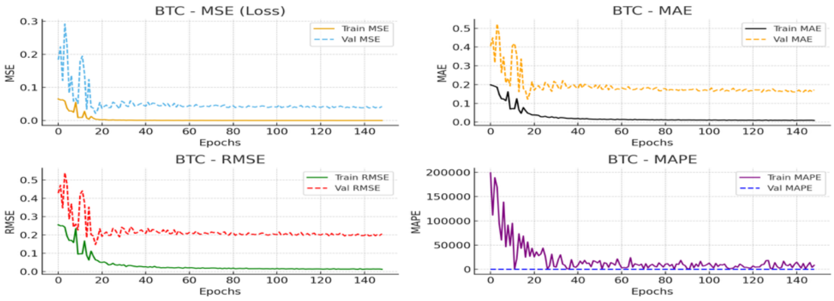


Figure. BTC – Training and validation curves for optimised CNN-LSTM

Mean ± std. deviation	MSE			RMSE			MAE			MAPE (%)			R^2 Score
	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
BTC	0.000166 ± 0.000032	0.01162 ± 0.00425	0.00694 ± 0.00179	0.01282 ± 0.00128	0.10640 ± 0.02102	0.08286 ± 0.01047	0.00902 ± 0.00069	0.08823 ± 0.01749	0.06990 ± 0.00982	2672.72 ± 1776.78	12.9317 ± 2.2639	0.2246 ± 0.0299	**0.4671** ± **0.1373**

Table. Error Metrics – MSE, RMSE, MAE, MAPE and R^2 for optimised BTC model

4.3.3. Comparative analysis with initial model

The CNN-LSTM architecture was first implemented following Vidal et al., producing relatively high errors with an MSE of approximately 1.8×10^{-2} on the training set and 9.14×10^{-1} on the test set, indicating poor learning and predictive performance. To address this, the CNN component was tuned independently using Optuna, as the LSTM had already demonstrated satisfactory results. This tuning led to notable improvements, reducing the training and test MSE to 2.6×10^{-3} and 4.49×10^{-1}, respectively (see table). The corresponding training curves (see Figure) show convergence of MSE around epoch 20, with stability thereafter apart from a transient peak near epoch 15 likely attributable to noise. Similarly, RMSE converges at approximately 0.02 and MAE at 0.025, both reflecting improvements over the initial model. Nevertheless, despite these enhancements, the CNN continues to underperform relative to the LSTM.

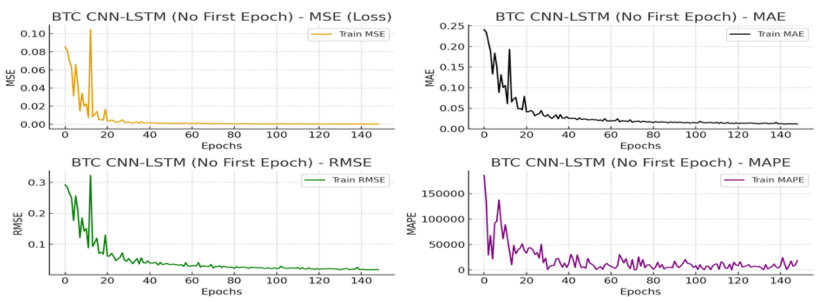


Figure. BTC – Training performance for a tuned CNN

Model	**Train MSE**	**Test MSE**
CNN-LSTM baseline	0.0018	0.09138
Tuned CNN	0.00026	0.04487
Optimised CNN-LSTM	0.00017	0.00694

Table. Comparison of training and testing loss across BTC models

4.3.4. Hyperparameter Tuning

Hyperparameter optimisation was carried out using Optuna (Bayesian optimisation) as described in Section 4.2. A total of 30 trials were executed across three runs (90 trials in total), including pruning and early stopping to minimise MSE based on validation performance. The optimal configuration, detailed in table below, was selected from these experiments. Following each run, the best-performing configuration was re-evaluated and the corresponding validation loss was recorded. From four evaluations, the configuration yielding the lowest validation loss was retained as the final model.

Hyperparameter	**Range of values tested**	**Optimised Parameter Value**
Window Size/Image size	[120, 200, 224]	224
Activation Function	[relu, tanh]	'relu'
Convolution Layers	[2, 3, 4]	2
Size of Conv2D	[32, 64, 128] [16, 32, 64]	[128, 64]
Dense layer	[64, 128, 256]	128
Dropout rate	[0.0, 0.5]	0.29145153405423596
Optimizer	[Adam, RMSprop]	RMSprop
Learning rate	[1e-5, 1e-3]	0.0004047368844437619
Batch Size	[16, 32, 64, 128]	64
Epochs	[30, 200]	150

4.3.5. Predictive performance

The figures below compare model predictions with ground truth values. The baseline CNN-LSTM (left) exhibited very poor performance, failing to capture market shifts and producing a stagnating trend until 2023 followed by a sharp decline. The optimised CNN-LSTM (right) improved predictive accuracy, with estimates closer to actual values, though large regime shifts remained challenging. The standalone CNN demonstrated limited accuracy, consistently underestimating price levels but successfully reflecting regime shifts and broader trends. Finally, the optimised CNN-LSTM (best seed) achieved the highest performance, closely approximating actual price magnitudes and reaching R^2>50% (see Appendix A,B for full results).

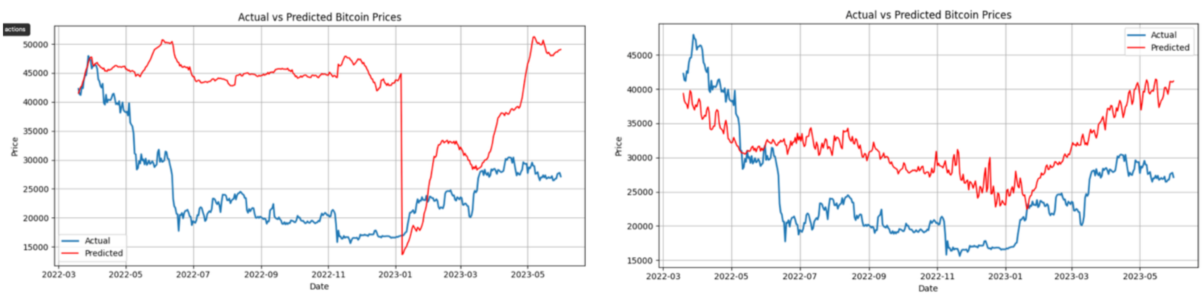
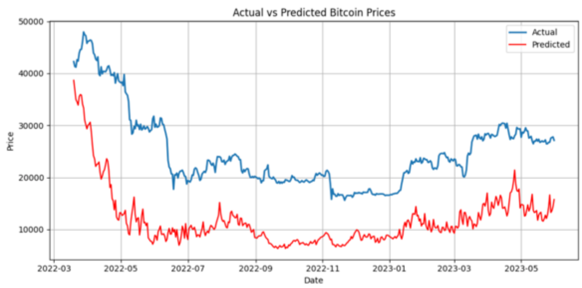


Figure. Original CNN-LSTM (left) and optimised CNN-LSTM (right) price prediction vs. actual



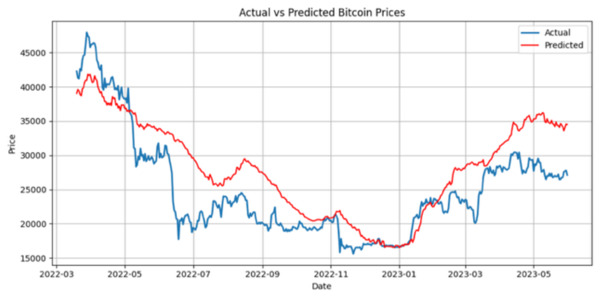


Figure. Tuned CNN model price prediction vs. actual (seed:42)

Figure. Bitcoin –Price prediction vs. actual

4.3.6. Comparative analysis of BTC, ETH, LTC and XRP performance

Mean ± std. deviation	**MSE**			**RMSE**			**MAE**			**MAPE (%)**			**R² Score**
	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
BTC	0.000166 ± 0.000032	0.01162 ± 0.00425	0.00694 ± 0.00179	0.01282 ± 0.00128	0.10640 ± 0.02102	0.08286 ± 0.01047	0.00902 ± 0.00069	0.08823 ± 0.01749	0.06990 ± 0.00982	2672.72 ± 1776.78	12.9317 ± 2.2639	0.2246 ± 0.0299	**0.4671 ± 0.1373**
LTC	0.000151 ± 0.000020	0.012829 ± 0.009323	0.00109 ± 0.00067	0.01225 ± 0.00080	0.10787 ± 0.04230	0.03203 ± 0.01009	0.00905 ± 0.00067	0.08050 ± 0.03269	0.02574 ± 0.00733	3549.791 ± 2345.899	18.98416 ± 6.47643	0.175007 ± 0.045331	**0.6681 ± 0.2033**
ETH	0.000120 ± 0.000040	0.06429 ± 0.05314	0.00823 ± 0.00296	0.01087 ± 0.00175	0.23005 ± 0.13060	0.08961 ± 0.01744	0.00740 ± 0.00116	0.19966 ± 0.12039	0.07015 ± 0.01047	3936.6507 ± 1627.8380	25.9167 ± 14.5832	0.21581 ± 0.02841	**0.4014 ± 0.2152**
XRP	0.000133 ± 0.000014	0.00537 ± 0.00304	0.000538 ± 0.000203	0.01153 ± 0.00060	0.07084 ± 0.02287	0.02291 ± 0.00445	0.00779 ± 0.00052	0.05594 ± 0.02087	0.01611 ± 0.00225	12234.336 ± 3546.669	18.00298 ± 6.46604	0.13667 ± 0.01538	**0.7604 ± 0.0906**

Table. Error Metrics as mean ± standard deviation for 4 cryptocurrencies

The comparative performance of the CNN-LSTM across ETH, LTC, and XRP reveals notable differences. The highest predictive accuracy was achieved for XRP, with R^2=76.04%±9.0%, followed by LTC at 66.81%±20%, while ETH and BTC attained substantially lower values of 40.14%±21.5% and 46.71%±13.7%, respectively. Across all currencies, validation errors exceeded training errors, with XRP displaying the most stable behaviour (validation loss ~10× training), whereas LTC and ETH exhibited far larger discrepancies (~100×), with ETH recording the highest validation loss. XRP also consistently achieved the lowest test errors. Nevertheless, test MAPE values remained elevated: XRP 13.7%, LTC 17.5%, ETH 21.6%, and BTC 22.5% (standard deviations ~1–4%). For BTC and ETH, the >20% deviation from actual values indicates substantial prediction error, which is critical given the high-risk nature of cryptocurrency trading. Even in the most stable case (XRP), CNN-LSTM performance was inferior to LSTM (13% error), suggesting limited generalisation of the hybrid architecture for cryptocurrency markets. For BTC and ETH, this limitation is likely exacerbated by their pronounced volatility and extreme price swings.

