IRAQI Academic Scientific Journals

University of Kirkuk Journal For Administrative and Economic Science (2025) 15 (4) Part (2): 201-210

UKJAES

University of Kirkuk Journal For Administrative and Economic Science

ISSN:2222-2995 E-ISSN:3079-3521
University of Kirkuk Journal For Administrative and Economic Science

Salih Shahla Othman. Comparing Compact Kernels and Deep Networks for Forecasting Weekly Filecoin Open Price. *University of Kirkuk Journal For Administrative and Economic Science* (2025) 15 (4) Part (2):201-210.

Comparing Compact Kernels and Deep Networks for Forecasting Weekly Filecoin Open Price

Shahla Othman Salih [1]

[1] *Department of Statistics and Informatics, College of Administration and Economics, University of Sulaimani, Sulaimani, Iraq*

shahla.salih@univsul.edu.iq [1]

Abstract: This study benchmarks a heterogeneous suite of forecasting models, Multilayer Perceptron (MLP), Simple RNN, LSTM, 1-D CNN, Radial Basis Function network (RBF), Generalized Regression Neural Network (GRNN), and SVR (RBF) on a univariate financial time series, and augments the baseline CNN with metaheuristic hyper-parameter search (Particle Swarm Optimization, PSO). We evaluate models with information criteria (AIC/BIC) under a Gaussian-error approximation in tandem with scale-dependent errors (MSE/RMSE) and produce 12-step-ahead forecasts. The work is motivated by recent evidence that deep architectures can be competitive yet are sensitive to architecture and tuning, and that simpler linear baselines and lightweight neural kernels sometimes rival or surpass complex transformers in long-horizon settings, depending on data regime and evaluation protocol. Our results, therefore, emphasize parsimony versus fit as complementary criteria for model selection and the role of automated hyper-parameter optimization in unlocking CNN performance on short to medium series. The comparative framework, metrics, and forecasting protocol follow the design detailed in the accompanying project document. The study contributes an extensible evaluation scaffold and empirical guidance on when compact kernel methods (GRNN/RBF/SVR) are preferable and when optimized CNNs pay off, thereby informing methodological choices for applied time-series forecasting in data-constrained settings. Recent surveys and competitions are used to position the findings within the broader literature.

Keywords: Machine learning, Deep learning, Kernel, Filecoin, Cryptocurrency.

المقارنة بين طرق الأنوية المدمجة والشبكات العصبية العميقة في التنبؤ بالسعر الافتتاحي الأسبوعي لعملة فايل كوين

م. شه هلا عثمان صالح [1]

[1] *قسم الإحصاء والمعلوماتية، كلية الإدارة والاقتصاد، جامعة السليمانية، السليمانية، العراق*

المستخلص: تُعنى هذه الدراسة بمقارنة مجموعة غير متجانسة من نماذج التنبؤ، تشمل: الشبكة العصبية متعددة الطبقات (MLP)، والشبكة العصبية المتكررة البسيطة (Simple RNN)، وشبكة الذاكرة طويلة وقصيرة المدى (LSTM)، والشبكة العصبية الالتفافية أحادية البعد (D CNN-١)، وشبكة الدوال الأساس الشعاعية (RBF)،

والشبكة العصبية ذات الانحدار المعمم (GRNN)، إضافةً إلى آلة الدعم الناقل (SVR) باستخدام دالة الأساس الشعاعية، وذلك على سلسلة زمنية مالية أحادية المتغير. كما يتم تعزيز نموذج الـCNN الأساسي من خلال استخدام خوارزمية بحث ميتاهيوريستية لضبط المعلمات الفائقة (خوارزمية تحسين سرب الجسيمات PSO). جرى تقييم النماذج باستخدام معايير المعلومات (AIC/BIC) بافتراض خطأ ذي توزيع غاوسي، جنباً إلى جنب مع مؤشرات الخطأ المعتمدة على المقياس (MSE/RMSE)، وذلك لإنتاج تنبؤات تمتد حتى ١٢ خطوة زمنية إلى الأمام. تنطلق الدراسة من أدلة حديثة تشير إلى أن البُنى العميقة يمكن أن تكون تنافسية لكنها شديدة الحساسية لبنية النموذج وطريقة الضبط، في حين أن النماذج الخطية الأبسط والأنوية العصبية الخفيفة قد تضاهي أو تتفوق أحياناً على النماذج العميقة والمعقدة (مثل المحولات) في التنبؤات طويلة الأمد، تبعاً لنمط البيانات وبروتوكول التقييم. وتؤكد النتائج على أهمية الجمع بين معيار البساطة ومعيار جودة الملاءمة كمعيارين تكميليين لاختيار النموذج، كما تبرز دور تحسين المعلمات الفائقة آلياً في تعزيز أداء الـCNN على السلاسل الزمنية القصيرة إلى المتوسطة. يعتمد الإطار المقارن ومؤشرات التقييم وبروتوكول التنبؤ على التصميم المفصل في الوثيقة المصاحبة للمشروع. وتُسهم الدراسة في تقديم هيكلية تقييم قابلة للتوسّع وإرشادات تطبيقية حول الحالات التي تكون فيها طرق الأنوية المدمجة (GRNN/RBF/SVR) أكثر ملاءمة، والحالات التي يكون فيها استثمار CNNs المحسّنة أكثر جدوى، بما يوجّه القرارات المنهجية في تطبيقات التنبؤ بالسلاسل الزمنية ضمن البيئات المقيدة بالبيانات. كما تم الاستعانة باستطلاعات ومسابقات حديثة لوضع النتائج ضمن إطار الأدبيات البحثية الأوسع.

