

Meta-learning Approach for Time Series Forecasting: First-order MAML and Reptile

Pratik Zinjad*

Department of Computer Engineering, Ramrao Adik Institute of Technology, Navi Mumbai, 400706, India

E-mail: pratikkzinjad@gmail.com

ORCID iD: <https://orcid.org/0009-0004-4343-5116>

*Corresponding author

Tushar Ghorpade

Department of Computer Engineering, Ramrao Adik Institute of Technology, Navi Mumbai, 400706, India

E-mail: tushar.ghorpade@rait.ac.in

ORCID iD: <https://orcid.org/0000-0003-1652-7264>

Vanita Mane

Department of Computer Engineering, Ramrao Adik Institute of Technology, Navi Mumbai, 400706, India

E-mail: vanita.mane@rait.ac.in

ORCID iD: <https://orcid.org/0000-0002-4303-4089>

Received: 16 October 2025; Revised: 03 December 2025; Accepted: 17 January 2026; Published: 08 April 2026

Abstract: Forecasting time series data especially in volatile sectors like financial markets, shows significant challenges due to non-linearity, non-stationarity and noise in the data. Traditional forecasting models most likely fail to generalize effectively across varying tasks without extensive retraining. This study investigates the application of meta learning techniques, particularly First-Order Model-Agnostic Meta-Learning (FOMAML) and Reptile, to make adaptability and generalization better in time series forecasting tasks. An extensive empirical study was done using three neural networks as base models, namely Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU) and Feed Forward Neural Network (FFNN) applied to four real-world stocks: TCS, TATASTEEL, GRASIM and DJIAHD. The models were evaluated under few-shot learning (defined here as 211-shot learning using sliding window samples) conditions with varying iteration counts (outer loops or epochs) and their effectiveness was checked using some common standard metrics like RMSE (Root Mean Squared Error), MAE (Mean Absolute Error) and R^2 (Coefficient of Determination). Outcomes have shown that meta-learning approach notably performs much better than traditional models with MAML (First Order) in particular showing quicker task adaptation as well as stable convergence behavior, especially when it used with GRU and LSTM as base models, as validated empirically on the GRASIM dataset where the MAML with LSTM configuration attained around 81.9% reduction in RMSE (dropping the value from 622.94 to 112.60 over the iterations). In all four stocks, reptile shows relatively steady performance. The study validates the potential of meta-learning as a powerful framework for time series forecasting problem in dynamic settings which offers robust algorithmic foundation for numerous future financial modeling applications.

Index Terms: Few-Shot Learning, GRU, LSTM, MAML, Meta-learning, Reptile, Stock Price Prediction, Time Series Forecasting.

1. Introduction

Forecasting time series data, especially in financial areas like stock markets, presents a persistent challenge because of the data's noisy, non-linear, and non-stationary characteristics. Conventional forecasting models most likely need an extensive amount of retraining when faced to new datasets or market conditions, which restricts their scalability and adaptability. This inefficiency emerges as one of the important factors in dynamic environments like equity markets, where swift changes required models that can generalize and adapt more quickly. Recent advancements that are done in meta-learning makes it an effective approach by enabling models to learn how to learn making quick adaptation to new tasks with minimum fine-tuning. Specifically, algorithms like MAML (used here in its First-Order (FOMAML) formulation for better efficiency) and Reptile have ability of efficient knowledge transfer among multiple

time series tasks, greatly minimizing the need for large size labeled datasets and extensive amount of training [1,2]. Inspired by the success of meta-learning in a number of other fields, this paper explores its application in time series forecasting to be specific to build adaptive models which not only maintain high accuracy but also minimize training time and computational cost. The Objectives of this research paper are as follows:

- Explore and implement MAML(First order) and Reptile algorithms for problem of time series prediction.
- Compare the effectiveness of meta-learning approaches with traditional time series forecasting methods like Autoregressive Integrated Moving Average (Arima), Seasonal Autoregressive Integrated Moving Average (Sarima), etc. using visualizations to show the results.
- Evaluate and explore the adaptability and generalization capabilities of MAML and Reptile algorithms across different conditions or tasks.
- Demonstrate the effectiveness of meta-learned models using low-data and few-shot learning scenarios, using real world historical data.
- Analyze the pros and cons of each meta learning approach through empirical results and visualizations while providing insights for future improvement.