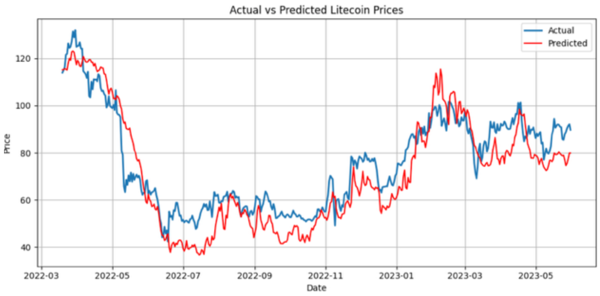


Figure. Litecoin –Price prediction vs. actual (seed:91) **Figure.** Ethereum –Price prediction vs. actual (seed:45)

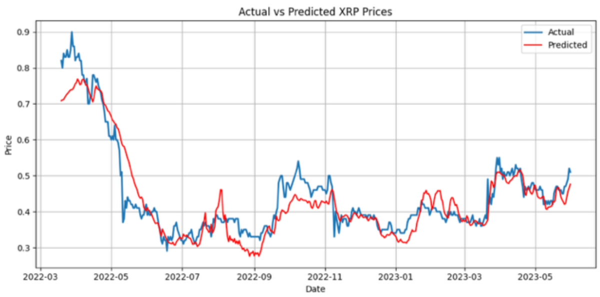


Figure. XRP –Price prediction vs. actual (seed:45)

The figures below present the price prediction performance of the best CNN-LSTM models for each currency. The LTC model demonstrates improved accuracy in magnitude prediction relative to BTC and captures market shifts, though still less effectively than the standalone LSTM. XRP achieves the highest performance, closely reproducing both price levels and general market dynamics. In contrast, ETH exhibits the weakest performance, with the model failing to predict prices or regime shifts, indicative of overfitting and poor generalisation. Contrary to expectations, the CNN feature extraction does not appear to enhance LSTM performance by contributing additional temporal or directional information. While Vidal et al. reported an 18% reduction in MSE over LSTM and errors on the order of 10^{-8} for gold prices, these results show that even for the relatively stable XRP data, the CNN-LSTM hybrid fails to outperform the LSTM, suggesting that the benefits observed in less volatile markets such as gold do not readily transfer to cryptocurrencies.

4.4. LSTM-HMM

This section presents the results of the LSTM-HMM models in terms of training performance, comparison with the initial model, hyperparameter tuning, predictive accuracy, and cross-currency performance.

4.4.1. Model Training Performance

The LSTM-HMM model was trained, validated and tested using the same split as prior models with a window of 120 days as in 4.2. The 'GaussianHMM' was fitted on the training set with 3 states (n_components) as determined in this study. The posterior probabilities were inferred for each rolling sequence per sample (H_train, val, test). The 64-dimension feature vector output of the HMM was concatenated with the output of the LSTM to predict price y_{t+1}. The training converges around 25^{th} epoch at around ~0.0005 train MSE, ~0.002 validation MSE at epoch 60. The RMSE, MAE training curves also converge and stabilise at e=20 ~0.02 train, ~0.04 validation MSE; ~0.015 train, ~0.03 validation MSE respectively. The LSTM-HMM model performs better than the CNN-LSTM with a lower validation error albeit still x10 larger than the training error.

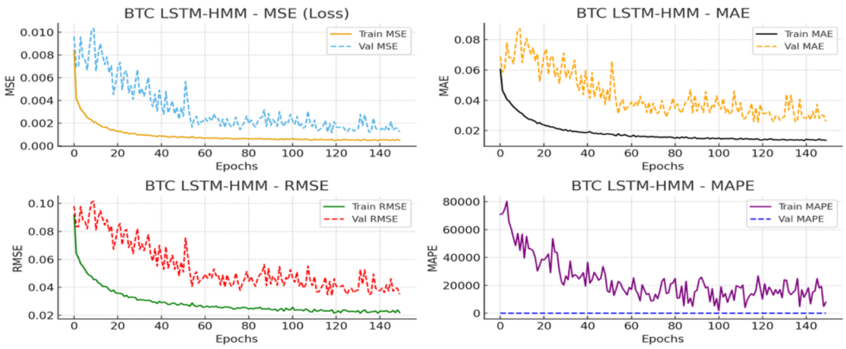


Figure. BTC – Training and validation curves for optimised LSTM-HMM

From the table below, the testing error of 2.8 ± 0.6 x10^{-4} is smaller than the training MSE in contrast with the previous model. This shows that the LSTM-HMM generalises well to the unseen data and captures the fast and drastic market shifts. This is consistent across the error metrics RMSE, MAE and test MAPE shows a 4% error in predictions. Lastly, the model's R^2 is 93.27% ± 1.5% which is very high, with low deviation; the model explains >93% of the variance. This is the highest performance for BTC prediction of all three models.

Mean ± std. deviation	**MSE**			**RMSE**			**MAE**			**MAPE (%)**			**R² Score**
	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
BTC	0.000481 ± 0.000021	0.001387 ± 0.000239	0.00028 ± 0.00006	0.02194 ± 0.00047	0.03715 ± 0.00320	0.01679 ± 0.00182	0.01393 ± 0.00023	0.02815 ± 0.00277	0.01211 ± 0.00209	14,482.03 ± 9,469.416	3.714 ± 0.266	0.04086 ± 0.00674	**0.9327** ± **0.0148**

Table. Error Metrics – MSE, RMSE, MAE, MAPE and R² for optimised BTC model

4.4.2. Comparison with initial approach

Initially, the GaussianHMM was used to infer the transition probabilities between the hidden states and the emission parameters i.e. mean and variance of Gaussian distributions for each state. The sequence of states with the highest likelihood was inferred and used the input into the HMM with one-hot encoding. The output is then concatenated with the LSTM's through a regression layer. This model was replaced with the LSTM-HMM in 4.4.1. as the computation of state sequence leads to data leakage. The model 'observes' the state transition information during training, validation, testing that is outside the sample. Even if the window input to predict the state sequence is sliced into train/val/test, the model's state transition sequence does not align with the LSTM's input leading to poorer learning performance with ~0.0011 test MSE and R^2=75.1% compared with ~0.00028 and 93.3% in the final model. As such, the causal HMM with posterior probabilities only dependent on the previous sequence was retained (see Appendix B for original HMM training curve).

<u>Hyperparameter Tuning</u>

Similar to the two models, Optuna (Bayesian optimisation)was used to optimise the hyperparameters. The same number of trials was conducted with three runs, pruning. Please see below table outlining the most optimal model configuration. The activation function for HMM is ReLU and for LSTM Tanh.

Hyperparameter	**Range of values tested**	**Optimised Parameter Value**
Window Size	[90, 120, 224]	120
Hidden Layers	[2, 3, 4]	3

Units per Layer	[16, 32, 64, 128] [8, 16, 32, 64] [2, 4, 8, 16]	[64, 8, 8]
Dense Layer	[4, 8, 32, 64] [4, 8, 32, 64]	[64, 64]
Dropout rate	[0.0, 0.5]	0.3345431
Learning rate	[1e-5, 1e-2]	0.0009718
Activation Function	[‘relu’, ‘tanh’]	[‘relu’-hmm ; ‘tanh’-lstm]
Optimizer	[‘adam’, ‘rmsprop’]	RMSprop
Batch Size	[16, 32, 64, 128]	16
Epochs	[30, 200]	150

4.4.4. Predictive performance

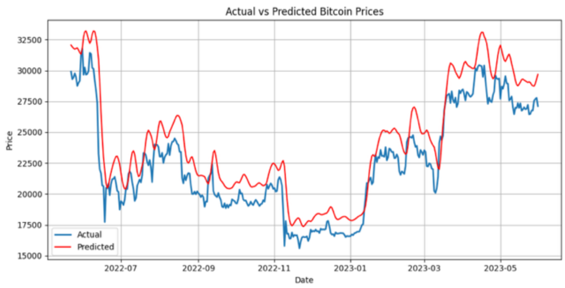


Figure. BTC – Original LSTM-HMM model (seed:91)

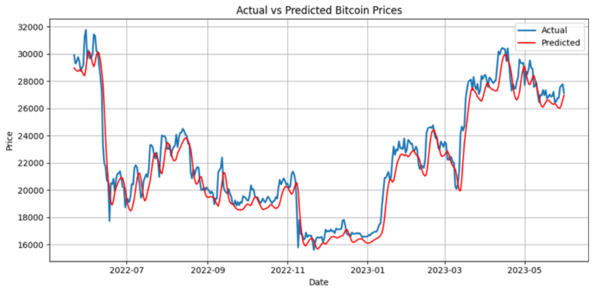


Figure. BTC – Optimised model price prediction

The initial model prediction was quite high as per the left figure, however, the bitcoin price prediction graph of the optimised model, exhibits a higher accuracy. The price magnitude difference between actual and prediction is minimal and the model captures the market shifts almost exactly. By incorporating state transition probabilities, the LSTM-HMM effectively captures volatility with negligible phase shift, a task particularly challenging for Bitcoin given its high variability. The GaussianHMM further identifies distinct neutral, bull, and bear regimes, enhancing interpretability of the predictive dynamics.

4.4.5. Comparative analysis of BTC, ETH, LTC and XRP performance

The LSTM-HMM demonstrates strong performance across all four cryptocurrencies, with test MSEs $(1.3\pm0.2)\times10^{-4}$ for XRP, $(2.4\pm1.9)\times10^{-4}$ for LTC, $(5.1\pm1.4)\times10^{-4}$ for ETH and $(2.8\pm0.6)\times10^{-4}$ for BTC.

These are lower than in previous models and below training error except for ETH (3.2×10^{-4}) concluding that the model generalises well for the currencies. In terms of R^2, BTC achieved the highest score at 93.27%±1.5% followed by LTC 88.82%±8.9%, ETH 83.83%±4.5% and XRP, the lowest at 71.75%±4.9%. Despite its stability, XRP underperformed compared with earlier models, whereas the more volatile BTC and ETH achieved >80% variance explanation. Test MAPE confirms robust generalisation, with BTC lowest at 4.1%, LTC at 5.7%, and ETH and XRP around 9.5% erroneous predictions, underscoring consistent accuracy across assets.