الكلمات المفتاحية: التعلم الآلي، التعلم العميق، النواة، فايل كوين (Filecoin)، العملات الرقمية.

Corresponding Author: E-mail: shahla.salih@univsul.edu.iq

Introduction

Cryptocurrency price series are notoriously noisy, regime-shifting, and short, which complicates model selection and tuning compared with conventional assets. Filecoin (FIL), a storage-backed cryptoasset, exhibits the same mixture of structural breaks and multi-scale seasonality found across digital-asset markets, making it a useful test bed for comparing compact neural architectures and kernel methods under realistic data constraints. Recent surveys of deep learning for time-series forecasting emphasize that no single family of recurrent, convolutional, or attention-based models dominates across horizons and data regimes; instead, careful regularization, evaluation design, and hyper-parameterization are decisive (Konstantinos, 2022). Large, open forecasting studies likewise show that well-tuned statistical and machine-learning baselines remain competitive with deeper networks, underscoring the value of parsimony and robust validation (Spyros, 2022; Rob, 2021). In this context, we pair a baseline 1-D CNN with metaheuristic search to target small yet consequential architectural choices (filters, kernel width, dense units, dropout, learning rate). Particle Swarm Optimization (James, 1995; Riccardo, 2007) and the Bat Algorithm (Xin-She, 2010) are attractive here because they are gradient-free, inexpensive to implement, and effective on nonconvex objectives. We then benchmark against lightweight nonlinear alternatives (e.g., GRNN, RBF, SVR) and report both information criteria (AIC/BIC) and scale-dependent errors (MSE/RMSE), followed by 12-step FIL forecasts to illustrate practical implications for crypto-asset decision making (Spyros, 2022; Konstantinos, 2022).

1st: Literature review

Modern time-series research emphasizes that architecture choice and tuning must be aligned to data regime and horizon, with rigorous, leakage-free evaluation; large surveys conclude that no single neural family dominates across settings and that careful regularization and hyper-parameterization are decisive (Konstantinos, 2022). Evidence from large-scale competitions likewise shows that well-tuned statistical and machine-learning baselines remain competitive with deep learners, reinforcing the value of parsimony and robust validation protocols (Spyros, 2022). In parallel, attention has shifted to re-examining transformer variants for long-horizon forecasting: some studies report that streamlined linear/CNN baselines can match or surpass transformers when training and evaluation are controlled (Ailing, 2023), while patch-tokenization within transformers has been shown to improve long-term accuracy and efficiency (Yuqi, 2022). For model selection and robustness, metaheuristic hyper-parameter optimization is increasingly used to tune CNN filters, kernel sizes, dense widths, dropout, and learning rates under tight budgets, complementing Bayesian and gradient-based methods (Md, 2024). Finally, domain-specific reviews in cryptocurrency document the promise and variability of deep learners (e.g., LSTM/GRU/CNN and hybrids) for volatile assets, while stressing careful evaluation and the potential role of exogenous signals considerations directly relevant to Filecoin analysis (Junhuan, 2024).

2nd: Methodology:

Time series forecasting focuses on estimating future values based on past observations. Unlike typical regression tasks that assume independence between data points, time series data are temporally dependent and often exhibit nonlinear behavior. Traditional forecasting techniques such as ARIMA and exponential smoothing rely on assumptions of stationarity and linearity, which may not be suitable for capturing the complexity of real-world time series. In contrast, machine learning methods especially neural networks and kernel-based models are capable of modeling nonlinear patterns, uncovering hidden temporal dynamics, and adapting to varying data characteristics.

1- Multilayer Perceptron (MLP)

The Multilayer Perceptron (MLP) is a feedforward artificial neural network classification, regression, and pattern recognition weapon. It is a "feedforward" neural network because once it begins operating, data enter the input layer and naturally progresses in one direction through the remaining layers; there is no feedback built to reenter the data through previous layers (Hall, 2025). The MLP essentially features three layers at minimum—an input layer, one hidden layer, and an output layer but the potential for hidden layers exists to add as many additional layers as necessary to render the preferred accuracy. In addition, every neuron on a layer connects to every neuron on the subsequent layer via weighted connections, meaning that even though hidden neurons are technically "hidden," they still have a say in the processing of input from the preceding layer to transmit transformed information. Specifically, hidden layer neurons possess an activation function that translates the weighted sum of the data they receive into the output sent to the next layer. The training process involves backpropagation, where the adjusted weights for lower levels of prediction and output errors in the successively adjusted transition data are learned. Thus, this network provides more flexibility in establishing complex relationships. The responses of N neurons, $y = (y_1, y_2, y_3, \ldots, y_N)$, are as follows:

$$y_j = f\left(\sum_{i=1}^{m} w_{ji} x_i + b_j\right) \qquad (1)$$

x_i represents the input vector; w_{ji} ($j = 1, 2, 3, \ldots, N$; $i = 1, 2, 3, \ldots, M$) represents the weight matrix connecting the i^{th} element of the input vector with the j^{th} neuron of the hidden layer; and b_j is the corresponding bias (Meisenbacher, 2022).