The scope of this paper lies in making time series forecasting capabilities better through the application of meta-learning algorithms, specifically MAML(First Order) and Reptile. Traditional models often need large amounts of training when applied to new datasets or unseen tasks, On the other hand MAML and Reptile offer a better alternative by enabling faster adaptation to unseen time series tasks by minimal gradient updates. This flexibility can be particularly important in financial markets, where stock price behavior is non-stationary as well as there is possibility of frequent changes due to macroeconomic and firm-specific events. Doing training on multiple financial time series, this investigation or study aims to construct a meta-learner capable of generalizing across multiple number of tasks, allowing efficient fine-tuning for new or unseen prediction tasks with limited amounts of data. The few-shot learning framework can further help to minimize data reliance and will also make forecasting accuracy better in such scenarios where data is less. This study expands on the foundational work previously done by Finn et al. [3], who introduced MAML as a second order gradient based meta-learning approach, here in this work we adopted the First-Order approximation (FOMAML) suggested by the authors, which ignores second derivatives to minimize computational expense while keeping performance high and Nichol et al. [4], who proposed Reptile which is also one of the computationally efficient first order approach. Utilizing these meta learning approaches we can create scalable and quick learning predicting models that can effectively be implemented using various stock market datasets. Theoretically, this study aligns with the idea of Adaptive Markets Hypothesis (AMH) given by Andrew Lo [5] which views markets as evolutionary systems that features temporary inefficiencies instead of purely random walks. As a result, here we used meta-learning to leverage these changing regimes which enable model to quickly adapt and catch temporary patterns that static models generally miss out. The main contribution of this paper is outlined below:

- We propose a thorough empirical investigation of two meta-learning approaches those are MAML(First Order) and Reptile, in the context of time series prediction.
- We merged these meta-learning approaches with various base models like LSTM networks, GRU and FFNN, to show their adaptability in various financial time series prediction tasks.
- Our experimental work specifically focuses on stock price prediction problem, where we use few-shot learning (211shot learning) settings while training used models using varying iteration counts (10, 20,, till 100), which makes us easy to systematically investigate the factors like convergence speed, stability, and forecasting performance.
- We studied the behavior of each base model (neural network architecture) using with both MAML(First Order) and Reptile frameworks to identify which combinations give us better performance using low data and task variant settings.
- In contrast to traditional forecasting models our suggested strategy focuses on task adaptation and generalization across various temporal tasks which is especially vital in fast paced financial markets where quick learning from small amount data is very much needed.

This paper is structured into five main sections. The Related Work or Literature Survey is presented in Section 2 of the paper which starts with a thorough review of existing systems and methodologies used for the problem of time series forecasting, followed by limitations and research gaps in existing systems. The Proposed System is described in section 3, including an introduction to meta-learning, the architectural framework used, algorithmic implementation using FOMAML and Reptile, and the hardware-software required to carry out this work. It also outlines the evaluation metrics used to explain the performance of the investigation done. The experiment and results are explained in detail in Section 4, where the proposed approach is assessed using real stock market stocks i.e TCS, Tata Steel, Grasim and DJIAHD, along with a comparative analysis against various traditional models like Arima, Sarima, etc. The investigation's findings are summarized in section 5, which also discusses possible directions for future research.