Mean ± std. deviation	**MSE**			**RMSE**			**MAE**			**MAPE (%)**			**R^2 Score**
	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
BTC	0.000481 ±	0.001387 ±	0.00028 ±	0.02194 ±	0.03715 ±	0.01679 ±	0.01393 ±	0.02815 ±	0.01211 ±	14,482.03 ±	3.714 ±	0.04086 ±	**0.9327** ±

	0.000021	0.000239	0.00006	0.00047	0.00320	0.00182	0.00023	0.00277	0.00209	9,469.416	0.266	0.00674	**0.0148**
LTC	0.000585 ± 0.000037	0.001508 ± 0.000157	0.00024 ± 0.00019	0.02418 ± 0.00076	0.03880 ± 0.00206	0.01461 ± 0.00580	0.01580 ± 0.00048	0.02674 ± 0.00166	0.01210 ± 0.00616	9,792.855 ± 5,393.558	5.909 ± 0.390	0.095 ± 0.046	**0.8882 ± 0.0893**
ETH	0.000328 ± 0.000016	0.003390 ± 0.001739	0.00051 ± 0.00014	0.01811 ± 0.00045	0.05705 ± 0.01426	0.02245 ± 0.00316	0.01116 ± 0.00013	0.04701 ± 0.01336	0.01706 ± 0.00319	8.084.199 ± 8,499.044	5.752 ± 1.413	0.057 ± 0.011	**0.8383 ± 0.0450**
XRP	0.000369 ± 0.000003	0.000491 ± 0.000089	0.00013 ± 0.00002	0.01921 ± 0.00009	0.02212 ± 0.00201	0.01153 ± 0.00100	0.01164 ± 0.00020	0.01813 ± 0.00184	0.00933 ± 0.00073	11,470.393 ± 4,196.052	6.539 ± 0.918	0.097 ± 0.009	**0.7175 ± 0.0489**

Table. Error Metrics as mean ± standard deviation for 4 cryptocurrencies

The prediction graphs corroborate the error metrics, indicating high accuracy in value prediction and effective capture of market shifts, with only minor phase lags for LTC and XRP. XRP performs less reliably at lower price magnitudes (≈0.5), where deviations from the ground truth are more pronounced. By contrast, Bitcoin attains the strongest performance under LSTM-HMM, underscoring the benefit of GaussianHMM in modelling volatility via latent state transitions. This regime-based representation enables adaptation to sharp fluctuations characteristic of highly volatile assets such as BTC and ETH, while providing limited gains for more stable, low-magnitude series like XRP, where regime dynamics are weaker.

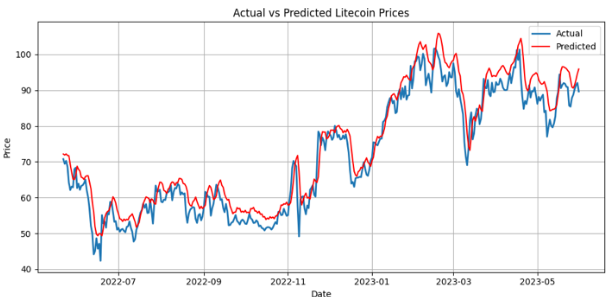


Figure. Litecoin–Price prediction vs. actual (seed:45)

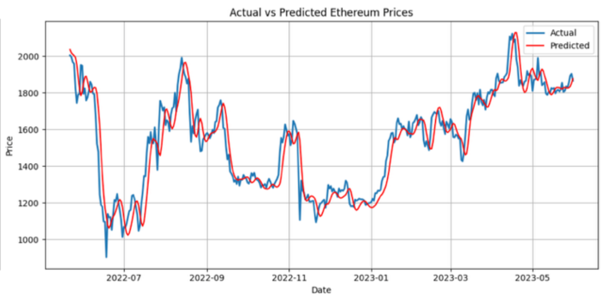


Figure. Ethereum–Price prediction vs. actual (seed:91)

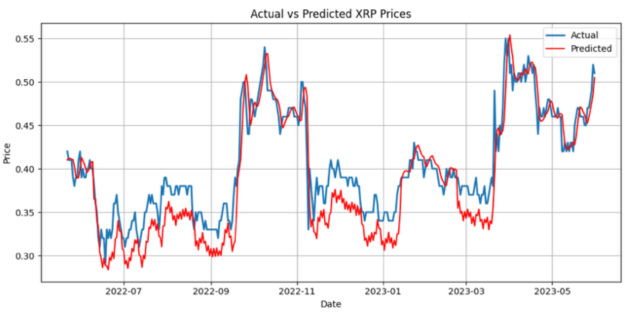


Figure. XRP–Price prediction vs. actual (seed:42)

4.5. Comparison

The table below summarises the test MSE across all cryptocurrencies for the three models. This allows for a clear conclusion: BTC, LTC and ETH perform better on unseen data with LSTM-HMM than other models with a small standard deviation. XRP performs better with the LSTM with negligible deviation.

Model	**MSE**			
Currency	**BTC**	**LTC**	**ETH**	**XRP**
LSTM	$2.14 \times 10^{-3} \pm 1.65 \times 10^{-3}$	$0.096 \times 10^{-3} \pm 0.005 \times 10^{-3}$	$0.529 \times 10^{-3} \pm 0.127 \times 10^{-3}$	$0.052 \times 10^{-3} \pm 0.001 \times 10^{-3}$
CNN-LSTM	$6.94 \times 10^{-3} \pm 1.79 \times 10^{-3}$	$1.09 \times 10^{-3} \pm 0.67 \times 10^{-3}$	$8.23 \times 10^{-3} \pm 2.96 \times 10^{-3}$	$0.538 \times 10^{-3} \pm 0.203 \times 10^{-3}$
LSTM-HMM	$0.28 \times 10^{-3} \pm 0.06 \times 10^{-3}$	$0.24 \times 10^{-3} \pm 0.19 \times 10^{-3}$	$0.51 \times 10^{-3} \pm 0.14 \times 10^{-3}$	$0.13 \times 10^{-3} \pm 0.02 \times 10^{-3}$

Table. Test MSE for BTC, LTC, ETH and XRP for all three models

4.6. Regime shift

(See Appendix B. for LTC, ETH and XRP graphs). This simple classifier serves as a diagnostic such that models that (a) match regime colours and (b) minimise phase shifts are judged to better capture market shifts. Since the volatility quantiles are based on the original set, deviations in prediction result in poor regime detection.

<u>LSTM Model</u>

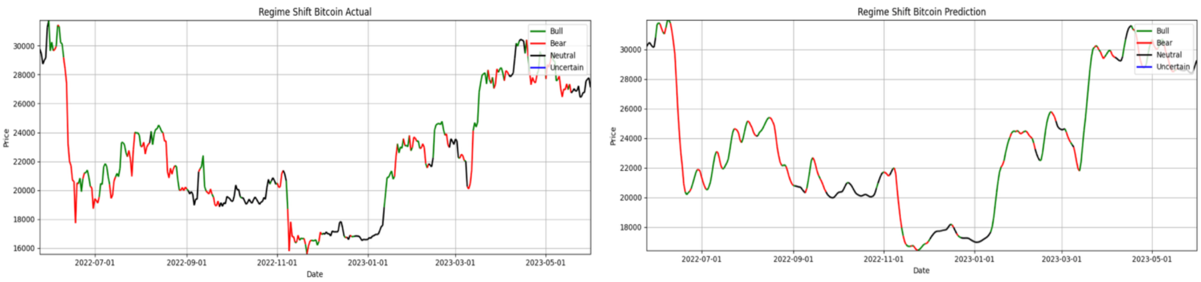


Figure. Bitcoin – Regime Shift Classification Actual (left) and Prediction (right) ; seed:42

The predicted series exhibit smoothing that stabilises noise but delays regime onsets. Bear markets are detected more reliably than bull. For BTC/ETH, high volatility amplifies small prediction errors, producing neutral mislabels near emerging bull runs. XRP shows the best alignment: bull periods (10/2022, 03/2023) and the 11/2022 drawdown are correctly labelled. LTC underperforms—early-2023 bull is often neutral and the post-FTX bear only partly identified. Overall regime fidelity ranks XRP > BTC/ETH > LTC, reflecting asset volatility and the classifier's windowed averaging.

<u>CNN-LSTM Model</u>

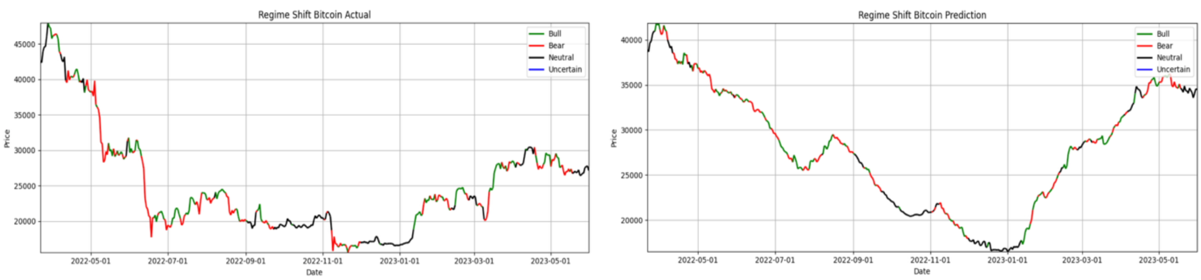


Figure. Bitcoin – Regime Shift Classification Actual (left) and Prediction (right) ; seed:42

The CNN-LSTM underperformed, and regime classification reflects this. Predicted series are overly smoothed and fragmented, yielding short bull/bear segments that misrepresent sustained regimes; the plots are weak indicators of market states. Across currencies, classification accuracy is poor: labels follow local directionality rather than persistent trends, consistent with error metrics indicating inadequate volatility modelling. For ETH, the original data show one credible shift (04/22–07/22), yet the prediction fails to recover meaningful regimes. LTC's original series captures mainly the 2022 bear; predictions for LTC and XRP mislabel transient moves as regimes.

<u>LSTM-HMM Model</u>

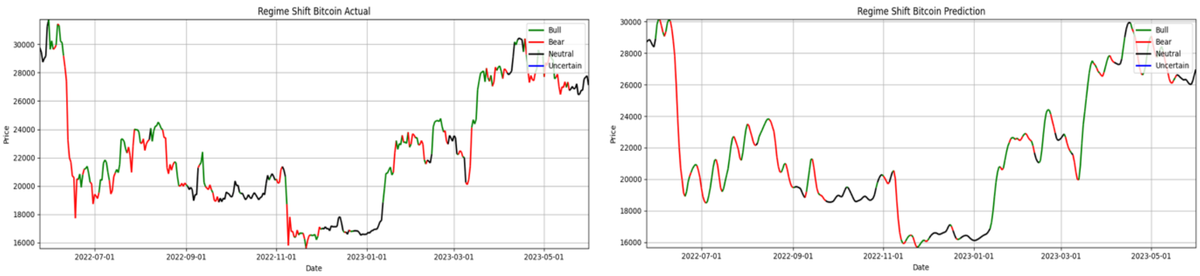


Figure. Bitcoin – Regime Shift Classification Actual (left) and Prediction (right) ; seed:91

The LSTM-HMM delivers substantially higher accuracy than other models. For BTC, it achieves the best performance, with regimes clearly delineated and major bullish phases identified. Short bear spells are harder to detect, often classified as neutral due to small return changes, and a mild smoothing effect remains. For LTC, the classifier mislabels early-2023 bull periods (01/23–03/23; 04/23–05/23) as neutral. For ETH, smoothing aids bear detection, though the early-2023 bull is misclassified. XRP's noisier series yields mixed results, despite correct bull labels in 09/2022 and early-2023.