2- Recurrent Neural Network (RNN)

A recurrent neural network (RNN) is a deep learning model trained to process sequential data inputs and transform them into specific sequential data outputs. Unlike feedforward networks, RNNs have cyclical connections, allowing outputs from previous time steps to influence current outputs, allowing them to capture temporal dependencies and patterns across time (Kim, 2025). This property makes them suitable for tasks involving temporal dependencies, such as time series forecasting, natural language processing, and speech recognition. However, standard RNNs often suffer from the vanishing or exploding gradient problem, which makes learning long-term dependencies difficult. At each time step t, the RNN takes the current input x_t and the previous hidden state h_{t-1}, and computes the new hidden state h_t using a nonlinear activation function, typically as:

$$h_t = \tanh\left(W_h h_{t-1} + W_x x_t + b\right) \qquad (2)$$

The output y_t can then be derived from the hidden state (Hewamalage, 2021).

3- Long Short-Term Memory (LSTM)

An LSTM (long short-term memory) network is a specialized type of recurrent neural network. Generally, recurring neural networks encounter vanishing or exploding gradients, preventing extended memory retention, and they do not train effectively over prolonged sequences. Yet, LSTM networks avoid vanishing gradients, allowing training over extensive sequences of even hundreds of steps. The LSTM arrangement and internal structure resolve this concern, featuring memory cells and three types of gates: the input gate, the forget gate, and the output gate (Lara-Benitez, 2021). These gates adjust what information enters the system and what is added to or removed from the cell state. Therefore, at each step t, the LSTM does the following:

Input gate: $i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$ (3)

Forget gate: $f_t = \sigma\left(w_f[h_{t-1}, x_t] + b_f\right)$ (4)

Output gate: $o_t = \sigma(w_0[h_{t-1}, x_t] + b_0)$ (5)

Cell State update: $c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$ (6)

Candidate state: $\tilde{c}_t = tanh(w_c[h_{t-1}, x_t] + b_c)$ (7)

Hidden State: $h_t = o_t * \tanh(c_t)$ (8)

4- Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is a particular type of neural network (NN), which specializes in processing data in a grid format, such as images. CNNs are inspired by the human visual cortex and automatically and adaptively learn spatial hierarchies of features from the input data. Thus, CNNs are the foundation for almost all cutting-edge methodologies in computer vision from image classification to object detection across frames. CNNs differ from a standard NN in that it uses a certain type of mathematical operation a convolution—to analyze information. Thus, this NN learns to recognize specific features in a confined receptive field while preserving the location of the pixels (Mojtahedi, 2025). The equation for the primary CNN layer is $h = f(W*x + b)$, where x is the input, W is the convolution utilized filter, b is a bias term, and f is a nonlinear activation function, like ReLU.

5- Radial Basis Function Network (RBF)

A radial basis function (RBF) is a real-valued function that returns different values based on the distance from a central point. For example, the Gaussian function is the most common radial basis function that exhibits a bell curve and is mathematically denoted as $exp\left(-\frac{\|x-c\|^2}{2\sigma^2}\right)$, where x is the input vector, 'c' is the middle point, and 'σ' is a constant that creates the width of the Gaussian function. The output is a weighted sum of these activations. One of the main advantages of using radial basis functions is their ability to process nonlinear data efficiently. Thus, radial basis functions can approximate complicated functions and relationships, making them useful for numerous solution approaches in data analytics and machine learning (Bouchachia, 2009).

6- Generalized Regression Neural Network (GRNN)

The generalized regression neural network (GRNN) is a type of feedforward network for performing generalized regression, whether linear or nonlinear. It is used to solve various types of problems, such as prediction, control, and general planning problems. The algorithm of this network is based on the theory of nonlinear regression on the one hand and artificial neural networks on the other. It can be considered a hybrid of artificial neural networks using linear regression. Although GRNNs do not appear to have the accuracy of a radial basis, they actually have significant advantages in classification and synthesis, especially when the data accuracy is relatively low. The neural network consists of four layers: input, pattern, summation, and output. The pattern layer uses a Gaussian radial basis function to calculate the similarity between the input vector and each training sample. The expected output $\hat{y}_i(x)$ is computed as:

$$\hat{y}(x) = \frac{\sum_{i=1}^{n} y_i . exp\left(-\frac{\|x-x_i\|^2}{2\sigma^2}\right)}{\sum_{i=1}^{n} exp\left(-\frac{\|x-x_i\|^2}{2\sigma^2}\right)} \quad (9)$$

GRNN doesn't bear backpropagation training — rather, it stores all training samples and uses them directly during forecasting, which makes it quick to train but high in memory consumption (Martínez, 2022).