2. Related Works

2.1. Survey of Existing System

Time series forecasting plays a crucial role in various real-world applications which includes financial market prediction, energy consumption estimation, weather forecasting, and healthcare analytics. Traditional models like exponential smoothing and ARIMA most probably suffer with the problem of non-linearity and noise when working with modern datasets. On the other hand, while using deep learning-based architectures like LSTM [6], GRU and FFNN after investigating the findings have shown strong capabilities in catching temporal patterns and long-term dependencies. But these models want huge amounts of data as well as training when learning needs to get transfer to new tasks, which will cause an inefficiencies in dynamic or low-resource situations. To fix these cons, recent research has been done towards meta learning, a concept that make models to quickly adapt to new tasks with small amount of data. Meta learning approaches, particularly MAML [3,7] and Reptile [4], have shown promising results in the domains like computer vision and reinforcement learning. These algorithms aim to learn a good initialization of model parameters such that a few gradient steps with a small amount of task-specific data can yield optimal performance. Despite their success to learn general tasks easily, their application to time series forecasting remains an emerging area of research. Some efforts have been made and tried to integrate meta-learning with time series forecasting. Chang et al. [8] used a meta-learning framework for short term stock price-trend prediction, leveraging the rapid adaptation capability of meta-learners successfully. Song et al. [9] given a weighted meta extreme learning model for stock prediction, implementing and showing that meta learning can improve generalization across various financial time series tasks. Similarly, Maya et al. [10] used a meta-transfer learning model to handle missing data values in time series, to show improved adaptability across varying temporal patterns.

A novel direction was introduced by van Zyl [11], who introduced late fusion using representation based learning and meta-learning for forecasting, suggesting that learning useful data representations before meta-training can make performance better across various tasks. Another magnificent work done by Xiao et al. [12], where a multi stage loss optimization strategy was developed to make gradient-based meta-learning approaches better, thereby enhancing the convergence behavior of models like MAML and Reptile. Furthermore, comprehensive surveys done by Hospedales et al. [13] and Barman et al. [14] have worked on meta-learning architectures, categorizing optimization-based algorithms (like MAML, Reptile), model-based learners, and metric-based approaches. These surveys give foundational understanding and explain emerging challenges in applying meta-learning techniques to sequential data such as time series. Several hybrid models have also emerged that indirectly relate to the concept of meta-learning. Samanta et al. [1] elaborated a Meta-cognitive Recurrent Fuzzy Inference System (McRFIS-MN) with memory neurons, focusing on quick learning with online adaptation in temporal domains. Li et al. [2] proposed a hybrid forecasting model that combines LSTM with LMS filtering and ISSA-based optimization algorithm. Although it is not a classical meta-learning technique, these models have similar goals of rapid adaptation and generalization. Parallel advances have been made in representation based meta-learning techniques. For example, Gutowska et al. [15] explored constructing meta-learning model for unsupervised anomaly detection while providing insight into learning task specific embeddings. Other works like Noor and Fatima [16] worked specifically on adapting meta-learning strategies specifically for financial time series, practically showing the impact of task design and feature distribution alignment in low-resource settings. Recent studies have also introduced dynamic models tailored for long-term time series forecasting. Ma'sum et al. [17] developed a Meta-Transformer Networks model that uses attention-based dynamic modeling, which fits well with the time-varying nature of real-world data. In the context of few-shot and continual learning, works such as those by Chi et al. [18], Zhuang et al. [19], and Zhao et al. [20] proposed meta-learning strategies that could most likely be adapted to incremental time series forecasting, mostly under changing distributions or task settings. There is limited empirical proofs comparing the convergence speed, stability, and transferability of these meta-learning algorithms in temporal forecasting situations. For instance, although Wen and Li [21] improved LSTM models using attention mechanisms to make forecasting performance better, their work does not show cross-task model adaptation, which we see in most of the meta-learning approaches.

While several studies have explored meta-learning algorithms for time series tasks in various ways, a comprehensive empirical investigation of MAML and Reptile merging with deep sequence models, remains underexplored. This highlights a gap here in understanding how gradient-based meta-learners perform in dynamic time series settings/conditions and whether their theoretical advantages get translate to solve practical forecasting problems.

2.2. Limitations of Existing System / Research gap

Despite receiving more attention to meta-learning and its applications in various sectors including time series forecasting, existing research reveals several critical limitations that highlights the need for more systematic empirical investigation, particularly for meta-learning algorithms like MAML and Reptile.

- Inadequate Exploration of Base Model Influence:

The choice of base model is particularly important in time series forecasting, as temporal dependencies and

sequence modeling is crucial for predictive performance. Existing studies tend to adopt fixed or simplistic base models, mostly a feedforward or convolutional neural network therefore it will limit the study of how different architectures (e.g., LSTM, GRU, FFNN) works with meta-learning algorithms. Works such as Li et al. and Wen & Li explain the diversity of modeling requirements in time series but do not evaluate these within a context of meta-learning [2, 6].