5. Discussion

This chapter evaluates whether the results meet the stated objectives, answers the research questions and hypotheses, and positions the findings within the wider literature.

5.1. Results & objectives

Objective 1 – LSTM baseline (regression)
The LSTM achieved satisfactory accuracy overall on daily data (2018–2023). Test performance ranked XRP > LTC > ETH > BTC, with highest R^2 for LTC (≈95.5%) and XRP (≈88.9%), and weakest for BTC (≈49.4%, high variance). The success criterion - LSTM improving on García-Medina's MSE ≈ 0.01- is met; all currencies' test MSE are lower. However, generalisation is asset-dependent (poor on BTC). **RQ1:** LSTM can predict daily crypto prices through volatile periods but with uneven robustness across assets.

Objective 2 – CNN-LSTM with GAF/MTF
The success criterion (≥17% MSE reduction vs LSTM) is not met. CNN-LSTM test MSEs are ≈10× higher than LSTM in several cases; validation losses are high and unstable. GAF/MTF encodings and the CNN did not yield the anticipated gains from gold-volatility studies, suggesting weak transfer to daily crypto price-level prediction under multi-year regime cycling. **RQ2:** Using GAF/MTF with a CNN did not reduce error nor improve regime fidelity; H2 rejected.

Objective 3 – LSTM-HMM
The LSTM-HMM attained the lowest test MSE and highest R^2 overall (BTC ≈ 93% R^2) with small deviation across seeds, indicating robustness. The ≥17% reduction vs LSTM is partially met: clear gains on BTC/ETH(and often LTC), but XRP does not consistently improve over LSTM. **RQ3:** Adding a Gaussian HMM improves performance on the most volatile assets (BTC/ETH), consistent with the mechanism of state persistence and improved timing at regime boundaries; H1 partially accepted

Objective 4 – Best model and literature benchmarks
Across currencies, LSTM-HMM is the best performer and meets the external benchmarking criterion i.e. outperforms literature baselines as listed below on BTC and, in most settings, on other assets. Thus Objective 4 is met for the LSTM-HMM.
From the literature, a comparison shows that for Gautam, 2025, the hybrid XGBoost-LSTM yielded $MAPE_{Test}$ =0.0488 and $RMSE_{Test}$ =0.0659. In comparison, Zhang et al., 2024, obtained best $RMSE_{Test}$ =0.020 for conv-LSTM model for BTC. Lastly, Sivakumar, 2024, obtains R2 =0.803 for the HMM-LSTM (other metric not comparable since not normalised). This study's model LSTM-HMM outperforms the above literature, for BTC, $RMSE_{Test}$ =0.01679±0.00182, $MAPE_{Test}$=0.04086±0.00674 and R^2 =93%

5.2. Results in the context of the literature

Findings align with studies showing deep models outperform classical volatility families under stress and non-stationarity. The LSTM baseline's mixed generalisation echoes evidence that architecture is insufficient without regime awareness. The CNN-LSTM with GAF/MTF contradicts results from gold volatility: while image encodings improved CNN-LSTM for gold, they did not translate to daily crypto price-level prediction across multi-year regimes. Likely causes include (i) heavier tails and sharper regime flips in crypto, (ii) weaker stationary motifs for convolutional extraction, and (iii) mismatch between image receptive fields and crypto periodicities. The LSTM-HMM results are consistent with regime-aware sequences in macro/finance: latent-state decoding reduces phase lag, improving alignment through bull/bear transitions. The lowest error is for BTC/ETH as intuitively, data with more volatility and peaks make state inference more informative. The XRP's performance suggests that when magnitudes are small then regime boundaries are unclear. HMM signals then add less incremental value to the LSTM's performance.

5.3 Robustness, validity, scope

Design strengths

- Chronological splits (65/10/25) and multiple random seeds reduce optimistic bias and quantify variance.
- Consistent preprocessing and uniform evaluation across assets enable fair comparisons.
- A rule-based regime labelling procedure (volatility quantiles combined with a short-horizon directional signal) offers a model-agnostic check of regime fidelity.

Validity

- A small, volatile validation set inflates variance and may undermine the reliability of hyperparameter tuning.
- MAPE is unstable near zero (especially after min–max scaling), so inference should prioritise MSE/RMSE/MAE and R^2.
- HMM state count and initialisation introduce variance; despite mitigation with multi-seed training and EM iterations, local optima remain possible.

Scope – Results are most transferable to daily horizons for BTC, ETH, LTC, and XRP under 2018–2023-type regimes characterised by pronounced market shifts. Application to intraday settings will require recalibration. LSTM-HMM is particularly effective for high-volatility assets with persistent regime spells; gains are smaller where price magnitudes are low.

5.4 Implications and recommendations

Implications

- For deployment on volatile assets, LSTM-HMM should be preferred to reduce timing errors around regime transitions.
- CNN-LSTM with GAF/MTF is not recommended for daily crypto price-level prediction without substantial redesign of features and architecture; advantages observed in the gold market do not transfer directly.

Recommendations

- Validation protocol: enlarge or stratify the validation window, cross-validation to stabilise tuning under regime churn.
- CNN-LSTM revision: if retained, revisit image construction (adaptive windows, multi-scale GAF/MTF, alternative quantizers) and consider temporal CNNs on raw sequences rather than image proxies.

In conclusion, the objectives are largely satisfied: LSTM-HMM emerges as the most dependable approach for daily cryptocurrency prices across major regime shifts; the LSTM baseline is competitive but asset-sensitive; and CNN-LSTM (GAF/MTF) does not generalise to daily crypto as anticipated. With stronger validation and continued regime-aware modelling, the framework is positioned for robust, real-world forecasting.

6. Evaluation, Reflections, and Conclusions

6.1. Evaluation

This project set out to forecast daily prices for BTC, ETH, LTC, and XRP (2018–2023) and to assess robustness under regime shifts using LSTM, CNN-LSTM (GAF/MTF), and LSTM-HMM. The objectives were largely achieved. The literature review established a sound rationale for deep sequence models and regime awareness; methods were implemented with chronological splits, multiple seeds, and consistent preprocessing to support fair comparison. Empirically, LSTM provided a competitive baseline but exhibited asset-dependent performance (strong on LTC/XRP, weaker on BTC). CNN-LSTM underperformed the baseline across currencies, contrary to expectations from gold-volatility studies. LSTM-HMM yielded the best overall accuracy and stability, generalising on three of four assets and reducing phase lag around regime changes. Planning was adequate although the validation set size constrained tuning fidelity.

6.2. Conclusions and contribution

In conclusion, the LSTM-HMM is the most reliable model for daily crypto price prediction across multi-year, market shifts, delivering lower error and higher (R^2) for BTC/ETH/LTC and competitive results on XRP. The rule-based regime classifier—though simple—proved useful as a diagnostic and, when paired with accurate predictions, produced coherent regime labelling that could be repurposed as a trading signal input. Conversely, the CNN-LSTM (GAF/MTF) hypothesis is rejected: adding image encodings did not translate into gains for daily price-level prediction in cryptocurrencies. The evidence suggests that (i) CNNs struggled to exploit crypto's noisy, weakly stationary structure on the daily horizon, and (ii) concatenation of CNN and LSTM features introduced representational mismatch rather than complementary signal. Overall, the project's contribution is a comparative, regime-aware benchmark showing that explicit latent-state modelling (LSTM-HMM) improves phase fidelity and out-of-sample robustness where volatility and regime persistence are pronounced.

Implications: For practitioners, regime-aware sequence models should be preferred for volatile assets; regime diagnostics should complement error metrics in model selection. For researchers, results caution against uncritical transfer of CNN-image encodings from commodities to crypto price levels and motivate hybrids that align representation with temporal state dynamics.

6.3. Future work

- Regime-to-policy: Integrate regime-aware forecasts into reinforcement learning to evaluate decision quality (risk-adjusted returns) under live regime switching; this requires larger, recent datasets and careful transaction-cost modelling (also adequate compute).
- Explainability: Add post-hoc and intrinsic XAI (e.g., LIME for LSTM-HMM states) to interpret drivers of regime transitions and forecast sensitivity.
- Sentiment/news: (e.g., FinBERT) and test marginal utility per asset; evaluate its contribution to regime switching capture.
- Model families: Compare LSTM-HMM with state-space models and switching**.**
- Transformers: on daily horizons, using the same regime-fidelity diagnostics.

6.4. Reflections (process, rationale, limitations)

Learning and process

Objectives and RQs were addressed systematically; results were interpreted against prior work; uncertainty was handled with seeds and regime diagnostics. In retrospect, three design choices merit revision.

- Task framing: Early codebases assumed classification; pivoting to regression was necessary and successful, but should have been planned explicitly with dual pipelines.
- CNN-LSTM design: The appeal of GAF/MTF (from gold) was strong, yet crypto's statistical properties differ materially. A better design would place CNN features upstream of a temporal learner (e.g., LSTM) or adopt multi-scale encodings rather than simple concatenation.
- Noise control: A Kalman filter was scoped for denoising but ultimately unnecessary given LSTM-HMM performance; nevertheless, an uncertainty-aware variant could further stabilise inference.

<u>Limitations</u>

- Feature parsimony (primarily Close) likely limited signal richness; future work should use multivariate inputs by concatenating different datasets
- Validation size and volatility inflated tuning variance; blocked/rolling CV would improve reliability
- Metric sensitivity: MAPE can be unstable near zero; primary reliance on RMSE/MAE/(R^2) is preferable
- Generalisability: Findings are strongest for daily input price; extension to intraday or post-2023 dynamics requires re-evaluation.

<u>Reflection</u>

- Plan dual task tracks (classification + regression) from inception;
- expand validation via rolling splits;
- prototype Conv-temporal hybrids before image concatenation;

In summary, this project demonstrates that latent-state, regime-aware sequence modelling outperforms architecture-only hybrids for daily cryptocurrency prices across turbulent years. LSTM-HMM offers a practical, interpretable path to robust forecasting, from average-fit accuracy to regime-consistent predictions that matter in deployment.

Word Count – 12,405

Word count is calculated based on Chapter 1 to 6 , cover page, table of contents, glossary/reference and appendices are excluded.