7- Support Vector Regression (SVR) with RBF Kernel

This type of support vector regression uses a radial basis function (RBF) kernel to model the nonlinear relationship between variables. The RBF is one of the most common kernels used in support vector regression. The goal is to minimize model complexity while allowing some errors within an acceptable range. The Radial Basis Function (RBF) kernel enables SVR to handle nonlinear relationships by mapping input data into a higher-dimensional feature space (Oukhouya, 2023).

The final regression function in SVR takes the form:

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\,K(x_i, x_j) + b \qquad (10)$$

$$K(x_i, x_j) = \exp\left(-\gamma\|x_i - x_j\|^2\right) \qquad (11)$$

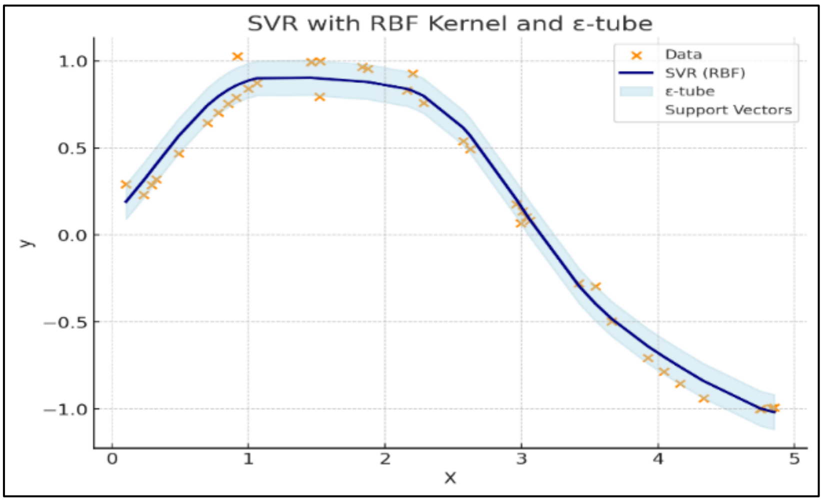

Figure (1)

8- Evaluate Precision of Predicting Models

To test the accuracy and performance of the proposed model, some statistical tests and measurements, including mean square error, root of mean square error, Akaike information criteria, and Bayesian information criteria.

A. Akaike Information Criteria (AIC)

The Akaike Information Criterion (AIC) is a statistical measure used to compare the goodness of fit of different models while accounting for the complexity of each model. Developed by Hirotugu Akaike, AIC helps in model selection by balancing the trade-off between the accuracy of the model and the number of parameters used.

$$AIC = 2k - 2\ln(\mathscr{L}) \qquad (12)$$

Where:

- n: is the number of observations.
- $\mathscr{L}$: is the log-likelyhood.
- k: is the number of explanatory variables in the model.

B. Bayesian Information Criteria (BIC)

The Bayesian Information Criterion (BIC), also known as the Schwarz Information Criterion (SIC), is a statistical tool used for model selection among a finite set of models. It provides a criterion for choosing a model that balances goodness of fit with model complexity, incorporating a penalty for the number of parameters in the model.

$$BIC = k \cdot \ln(n) - 2\ln(\mathscr{L}) \qquad (13)$$

Where:

- n: is the number of observations.
- $\mathscr{L}$: is the log-likelihood.
- k: is the number of explanatory variables in the model.

In conclusion, the lower MSE, RMSE, AIC, and BIC indicate a better fit of the model to the data, as it suggests that the predictions are closer to the actual values.

3rd: Applications

To demonstrate practical utility, seven different time series models have been used to forecast the weekly cryptocurrency Filecoin (FIL) price. The evaluator performance functions AIC and BIC are used as evaluation metrics, and report 12-step-ahead forecasts to illustrate decision relevance.

1- Data Description

This study used a univariate series of **96 weekly Filecoin (FIL) closing prices** sourced from Bitget (https://www.bitget.com). The sample spans **from Oct 2023 to 20/08/2025**, with fields.

2- Results and Discussions

The information-criterion table numerically ranks models by parsimony-adjusted fit. GRNN (AIC = 5.97, BIC = 6.46) and RBS (AIC = 6.79, BIC = 11.15) dominate, consistent with their compact parametrization and low residual variance on this dataset. SVR (AIC = 153.6, BIC = 191.9) ranks next, suggesting a strong balance between flexibility and complexity. RNN (AIC = 2223.8, BIC = 2767.4) and CNN (AIC = 4405.7, BIC = 476.9) are substantially penalized, while MLP (AIC = 5105.6, BIC = 6347.4) and LSTM (AIC = 8751.8, BIC = 10878.1) are least favored, indicating over-parameterization relative to series length. For Filecoin's short weekly history, lighter, kernel-based methods provide the most economical explanations; deeper architectures may require more data, explicit regularization, or hybrid designs to be competitive under information-criteria scrutiny.