- Limited Comparative Studies on MAML(First Order) and Reptile using Neural prediction Models:

Even though MAML and Reptile have been effectively applied to solve number of machines learning related issues, their comparative performance in context of time series problem remains underexplored. Most of the existing studies use these algorithms either for classification tasks or on single model framework without doing a thorough evaluation across various different deep learning-based models for prediction such as LSTM, GRU, FFNN, etc. For example, Van Zyl and Chang et al. used meta-learning techniques for time series tasks, but did not investigate the subtle variation in performance when these algorithms are used with various base models under the same experimental conditions [8,11].

- Less Evaluation Across Varying Inner Loop Iterations:

Meta-learning algorithms mostly rely heavily on the number of inner loop updates for proper task adaptation. However, current approaches typically keep fixing the number of iterations, while ignoring the possible difference in learning dynamics and convergence behavior that might come with different iteration counts. As mentioned by Yao et al. and Nichol et al., inner loop design plays a pivotal role in performance, yet few studies showed a systematic variation in inner loop iterations, especially in the context of time series [12].

- Training Stability and Convergence in Temporal Contexts:

The training stability and convergence behavior of MAML and Reptile algorithm on time series forecasting in existing literature remains unexplored. Although these aspects are studied in image-based tasks, temporal data introduces unique convergence challenges due to its inherent sequential dependencies and variable input lengths. Yao et al. and Nichol et al. acknowledge optimization issues but do not extend this analysis to recurrent or sequence based models [12].

- Less Focus on Model-Agnostic Meta-Learning for Time Series Problems:

While MAML and Reptile are designed to be model-agnostic, their applications for time series forecasting problems remain underdeveloped. Surveys such as those by Hospedales et al. and Son et al. primarily focus on vision and NLP tasks, leaving temporal domains relatively underexplored. The unique characteristics of time series data like seasonality, trends and temporal dependencies adds some additional challenges that are not properly solved in general meta-learning frameworks [13,22].

- Few-Shot Prediction Remains Underrepresented:

Despite the theoretical alignment of meta-learning with the concept of few-shot learning, few-shot forecasting has not received proper attention. Most of the research remains focused on classification or regression kind of problems in static domains. Although Pilyugina et al. discuss meta-learning strategies in data-scarce conditions, empirical benchmarks for few-shot time series forecasting, particularly using MAML and Reptile, remain limited [23].

- Absence of Standardized Evaluation Metrics:

Another limitation in existing literature survey is the lack of consistency in use of evaluation metrics, making cross study comparisons difficult. While metrics such as MAE, RMSE and R^2 are commonly used, they are not uniformly adopted across current meta-learning studies. Furthermore, very few works simultaneously evaluate the combined effects of base model architecture, inner loop iterations, and meta-learning algorithm on forecasting accuracy using a proper unified metric framework [23,24].

3. Proposed Approach

Time series forecasting is used in various sectors includes finance, healthcare, etc. Traditional deep learning algorithms need large amounts of data and more computational resources for training when compared to meta-learning algorithms. Also, meta-learning techniques like MAML(First Order) and Reptile provide an efficient approach for training models that generalize well across multiple tasks. This study explores a meta-learning framework using LSTM, GRU, and FFNN as base models for stock price prediction. The proposed framework consists of a meta-learning

pipeline that includes preprocessing, base model training, meta-update, and evaluation phases, as shown in Fig 1.

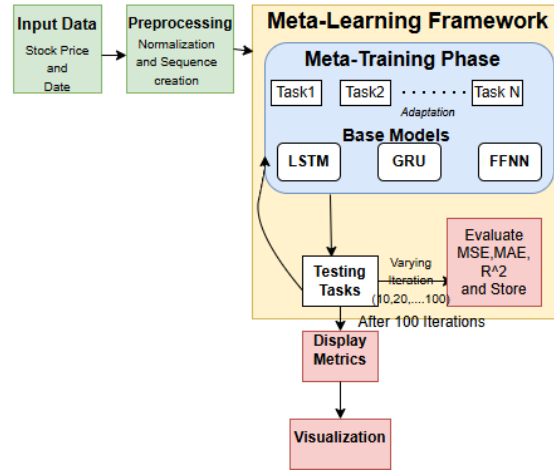


Fig.1. Meta-learning framework

3.1. Architecture/Framework

The following sections describe each component of the architecture in detail.