N.B. The source code from Andrés García-Medina and Aguayo-Moreno's paper (section 3.3., 4.2.) was not included in the codes provided in the Appendices below. I have also not uploaded it to the GitHub Repository as it is open to the public. Whilst, the code was implemented with this study's dataset as a benchmark, it is not my property hence I did not include it in my work, only the graphs obtained in section 4. and Appendix B.

Glossary

Term	Definition
Regime shift	A structural change in market conditions where return/volatility dynamics transition to a different, persistent state.
Bear	A sustained downward market phase characterised by falling prices
Bull	A sustained upward market phase characterised by rising prices
Neutral	A sideways market phase with no clear upward or downward trend
Momentum	The tendency for price to continue moving in its recent direction, often measured by rate and persistence of change
MACD	Moving Average Convergence Divergence: the difference between fast and slow EMAs, used with a signal line to gauge trend strength and reversals.
Log Returns	The natural log of the price ratio approximating continuous compounding

References

The website mybib.com was used to format the citations in Harvard Style.

Aanandhi, S.P., U, S.K., Akhilaa, S.P., Vardarajan, V. and Sathiyanarayanan, M. (2021). *Cryptocurrency Price Prediction using Time Series Forecasting (ARIMA)*. [online] IEEE Xplore.

Abba Suganda Girsang and Stanley, N. (2023). Hybrid LSTM and GRU for Cryptocurrency Price Forecasting Based on Social Network Sentiment Analysis Using FinBERT. *IEEE access*, 11, pp.120530–120540.

Aditya Mhaske (2018). *Cryptocurrency Price Analysis Dataset*. [online] Kaggle.com. https://www.kaggle.com/datasets/adityamhaske/cryptocurrency-price-analysis-dataset/data

agarciam (2022). *GitHub - agarciam/lstmgarch_project*. GitHub. Available at: https://github.com/agarciam/lstmgarch_project.

Amjad, M., Shah, D., Anava, O., Cuturi, M., Khaleghi, A., Kuznetsov, V. and Rakhlin, A. (2017). *Trading Bitcoin and Online Time Series Prediction*.

Andrés García-Medina and Aguayo-Moreno, E. (2023). LSTM–GARCH Hybrid Model for the Prediction of Volatility in Cryptocurrency Portfolios. *Computational Economics*.

Gautam, M. (2025). *crypto price prediction using lstm+xgboost*. [online] arXiv.org.

Ilyas Agakishiev, Wolfgang Karl Härdle, Becker, D. and ZUO, X. (2024). Regime switching forecasting for cryptocurrencies.

Khaniki, M.A.L. and Manthouri, M. (2024). *Enhancing Price Prediction in Cryptocurrency Using Transformer Neural Network and Technical Indicators*. arXiv.org.

Muhammad Mahmudul Karim, Md Hakim Ali, Yarovaya, L., Md Hamid Uddin and Hammoudeh, S. (2023). Return-volatility relationships in cryptocurrency markets: Evidence from asymmetric quantiles and non-linear ARDL approach. *International review of financial analysis (Online)/International review of financial analysis*, 90, pp.102894–102894.

MyBib (2025). MyBib. https://www.mybib.com/.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. *SSRN Electronic Journal*.

Rabiner, L. and Juang, B. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, [online] 3(1), pp.4–16.

Readthedocs.io. (2025). *API Reference — hmmlearn 0.3.3.post1+ge01a10e documentation*.

Salim, M. and Arif Djunaidy (2024). Development of a CNN-LSTM Approach with Images as Time-Series Data Representation for Predicting Gold Prices. *Procedia computer science*, 234, pp.333–340.

Sepehri, M.S., Mehradfar, A., Soltanolkotabi, M. and Avestimehr, S. (2025). *CryptoMamba: Leveraging State Space Models for Accurate Bitcoin Price Prediction*. arXiv.org.

Sivakumar, G. (2024). HMM-LSTM Fusion Model for Economic Forecasting. [online] arXiv.org.

Tang, S.H., Rosenbaum, M. and Zhou, C. (2023). *Forecasting Volatility with Machine Learning and Rough Volatility: Example from the Crypto-Winter*. [online] arXiv.org.

Vidal, A. and Kristjanpoller, W. (2020). Gold volatility prediction using a CNN-LSTM approach. *Expert Systems with Applications*, 157, p.113481.

Wang, Z. and Oates, T. (2015). Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks. *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Wu, J., Zhang, X., Huang, F., Zhou, H. and Chandra, R. (2024). *Review of deep learning models for crypto price prediction: implementation and evaluation*. arXiv.org.

Zhang, J., Wen, J. and Yang, Z. (2022). China's GDP forecasting using Long Short Term Memory Recurrent Neural Network and Hidden Markov Model. *PLOS ONE*, 17(6), p.e0269529.

Appendices

Appendix A – Results from Models

1. LSTM Results

Bitcoin

Seed	Best Epoch	MSE			RMSE			MAE			MAPE			R^2
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
42	27	0.000259	0.001320	0.000527	0.016100	0.036338	0.022952	0.00986	0.02658	0.01733	3137.5466	3.71412	0.057500	0.8752
45	105	0.000239	0.003260	0.003820	0.015475	0.057093	0.061806	0.00991	0.04727	0.04720	9449.7529	6.30585	0.152057	0.0952
91	36	0.000208	0.002380	0.002064	0.014439	0.048786	0.045427	0.00934	0.03800	0.03254	9314.5654	4.92539	0.103959	0.5112
Mean		0.000236	0.002320	0.002137	0.015338	0.047406	0.043395	0.00970	0.03728	0.03236	7300.6217	4.98179	0.104505	0.4939
± std. deviation		0.000026	0.000971	0.001648	0.000839	0.010446	0.019506	0.00031	0.01036	0.01494	3605.9623	1.29678	0.047281	0.3903

Litecoin

Seed	Best Epoch	MSE			RMSE			MAE			MAPE			R^2
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
42	119	0.000367	0.001339	0.000100	0.019147	0.036595	0.009976	0.01126	0.02537	0.00741	9040.8174	5.63680	0.05811	0.9528
45	117	0.000342	0.001367	0.000090	0.018504	0.036975	0.009479	0.01129	0.02519	0.00684	3990.4983	5.64525	0.05368	0.9574
91	119	0.000388	0.001384	0.000098	0.019704	0.037205	0.009916	0.01193	0.02512	0.00738	1185.0750	5.65690	0.05726	0.9534
Mean		0.000366	0.001364	0.000096	0.019118	0.036925	0.009791	0.01149	0.02523	0.00721	4738.7969	5.64631	0.05635	0.9545
± std. deviation		0.000023	0.000023	0.000005	0.000601	0.000308	0.000271	0.00038	0.00013	0.00032	3980.9717	0.01009	0.00235	0.0025

Ethereum

Seed	Best Epoch	MSE			RMSE			MAE			MAPE			R^2
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
42	18	0.000163	0.008576	0.000676	0.012752	0.092608	0.025991	0.00732	0.07572	0.01855	512.3653	9.07056	0.06420	0.7860

45	33	0.000168	0.005215	0.000456	0.012977	0.072215	0.021343	0.00707	0.06182	0.01646	1781.7650	7.38751	0.05475	0.8557
91	1	0.000177	0.014295	0.000455	0.013296	0.119561	0.021321	0.00722	0.10125	0.01484	1760.3755	11.63410	0.05048	0.8560
Mean		0.000169	0.009362	0.000529	0.013008	0.094795	0.022885	0.00720	0.07960	0.01661	1351.5019	9.36406	0.05647	0.8326
± std. deviation		0.000007	0.004591	0.000127	0.000273	0.023749	0.002690	0.00013	0.02000	0.00186	726.7923	2.13846	0.00702	0.0403

XRP														
Seed	Best Epoch	MSE			RMSE			MAE			MAPE			R^2
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
42	120	0.000232	0.000244	0.000054	0.015216	0.015615	0.007320	0.00811	0.01150	0.00511	1220.3114	4.02120	0.05235	0.8865
45	119	0.000204	0.000290	0.000052	0.014285	0.017035	0.007240	0.00837	0.01310	0.00485	9840.6787	4.51875	0.04962	0.8890
91	111	0.000219	0.000274	0.000051	0.014789	0.016542	0.007139	0.00844	0.01272	0.00480	7932.8926	4.46443	0.04958	0.8921
Mean		0.000218	0.000269	0.000052	0.014763	0.016397	0.007233	0.00831	0.01244	0.00492	6331.2942	4.33479	0.05052	0.8892
± std. deviation		0.000014	0.000024	0.000001	0.000466	0.000721	0.000091	0.00017	0.00083	0.00017	4527.8605	0.27293	0.00159	0.0028

2. CNN-LSTM Results

Bitcoin														
Seed	Best Epoch	MSE			RMSE			MAE			MAPE			R^2
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
42	7	0.000191	0.013170	0.005594	0.01384	0.11476	0.07479	0.00967	0.10171	0.06078	1302.27	15.3550	0.192666	0.5704
45	133	0.000188	0.019203	0.009744	0.01367	0.13857	0.09871	0.00975	0.10160	0.08457	8451.88	12.9136	0.282521	0.3113
91	22	0.000176	0.014874	0.006256	0.01325	0.12196	0.07909	0.00909	0.09451	0.06861	2035.68	12.5691	0.229289	0.5196
Mean		0.000166	0.01162	0.00694	0.01282	0.10640	0.08286	0.00902	0.08823	0.06990	2672.72	12.9317	0.2246	0.4671

± std. deviation		0.000032	0.00425	0.00179	0.00128	0.02102	0.01047	0.00069	0.01749	0.00982	1776.78	2.2639	0.0299	0.1373

Litecoin

Seed	Best Epoch	MSE			RMSE			MAE			MAPE			R^2
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
42	144	0.000133	0.022873	0.00183	0.01155	0.15124	0.04277	0.00847	0.11465	0.03327	2065.283	25.67935	0.221102	0.4450
45	138	0.000146	0.011163	0.00094	0.01209	0.10565	0.03057	0.00890	0.07734	0.02533	2329.796	18.52188	0.173438	0.7164
91	83	0.000172	0.004452	0.00052	0.01312	0.06673	0.02276	0.00979	0.04950	0.01862	6254.293	12.75126	0.130481	0.8429
Mean		0.000151	0.012829	0.00109	0.01225	0.10787	0.03203	0.00905	0.08050	0.02574	3549.791	18.98416	0.175007	0.6681
± std. deviation		0.000020	0.009323	0.00067	0.00080	0.04230	0.01009	0.00067	0.03269	0.00733	2345.899	6.47643	0.045331	0.2033