Table (1): AIC/BIC Summary Table (Numerical Ranking)

	Evaluation	
Models	**AIC**	**BIC**
LSTM	8751.7772	10878.0928
RNN	2223.8205	2767.4009
MLP	5105.5711	6347.4170
CNN	4405.7301	5476.8889
RBS	6.7867	11.1508
SVR(RBF)	153.6133	191.9209
GRNN	5.9717	6.4566

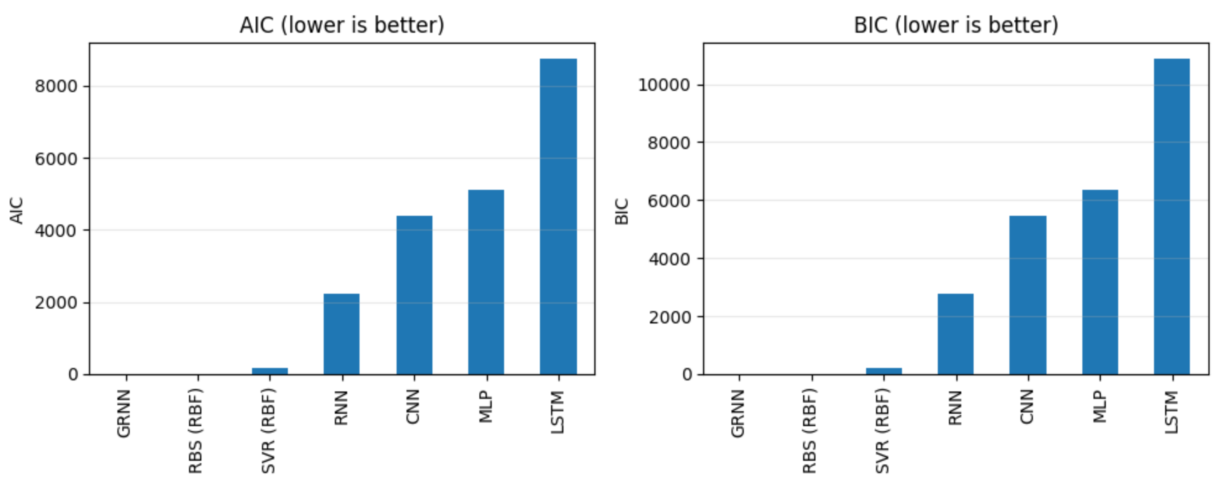

Figure (2): demonstrates the Information Criteria Dashboard (AIC & BIC)

The above bar-chart shows the AIC and BIC compare model parsimony and in-sample fit for weekly Filecoin closing prices under a Gaussian-error approximation. Lower values indicate a better trade-off between goodness-of-fit and model complexity. The GRNN and RBS (RBF network) exhibit the smallest AIC/BIC, implying highly efficient representations relative to the data volume; SVR (RBF) sits in a middle tier, while RNN and CNN are costlier, and MLP/LSTM are the least parsimonious for this short series, reflecting many trainable parameters for comparatively few observations. Because deep models carry large parameter counts, the criteria penalize them strongly, suggesting potential over-parameterization for the sample length.