- **Input Data and Preprocessing:** The dataset comprises of historical stock prices and corresponding timestamps. The preprocessing step involves:

Normalization: Min-Max Scaling is applied to convert the input values which will lie within a specific range after conversion, improving model convergence.

Sequence Creation: The input sequences are created with a look-back period of 60 time steps, enabling the model to capture proper temporal dependencies.

Testing was done on the final, most recent 20 percent of the chronologically ordered dataset.

- **Meta-Training Phase:** In this phase, the base models (LSTM, GRU, FFNN) are trained on multiple tasks extracted from different time segments of the stock price data. The training process is performed under two meta-learning paradigms: Here, task is distinct and contiguous segment of historical data(271 day window). Each dataset consisting of 271 records was partitioned into 10 distinct temporal task and each task provides a 211-shot learning condition after accounting for the 60 days look back window.

MAML(first order): The model learns an initial set of parameters using a first-order approximation (ignoring second derivatives) to minimizing computational overhead which allows fast fine-tuning on unseen tasks.

Reptile: The model updates its parameters by moving towards the average parameter values obtained from multiple tasks.

- **Meta-Model and Meta-Update:** The meta-model aggregates knowledge from multiple training tasks through meta updates. In MAML, meta-updates are computed by averaging gradients across tasks, whereas Reptile uses parameter interpolation to achieve better generalization.
- **Unseen Task Evaluation:** The trained meta-model is tested on an unseen forecasting task. The evaluation is conducted using three key metrics: MSE, MAE and R^2
- **Visualization and Comparison:** Predictions obtained from LSTM, GRU, FFNN models trained with MAML and Reptile and predictions obtained from traditional models are presented strategically in single figure of all the four stocks(4 figures i.e fig 2,3,4,5).

3.2. Algorithm

Algorithm 1 First-Order MAML(FOMAML) for Stock Price Prediction (Base Model: LSTM/GRU/FFNN)

- 1: **Initialize:** Base model parameters θ , learning rates α, β
- 2: **while** not converged **do**
- 3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
- 4: **for** each task $\mathcal{T}_i$ **do**
- 5: Sample dataset $\mathcal{D}_{i^{train}}, \mathcal{D}_{i^{test}}$
- 6: Compute training loss: $\mathcal{L}_i(\theta)$ on $\mathcal{D}_{i^{train}}$

```

7:      Compute adapted parameters:  $\theta'_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_i(\theta)$ 
8:  end for
9:      Compute meta-loss:  $\sum_i \mathcal{L}_i(\theta'_i)$  on  $\mathcal{D}_i^{\text{test}}$ 
10:     Update meta-parameter:  $\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_i \mathcal{L}_i(\theta'_i)$ 
11: end while

```

MAML aims to find an optimal initial set of model parameters represented as θ , which enables quick adaptation to new or unseen tasks with minimum gradient updates. The MAML framework uses dual-loop optimization method: an inner loop focused on task-specific learning and an outer loop intended for meta-level updates across a variety of tasks. In the inner loop, a specific task, $\mathcal{T}_i$, is randomly get selected from a distribution of tasks (here 10), $p(\mathcal{T})$. For every sampled task, the model undergoes training on a support dataset, $\mathcal{D}_i^{\text{train}}$. This training makes use of a task-specific loss function, $\mathcal{L}_i(\theta)$, which for regression problems is typically the MSE. The parameters adapted for the specific task, θ'_i , are then calculated by doing a gradient descent step:

$$\theta'_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_i(\theta) \quad (1)$$

Here, α represents the learning rate used for the task-specific adaptation.