Ethereum

Seed	Best Epoch	MSE			RMSE			MAE			MAPE			R^2
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
42	144	0.000101	0.11197	0.01029	0.010070	0.33462	0.10144	0.006653	0.29955	0.07302	3685.416	38.54871	0.21521	0.2517
45	113	0.000166	0.00700	0.00484	0.01288	0.08367	0.06958	0.00873	0.06599	0.05855	5675.500	9.95737	0.187707	0.6479
91	17	0.000093	0.07391	0.00957	0.009660	0.27186	0.09780	0.006814	0.23344	0.07888	2449.036	29.2441	0.24452	0.3044
Mean		0.000120	0.06429	0.00823	0.01087	0.23005	0.08961	0.00740	0.19966	0.07015	3936.6507	25.9167	0.21581	0.4014
± std. deviation		0.000040	0.05314	0.00296	0.00175	0.13060	0.01744	0.00116	0.12039	0.01047	1627.8380	14.5832	0.02841	0.2152

XRP

Seed	Best Epoch	MSE			RMSE			MAE			MAPE			R^2
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
42	149	0.000121	0.00803	0.000744	0.01102	0.08961	0.02728	0.00742	0.07415	0.01759	10983.266	23.73230	0.13196	0.6685
45	148	0.000149	0.00206	0.000338	0.01219	0.04537	0.01838	0.00839	0.03318	0.01352	9482.745	10.99214	0.12421	0.8496

91	139	0.000129	0.00601	0.000532	0.01138	0.07752	0.02306	0.00758	0.06050	0.01723	16236.997	19.28449	0.15386	0.7632
Mean		0.000133	0.00537	0.000538	0.01153	0.07084	0.02291	0.00779	0.05594	0.01611	12234.336	18.00298	0.13667	0.7604
± std. deviation		0.000014	0.00304	0.000203	0.00060	0.02287	0.00445	0.00052	0.02087	0.00225	3546.669	6.46604	0.01538	0.0906

3. LSTM-HMM

Bitcoin														
Seed	Best Epoch	MSE			RMSE			MAE			MAPE			R^2
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
42	148	0.000493	0.001367	0.00036	0.02220	0.03698	0.01883	0.01417	0.02787	0.01448	4689.276	3.628	0.04853	0.9160
45	115	0.000494	0.001635	0.00026	0.02222	0.04044	0.01620	0.01391	0.03105	0.01132	15165.77	4.013	0.03821	0.9378
91	119	0.000457	0.001159	0.00024	0.02139	0.03404	0.01535	0.01371	0.02554	0.01052	23591.04	3.502	0.03585	0.9442
Mean		0.000481	0.001387	0.00028	0.02194	0.03715	0.01679	0.01393	0.02815	0.01211	14482.03	3.714	0.04086	0.9327
± std. deviation		0.000021	0.000239	0.00006	0.00047	0.00320	0.00182	0.00023	0.00277	0.00209	9469.416	0.266	0.00674	0.0148

Litecoin														
Seed	Best Epoch	MSE			RMSE			MAE			MAPE			R^2
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
42	178	0.000543	0.001328	0.00014	0.02330	0.03644	0.01169	0.01528	0.02482	0.00919	13647.997	5.458	0.075	0.9353
45	144	0.000605	0.001614	0.00012	0.02459	0.04017	0.01086	0.01623	0.02761	0.00792	12101.306	6.132	0.062	0.9441
91	148	0.000608	0.001583	0.00045	0.02465	0.03979	0.02128	0.01591	0.02779	0.01917	3629.263	6.137	0.148	0.7853
Mean		0.000585	0.001508	0.00024	0.02418	0.03880	0.01461	0.01580	0.02674	0.01210	9792.855	5.909	0.095	0.8882
± std. deviation		0.000037	0.000157	0.00019	0.00076	0.00206	0.00580	0.00048	0.00166	0.00616	5393.558	0.390	0.046	0.0893

Ethereum

Seed	Best Epoch	MSE			RMSE			MAE			MAPE			R^2
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
42	188	0.000312	0.002248	0.00050	0.01765	0.04741	0.02247	0.01107	0.03824	0.01714	2554.923	4.803	0.057	0.8401
45	133	0.000344	0.002531	0.00066	0.01856	0.05031	0.02560	0.01132	0.04040	0.02021	3827.152	5.077	0.068	0.7924
91	137	0.000328	0.005391	0.00037	0.01811	0.07342	0.01927	0.01110	0.06238	0.01383	17870.523	7.376	0.047	0.8824
Mean		0.000328	0.003390	0.00051	0.01811	0.05705	0.02245	0.01116	0.04701	0.01706	8084.199	5.752	0.057	0.8383
± std. deviation		0.000016	0.001739	0.00014	0.00045	0.01426	0.00316	0.00013	0.01336	0.00319	8499.044	1.413	0.011	0.0450

XRP														
Seed	Best Epoch	MSE			RMSE			MAE			MAPE			R^2
		Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	Train	Val.	Test	
42	139	0.000362	0.000490	0.00009	0.01904	0.02213	0.00955	0.01186	0.01738	0.00732	12890.084	5.891	0.077	0.8068
45	144	0.000367	0.000554	0.00012	0.01915	0.02354	0.01082	0.01150	0.01943	0.00881	14437.450	7.188	0.091	0.7521
91	145	0.000371	0.000428	0.00015	0.01927	0.02070	0.01224	0.01179	0.01683	0.00985	8503.336	5.889	0.104	0.6829
Mean		0.000369	0.000491	0.00013	0.01921	0.02212	0.01153	0.01164	0.01813	0.00933	11470.393	6.539	0.097	0.7175
± std. deviation		0.000003	0.000089	0.00002	0.00009	0.00201	0.00100	0.00020	0.00184	0.00073	4196.052	0.918	0.009	0.0489

Appendix B – Graph Results

1. Dataset

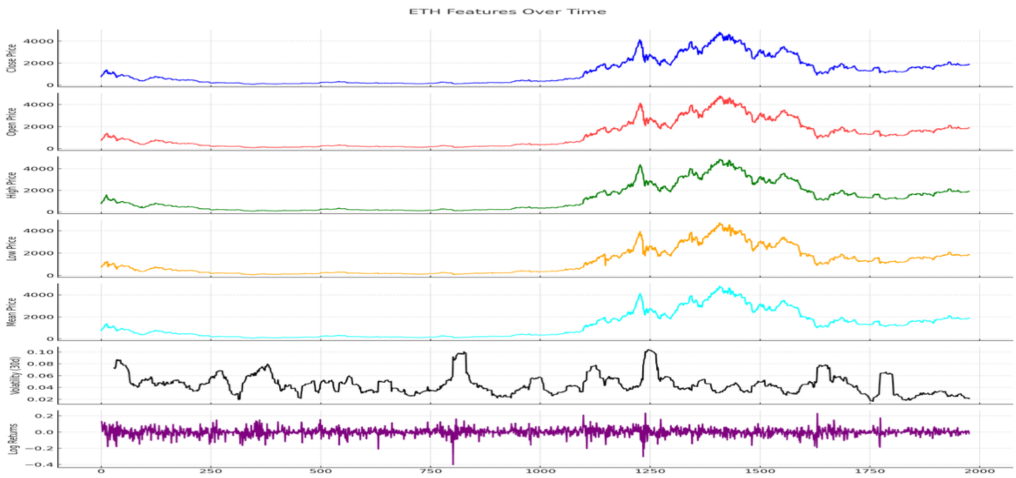


Figure. ETH Feature plot from 2018 to 2023

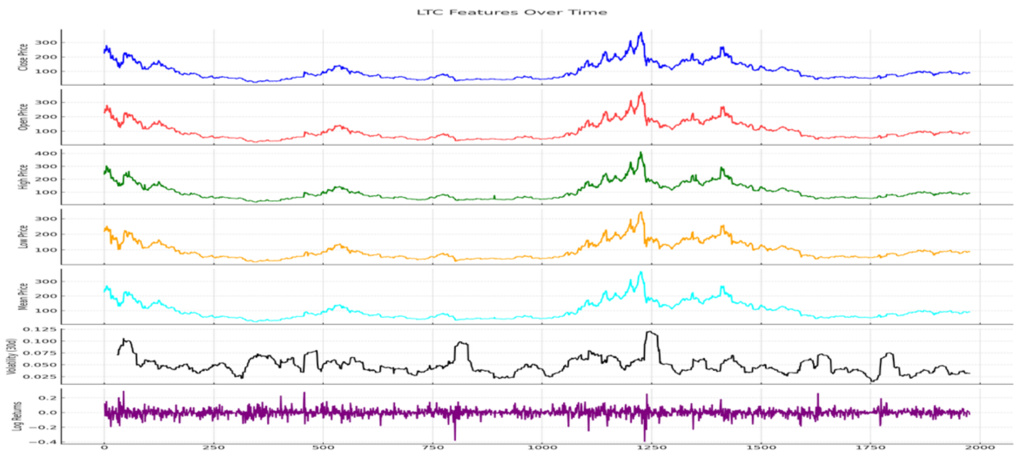


Figure. LTC Feature plot from 2018 to 2023

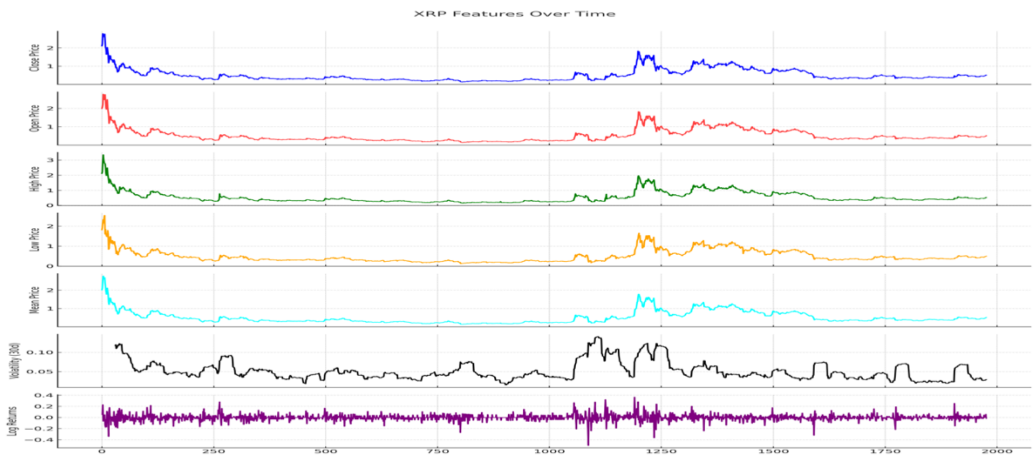
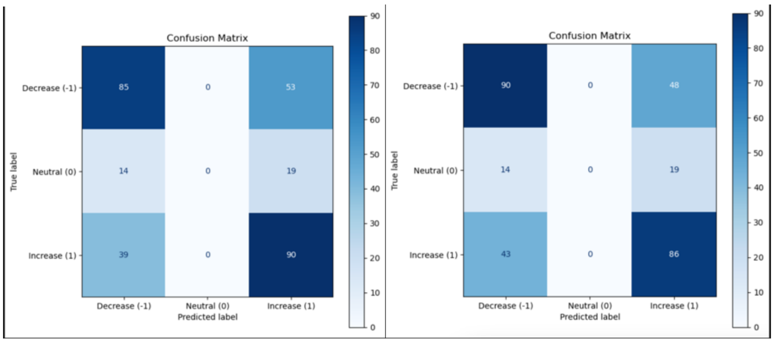