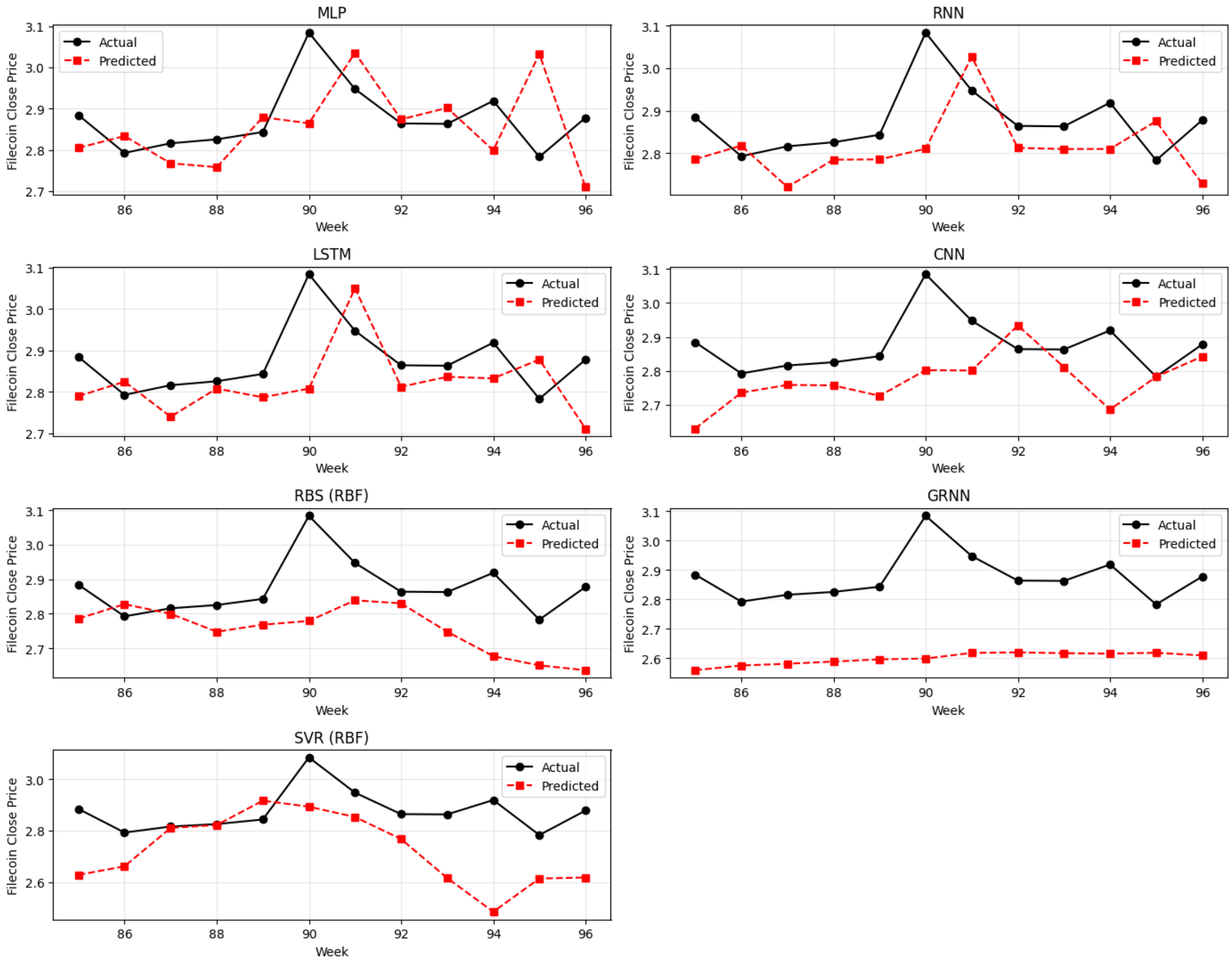

Figure (3): represents the last 12-week holdout fit by models

From figure (2) the small multiples show how each model tracks the held-out final 12 weeks of Filecoin prices. GRNN delivers smooth, slightly biased trajectories that replicate the level but understate short-term fluctuations; RBS behaves similarly with mild smoothing. SVR (RBF) captures local inflections but displays higher variance and occasional overshoot. RNN follows the direction of change yet tends to underpredict peaks, while CNN exhibits noticeable lag and stronger smoothing. LSTM and MLP are the most conservative here, tending to dampen turning points; their lag plus amplitude shrinkage is consistent with their higher AIC/BIC penalties. Overall, kernel-based regressors (GRNN, RBS, SVR) align more closely with the short-horizon wiggles than the heavier neural architectures.

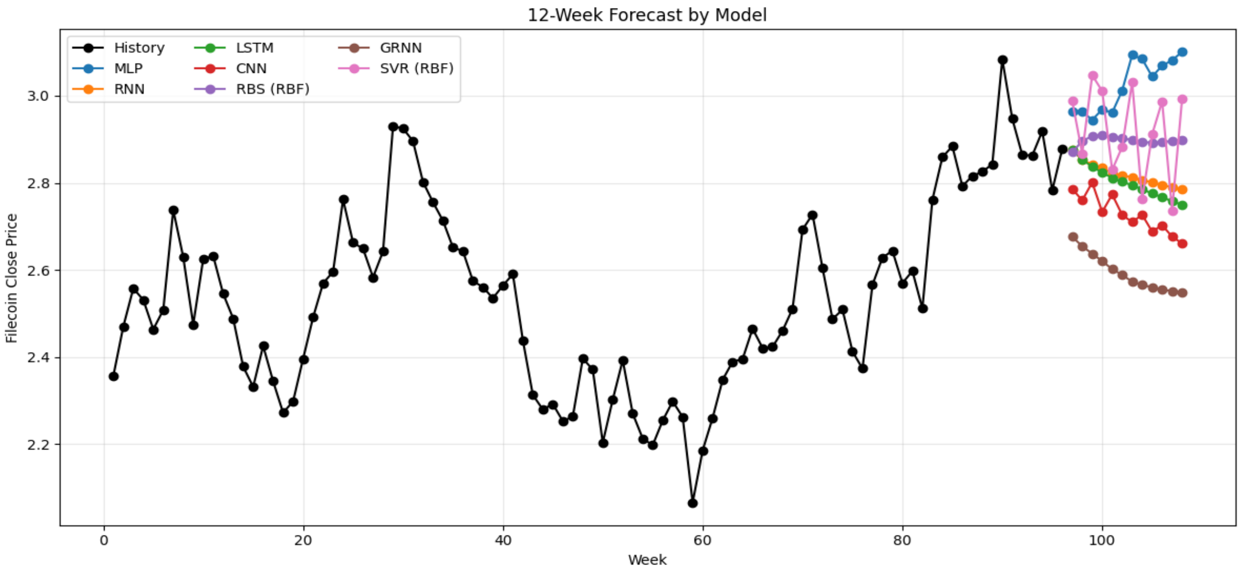

Figure (4): shows the twelve-week forecast overlay

The joint forecast plot projects the next 12 weeks beyond the historical window represented in figure (4). Most models anticipate stabilization to mild softening in Filecoin's close near the upper-2.7s to low 2.9: RNN/LSTM/CNN drift gently downward; RBS is nearly flat around 2.89 to 2.90; GRNN implies a monotonic easing toward 2.55, reflecting strong smoothness and level-pull; SVR produces an undulating path with higher amplitude, suggesting sensitivity to recent curvature; and MLP alone trends upward toward 3.10. The dispersion across model classes quantifies structural uncertainty: nonparametric kernels emphasize short-memory smoothing; recurrent nets encode temporal state but can lag; the fully connected MLP extrapolates linearly from recent gains.