The outer loop emphasis on the meta-learning aspect. The performance of the adapted or modified parameters θ'_i is assessed on a different query dataset, $\mathcal{D}_i^{\text{test}}$, for the same task. The meta-objective is then computed by adding up the losses for every task that was taken into consideration. To guarantee computational efficiency, we used here the First-Order MAML (FOMAML) approximation. In contrast to standard MAML which need to calculate second-order derivatives (Hessian matrices). FOMAML ignores second-order terms in order to approximate the meta-gradient. This preserves strong performance while drastically lowering memory use as well as computational expense. The original model parameters i.e θ , are subsequently updated based on this aggregated meta-loss gradient:

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_i \mathcal{L}_i(\theta'_i) \quad (2)$$

In this update rule, β signifies the meta-learning rate. This iterative process encourages the meta-model to learn a parameter initialization that exhibits strong generalization capabilities across diverse forecasting or regression tasks, allowing for efficient fine-tuning on new or unseen tasks with very few gradient steps.

Algorithm 2 Reptile for Stock Price Prediction (Base Model: LSTM/GRU/FFNN)

```

1: Initialize: Base model parameters  $\theta$ , learning rates  $\alpha, \beta$ 
2: while not converged do
3:   Sample batch of tasks  $\mathcal{T}_i \sim p(\mathcal{T})$ 
4:   for each task  $\mathcal{T}_i$  do
5:     Sample dataset  $\mathcal{D}_i$ 
6:     Perform k SGD updates:  $\theta'_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_i(\theta)$ 
7:   end for
8:   Update meta-parameter:  $\theta \leftarrow \theta + \beta(\theta'_i - \theta)$ 
9: end while

```

Reptile is a meta-learning algorithm that uses a first-order approximation to estimate the meta-gradient, which avoids the computational cost associated with second-order derivatives. In contrast to MAML, Reptile updates the global meta-parameters by shifting them in the direction of the task-specific parameters that are obtained after performing several steps of standard gradient descent on individual tasks. For each task $\mathcal{T}_i$ sampled from the task distribution, the base model is trained on the task-specific dataset $\mathcal{D}_i$ for k iterative gradient descent steps. This process leads to the task-adapted parameters θ'_i . Each step in this inner loop update can be represented as:

$$\theta \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}_i(\theta) \quad (3)$$

This update is repeated k times to obtain the final task-adapted parameters θ'_i . Following this task-specific training, the meta-update is performed according to the following rule:

$$\theta \leftarrow \theta + \beta(\theta'_i - \theta) \quad (4)$$

This meta-update effectively moves the initial parameters θ towards the optimized parameters (optimum value) θ'_i learned for various tasks or jobs. This approach gives quick convergence and effective adaptation to new or unseen tasks while doing the fine-tuning process.

3.3. Details of Hardware & Software

The experimental setup for this investigation was implemented using Google Colaboratory (Colab), a cloud-based interactive development environment or notebook that supports GPU and TPU acceleration for executing intense codes in various programming languages. Using Colab notebooks cloud infrastructure it gives the seamless execution of computationally intensive meta-learning algorithms, particularly MAML and Reptile, applied to time-series forecasting tasks. The virtual machine environment was equipped with an Intel Xeon processor and where applicable, computation was accelerated using an NVIDIA Tesla T4 GPU. The notebook provided 12 GB of dynamic RAM and integration with Google Drive was utilized for efficient data management, reading dataset and model checkpointing.

From a software perspective, the experiment was carried out in a Python 3.9+ environment along with PyTorch (version 1.13 or higher) as the primary deep learning framework. several other libraries were utilized to support different phases of the modeling pipeline: NumPy used for performing various numerical operations, Pandas used for performing data preparation and manipulation, and Scikit-learn used here for performing normalization procedures and evaluation metrics computation. Matplotlib was used to generate performance visualizations, while PrettyTable given organized display of quantitative results in tables. The Google Drive API was used here to safely access and save data while making sure that there is reliable and consistent file management during sessions.

The integration of these tools made experimentation easy as well as efficient with the utilization of neural architectures as base models under the meta-learning paradigm, supporting both scalability and reproducibility. This environment proved particularly effective for training and evaluating complicated models across various stock datasets, guaranteeing that the computational requirements of the investigation were satisfied without sacrificing performance as well as flexibility.