Figure. XRP - Feature plot from 2018 to 2023

2. LSTM

Appendices




```
F1 Score (macro): 0.41836590729905804
              precision    recall  f1-score   support

          -1     0.6118    0.6739    0.6414       138
           0     0.0000    0.0000    0.0000        33
           1     0.5743    0.6589    0.6137       129

    accuracy                         0.5933       300
   macro avg     0.3954    0.4443    0.4184       300
weighted avg     0.5284    0.5933    0.5589       300
```

Figure. Confusion matrix for original dataset and LSTM classification model & F1-score metrics

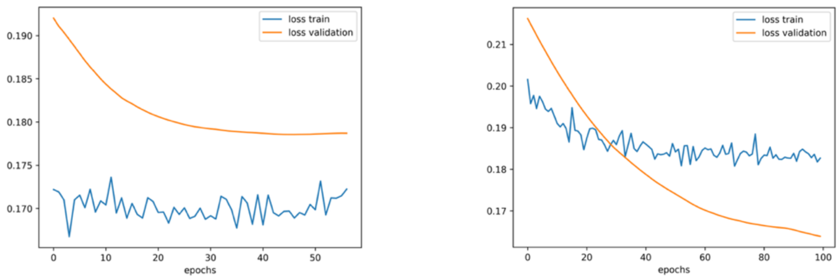


Figure. Best Training and validation performance for Garcia-Medina model using Cryptocurrency Kaggle dataset

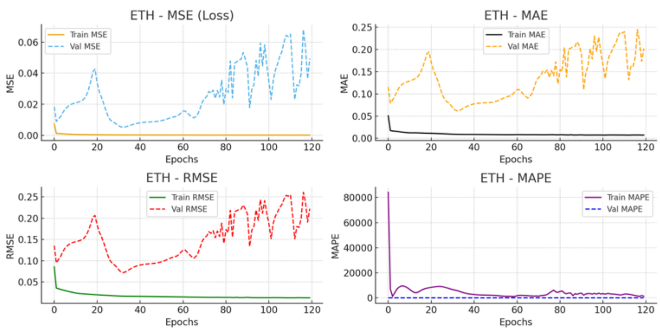


Figure. ETH – Training and validation curves

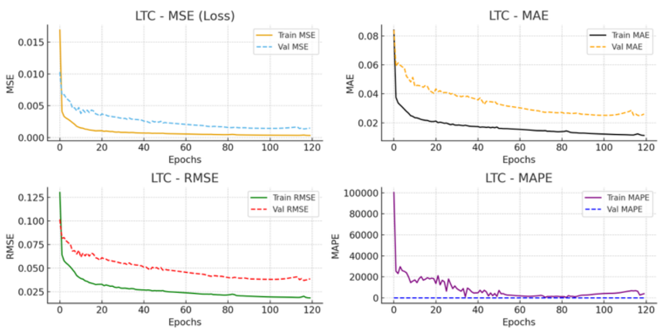


Figure. LTC - Training and validation curves

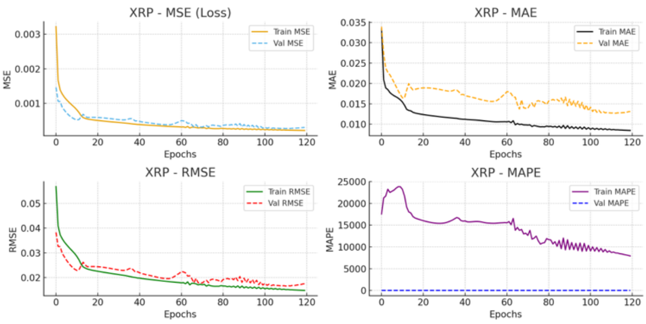


Figure. XRP - Training and validation curves

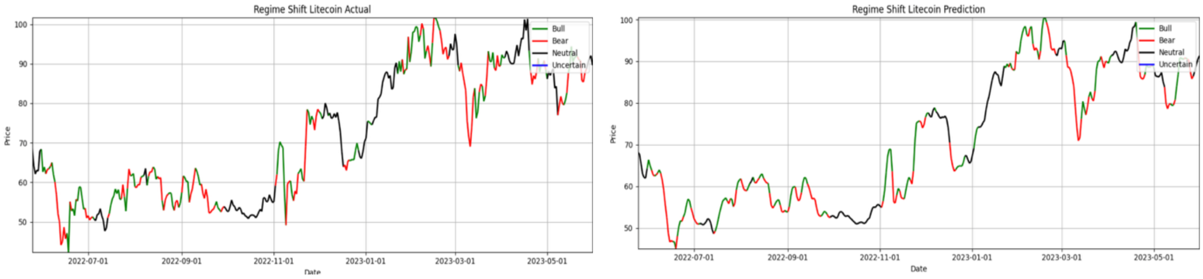


Figure. Litecoin – Regime Shift Classification Actual (left) and Prediction (right) ; seed:45

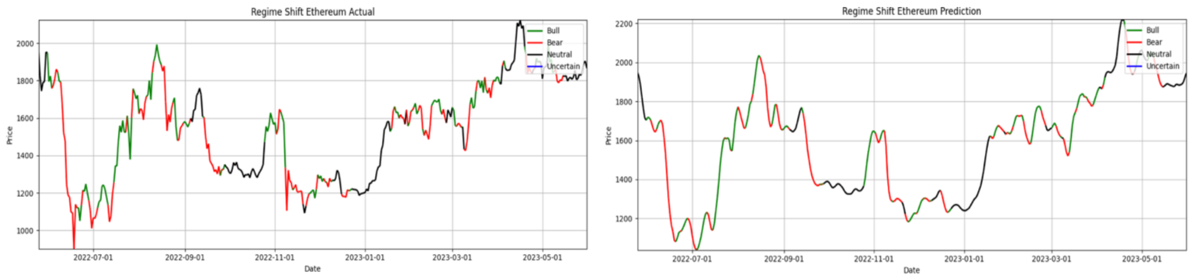


Figure. Ethereum – Regime Shift Classification Actual (left) and Prediction (right) ; seed:45

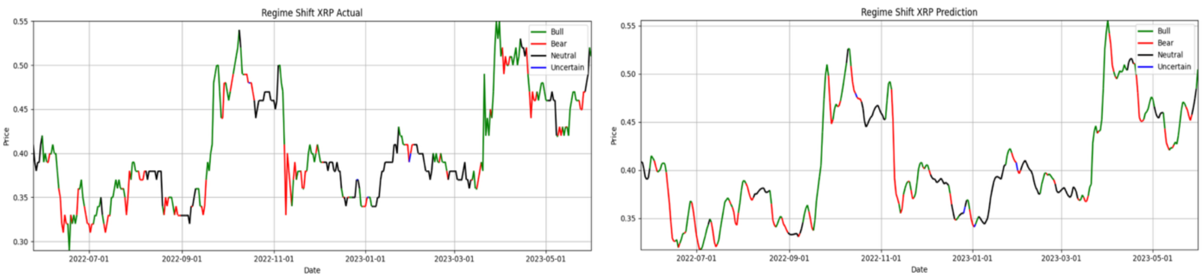


Figure. XRP – Regime Shift Classification Actual (left) and Prediction (right) ; seed:91

3. **CNN-LSTM**

Appendices

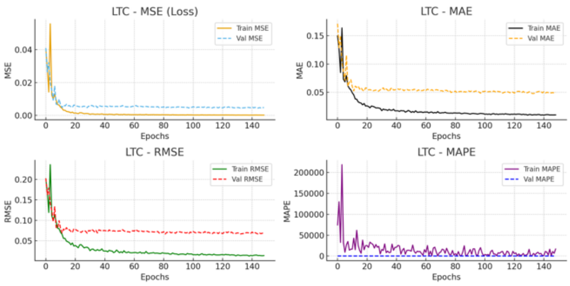


Figure. LTC – Training and validation curves

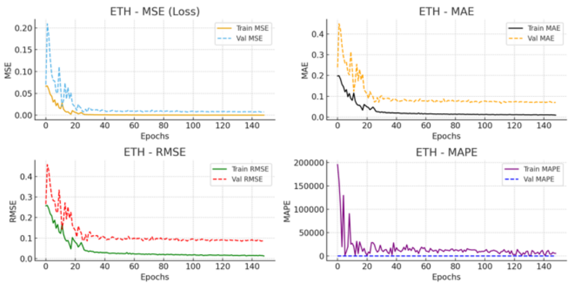


Figure. ETH – Training and validation curves

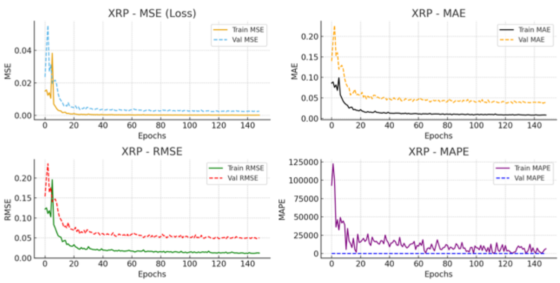


Figure. XRP – Training and validation curves

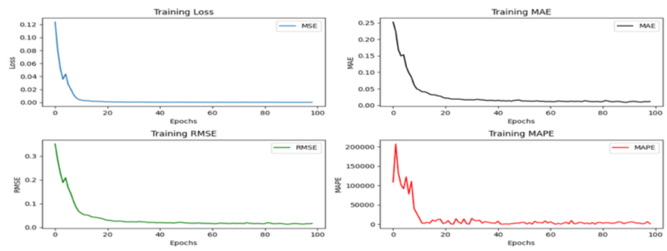


Figure. Original CNN-LSTM training performance

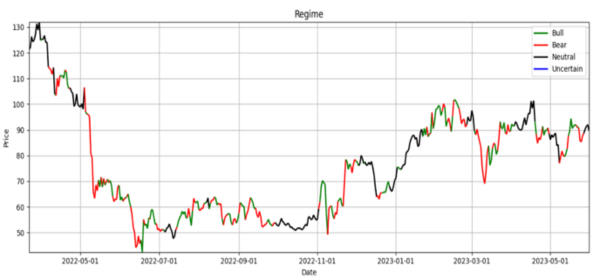


Figure. Litecoin – Regime Shift Classification Actual (left) and Prediction (right) ; seed:91