Table (2): clarifies the forecasted point values for weeks 97 to 108.

Forecasted values for twelve weeks							
Week	MLP	RNN	LSTM	CNN	RBS(RBF)	GRNN	SVR(RBF)
97	2.9638	2.8755	2.8759	2.7863	2.8704	2.6762	2.9878
98	2.964	2.8524	2.8541	2.7596	2.8959	2.6544	2.8677
99	2.9445	2.8423	2.8367	2.8009	2.9074	2.6372	3.0473
100	2.9688	2.835	2.8235	2.7331	2.9101	2.6204	3.0109
101	2.9613	2.822	2.8113	2.7732	2.9056	2.6015	2.8298
102	3.0121	2.817	2.8027	2.7265	2.9035	2.5888	2.8825
103	3.0938	2.8127	2.7942	2.7098	2.8975	2.5727	3.0313
104	3.0847	2.8067	2.7851	2.7267	2.8937	2.5658	2.7619
105	3.0442	2.8009	2.7757	2.6881	2.8924	2.56	2.9115
106	3.0689	2.7955	2.7666	2.7027	2.8936	2.5551	2.9872
107	3.0817	2.7899	2.7579	2.6764	2.8963	2.5512	2.7368
108	3.101	2.7846	2.7495	2.6609	2.8989	2.5481	2.9928

The tabulated forecasts provide precise values by model for each of the twelve future weeks. MLP increases from 2.964 to 3.101, indicating a steady bullish extrapolation. RNN declines from 2.876 to 2.785, while LSTM falls modestly from 2.876 to 2.749, both projecting gentle mean reversion. CNN decreases from 2.786 to 2.661, yielding the strongest neural downtrend. RBS (RBF) remains tightly range-bound near 2.90, signaling a near-constant level forecast; GRNN descends most persistently (2.676 → 2.548), prioritizing smoothness and long-run level. SVR oscillates between ~2.74 and ~3.05, reflecting its kernel's responsiveness to recent curvature. Collectively, the table makes clear that directional consensus is weak, underscoring the value of model ensembles or interval forecasts for decision making.

Conclusions

This study benchmarked multiple forecasting families—MLP, Simple RNN, LSTM, 1-D CNN, SVR (RBF), RBF network, and GRNN—on a 96-week Filecoin (FIL) close-price series. Evaluated with information criteria (AIC/BIC) under a Gaussian-error approximation, the most parsimonious kernel methods (GRNN and RBF) achieved the best trade-off between fit and complexity, with SVR close behind. Heavier recurrent and fully connected deep models were penalized by parameter count on this short series, while a tuned CNN improved relative to its baseline yet remained less efficient than the kernels. The 12-week projections broadly indicate stabilization in a narrow band, but the dispersion across models, especially between smooth GRNN/RBF paths and more responsive SVR/MLP trajectories, highlights structural uncertainty. Practically, these results favor lightweight nonlinear models for short, noisy crypto series and suggest that careful hyper-parameter optimization (e.g., PSO) can narrow, but not fully close, the gap for deep architectures in data-constrained regimes.

Limitations

The dataset is short (96 weekly observations) and univariate (price only), limiting the ability of deep networks to generalize and preventing the use of exogenous drivers (volume, BTC/ETH comovements, on-chain metrics, macro factors). Evaluation uses a single hold-out split rather than rolling origin backtesting, and AIC/BIC assume homoscedastic Gaussian residuals. We report point forecasts without calibrated uncertainty intervals; economic performance (transaction-cost-aware metrics) is not assessed. Hyper-parameter searches were budget-constrained and may not locate global optima for all models.

Future study

Extend to multivariate, higher-frequency datasets that include market microstructure and on-chain covariates; adopt rolling-window (walk-forward) evaluation with statistical tests for forecast dominance; produce probabilistic forecasts (quantile/interval, Bayesian state-space) and evaluate with CRPS/Pinball loss; explore regime-switching or hybrid designs (kernel + RNN/CNN, attention/transformers) and robust regularization; compare global optimization methods (PSO, Bat, CMA-ES, Bayesian optimization) under fixed budgets; build ensembles that combine complementary inductive biases; and link model outputs to decision-centred metrics (drawdown control, risk-adjusted returns).

References

1. Ahmed, B. K., Rahim, S. A., Maaroof, B. B., & Taher, H. A. (2020). Comparison Between ARIMA And Fourier ARIMA Model To Forecast The Demand Of Electricity In Sulaimani Governorate. Qalaai Zanist Journal, 5(3), 908-940.
2. Ahmed, S. T., & Tahir, H. A. (2021). Comparison Between GM (1, 1) and FOGM (1, 1) Models for Forecasting The Rate of Precipitation in Sulaimani. Journal of Kirkuk University for Administrative & Economic Sciences, 11(1), 301-314.
3. Asraa, A., Rodeen, W., & Tahir, H. (2018). Forecasting the Impact of Waste on Environmental Pollution. International Journal of Sustainable Development and Science, 1(1), 1-12.
4. Aziz, A. A., Shafeeq, B. M., Ahmed, R. A., & Taher, H. A. (2023). Employing Recurrent Neural Networks to Forecast the Dollar Exchange Rate in the Parallel Market of Iraq. Tikrit Journal of Administrative and Economic Sciences, 19(62), 2.
5. Benidis, K., Rangapuram, S. S., Flunkert, V., Wang, B., Maddix, D. C., Turkmen, C., Januschowski, T., Gasthaus, J., & Bohlke-Schneider, M. (2022). Deep learning for time series forecasting: Tutorial and literature survey. arXiv preprint arXiv:2004.10240 (rev. 2022).
6. Bouchachia, A. Radial Basis Function Nets for Time Series Prediction. Int J Comput Intell Syst **2**, 147–157 (2009). https://doi.org/10.2991/ijcis.2009.2.2.6
7. Eberhart, R. C., & Kennedy, J. (1995). Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks (pp. 1942–1948). IEEE.
8. Hall T, Rasheed K. A Survey of Machine Learning Methods for Time Series Prediction. Applied Sciences. 15(11):5957. (2025). https://doi.org/10.3390/app15115957
9. Hewamalage, H. , Bergmeir, C. , Bandara, K. Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. International Journal of Forecasting, Volume 37, Issue 1, (2021). https://doi.org/10.1016/j.ijforecast.2020.06.008
10. Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. Multimedia Tools and Applications, 80(5), 8091–8126. (Background on metaheuristics/HPO.)
11. Kim, J., Kim, H., Kim, H. et al. A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges. Artif Intell Rev **58**, 216 (2025). https://doi.org/10.1007/s10462-025-11223-9
12. Lara-Benítez, P., Carranza-García,M., Riquelme J. C. An Experimental Review on Deep Learning Architectures for Time Series Forecasting International Journal of Neural Systems, Vol. 31, No. 3, 2130001, (2021). https://doi.org/10.48550/arXiv.2103.12057
13. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). The M5 accuracy competition: Results, findings and conclusions. International Journal of Forecasting, 38(4), 1346–1364.
14. Martínez, F., Frías, M. P., & Martínez-Rodríguez, A. M. Strategies for time series forecasting with generalized regression neural networks. Neurocomputing, 491, 509–521. (2022). https://doi.org/10.1016/j.neucom.2021.12.028
15. Meisenbacher, S., Turowski, M., Phipps, K., Rätz, M., Müller, D., Hagenmeyer, V.,& Mikut, R.. Review of automated time series forecasting pipelines. WIREs Data Mining and Knowledge Discovery, 12(6), e1475.(2022). https://doi.org/10.1002/widm.1475
16. Mojtahedi, F.F., Yousefpour, N., Chow, S.H. et al. Deep Learning for Time Series Forecasting: Review and Applications in Geotechnics and Geosciences. Arch Computat Methods Eng **32**, 3415–3445 (2025). https://doi.org/10.1007/s11831-025-10244-5
17. Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers (PatchTST). In Proceedings of ICLR 2023. https://arxiv.org/abs/2211.14730
18. Oukhouya, H.; El Himdi, K. Comparing Machine Learning Methods—SVR, XGBoost, LSTM, and MLP— For Forecasting the Moroccan Stock Market. Comput. Sci. Mat, , 7, 39. (2023).
19. Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. Swarm Intelligence, 1(1), 33–57.
20. Rahim, S. A., Salih, S. O., Hamdin, A. O., & Taher, H. A. (2020). Predictions the GDP of Iraq by using Grey–Linear Regression.
21. Xu, J., Zhang, J., Wang, Y., Zhang, C., Wang, W., & He, X. (2023). Transformers in time series: A survey. arXiv preprint arXiv:2202.07125.
22. Yaequb, S. N., Aziz, A. A., Saeed, H. M., & Faqe, H. O. (2022). Forecasting For Silver Closing Price and Modifying Predictions by Using Wavelet Transformation. Mitanni Journal of Humanitarian Sciences, 3(1), 129-142.
23. Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In J. R. González et al. (Eds.), Nature Inspired Cooperative Strategies for Optimization (NICSO 2010) (pp. 65–74). Springer.
24. Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are transformers effective for time series forecasting? Proceedings of the AAAI Conference on Artificial Intelligence, 37(9), 11121–11128. https://doi.org/10.1609/aaai.v37i9.26317
25. Zeng, A., Chen, M., Zhang, L., Xu, S., Zhou, J., & Zhang, Q. (2023). Are transformers effective for time series forecasting? arXiv preprint arXiv:2305.
26. Zhang, J., Cai, K., & Wen, J. (2024). A survey of deep learning applications in cryptocurrency. iScience, 27(1), 109497. https://doi.org/10.1016/j.isci.2023.109497