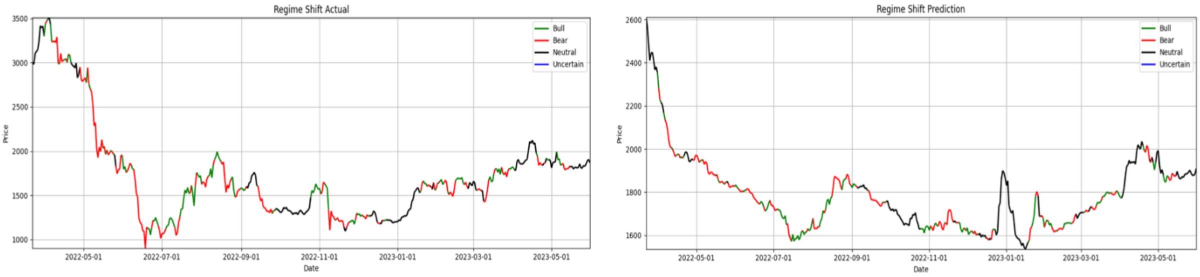


Figure. Ethereum – Regime Shift Classification Actual (left) and Prediction (right) ; seed:45

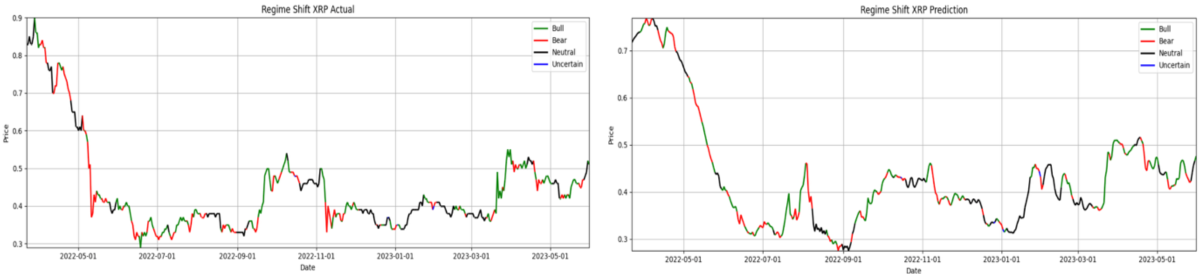


Figure. XRP – Regime Shift Classification Actual (left) and Prediction (right) ; seed:45

4. LSTM-HMM

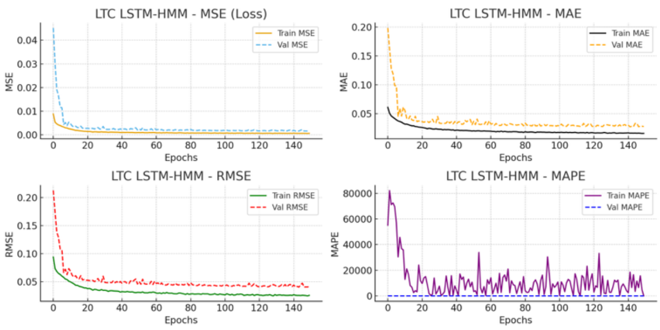


Figure. LTC – Training and validation curves

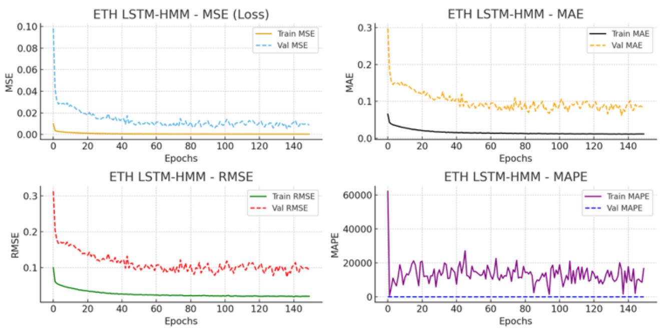


Figure. ETH – Training and validation curves

Appendices

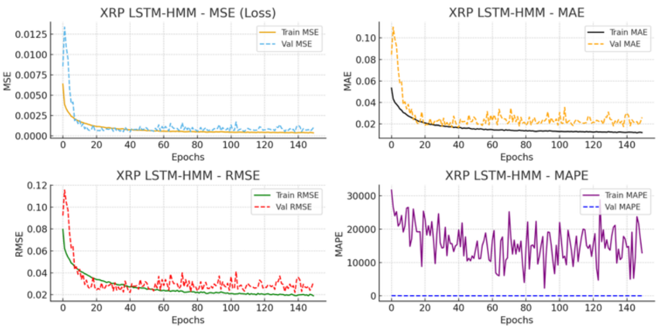


Figure. XRP – Training and validation curves

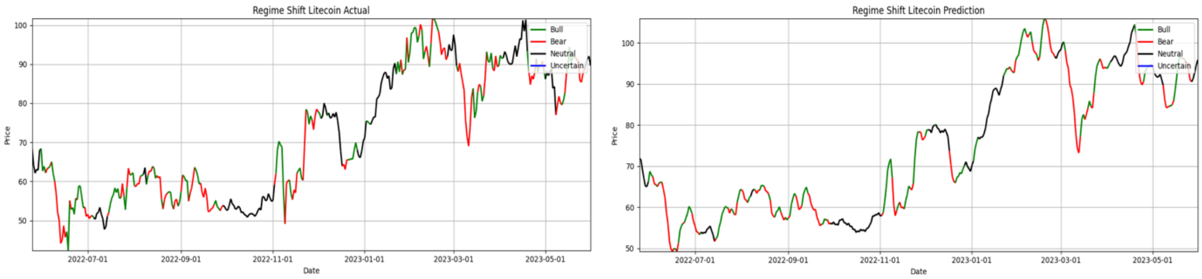


Figure. Litecoin – Regime Shift Classification Actual (left) and Prediction (right) ; seed:45

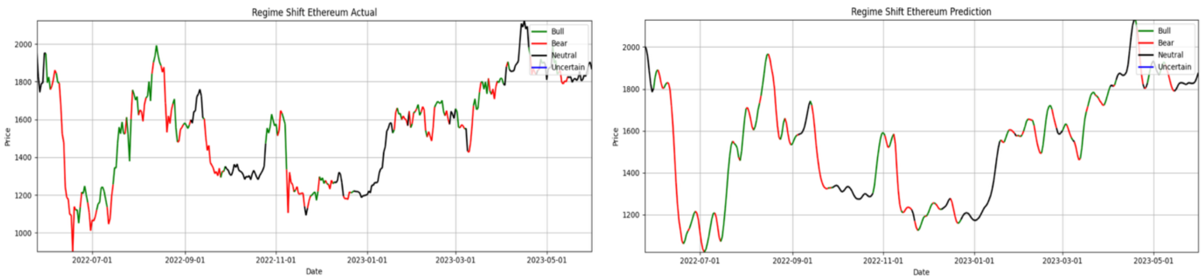


Figure. Ethereum – Regime Shift Classification Actual (left) and Prediction (right) ; seed:91

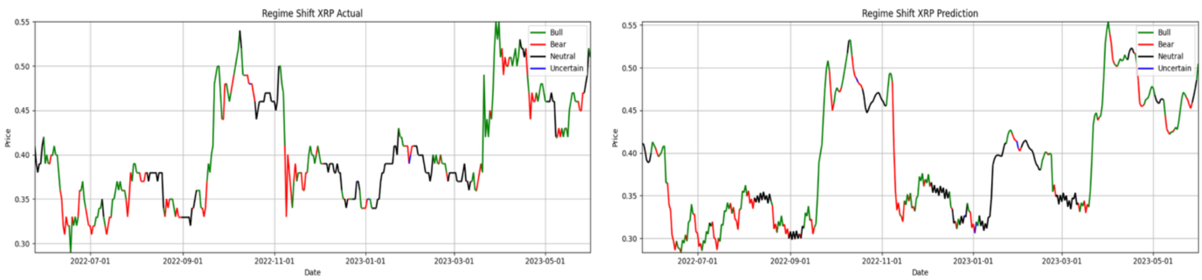


Figure. XRP – Regime Shift Classification Actual (left) and Prediction (right) ; seed:42

Appendices

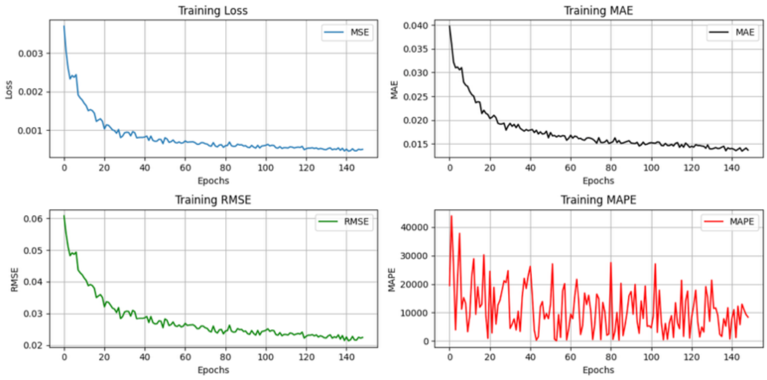


Figure. Original LSTM-HMM – Training performance curves

Appendix C – Use of Generative AI

The Gen AI tool used is ChatGPT. I have included below the specific cases in which it was used, the prompts responses and if any outputs were included in the above report.

- **Download Code as text file to include in the report**

Prompt: "How can you download a notebook code as text file from google colab?"

Response: "Yes. Easiest ways:

Built-in export (whole notebook as code)

Colab menu → File → Download → Download .py.

(You'll get a plain text Python script; rename to .txt if you want.)"

Action: I downloaded the codes as .py python files and renamed them as .txt to included in the appendices in a readable format as required by the guidelines.

- **Produce GADF and GASF matrices for the train and test set (section 4.3.)**

Prompt: "Can you produce two GASF and GADF images for each train and test sets (75%/25% split) represented in different images?"
Response: "All set! I generated GASF and GADF images for BTC using a 75%/25% chronological split:

- BTC Train GASF
- BTC Train GADF
- BTC Test GASF
- BTC Test GADF

If you'd like a fixed image size like 224×224 via resampling, say the word and I'll produce it too."

Action: I downloaded the 4 images which were included in section 4.3.1. GAF/MTF.

- **Produce MTF matrices for full set, train and test set**

Prompt: "Can you similarly produce three MTF matrices one for the full dataset (bitcoin) another for train and one for test (75%/25% split?"
Response: "Done! I generated MTF images for BTC using a 75%/25% chronological split and one full dataset:

- BTC Train MTF
- BTC Test MTF
- BTC Full MTF

If you'd like a fixed image size like 224×224 via resampling, say the word and I'll produce it too"
Action: I downloaded the 3 images and included them in section 4.3.1. GAF/MTF as visualisation of MTF.