3.4. Evaluation Parameters

To evaluate the effectiveness of the proposed meta-learning models, namely MAML(First Order) and Reptile, when integrated with various base learner models like LSTM, GRU, and FFNN for the tasks of stock price prediction, here we have employed the following standard performance metrics:

Root Mean Square Error (RMSE)

RMSE is a commonly used metric in regression based analysis to quantify the average magnitude of the errors between predicted and actual values. RMSE gives higher weight to larger prediction errors due to the squaring of difference. It is mathematically defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (5)$$

In Eq.5, y_i represents the actual or label value $\hat{y}_i$ denotes the predicted value, and n is the total number of input data points.

Mean Absolute Error (MAE)

MAE gives a measure of the average absolute difference between the predicted and observed values. Unlike RMSE, MAE treats all prediction errors uniformly, without differential weighting based on their magnitude. The formula for MAE is:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (6)$$

In Eq. 6, y_i is the actual or label value $\hat{y}_i$ is the predicted value, and n is the number of input data points.

Coefficient of Determination (R^2 Score)

R^2 score is a statistical measure that represents the proportion of the variance in the dependent variable that is predictable from the independent variables. It essentially indicates the goodness of fit of the model to the data. The R^2 score is calculated as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (7)$$

In Eq. 7, y_i represents the actual or label value, $\hat{y}_i$ is the predicted value, and $\bar{y}$ is the mean of the actual values. An R^2 score to 1 shows a better fit of the model to the data.

4. Experiments

This section explains a comprehensive evaluation of stock price prediction models using two well known meta-

learning approaches like MAML(First Order) and Reptile integrated with three neural network architectures: LSTM, GRU, and FFNN. The evaluation or investigation was done on four different stocks dataset those are TCS, TATASTEEL, GRASIM, and DJIA index—to check robustness and generalizability of the models across varying market behaviors. The Nifty Fifty dataset (2010–2021) was used as the source for TCS, TATASTEEL, and GRASIM, while the DJIAHD dataset (2009–2019) was used for U.S.-based stocks(Dow Jones Industrial Average Index). Each model was trained and tested using both meta-learning frameworks with a varying number of iterations, ranging from 10 to 100. Performance was rigorously evaluated using three standard statistical metrics: RMSE, MAE, and R^2 .

Fig. 2 and Table 1 together explain the comparative performance of meta-learning strategies: MAML and Reptile applied to three different deep learning architectures (LSTM, GRU, and FFNN) for prediction of the price of the TCS stock. When compared to conventional methods like ARIMA, SARIMA, ETS, and SES, which are unable to capture the intricacy and dynamic variations of stock price movements, visual analysis shows that models utilizing meta-learning, especially GRU and LSTM, produce forecasts that are significantly more accurate. Although Reptile doing consistently well, MAML shows more better improvements as the iteration count rises, suggesting better adaptability or learning for LSTM. The results confirm that GRU using with MAML is the most effective combination among all the models tested, achieving the lowest RMSE (151.13), the lowest MAE (120.05), and the highest R^2 score (0.90) at 80,90 iterations. FFNN found less suitable for temporal patterns in stock prices, with all meta learning method, as it shows very less gains over iterations and limited ability to capture time dependencies. In general, MAML proves to be the more powerful meta-learning strategy, and when paired with recurrent architectures, especially GRU, it significantly improves predictive accuracy in stock market modeling.

Note: To validate the statistical significance of our observations, we did the Diebold-Mariano (DM) test. Comparing the proposed best performing MAML(First Order) with GRU model against the ARIMA baseline we got a p-value of $p < 0.001$. Furthermore, a comparison against the alternative meta-learning approach (Reptile) also resulted in a significant p-value ($p < 0.001$). These results confirm that the performance improvements of the MAML-GRU framework are statistically significant and not due to random chance or luck.

Table 1. Performance comparison of MAML(First Order) and reptile with different base models and iterations on tcs stock data

TCS Stock		Metric	Number of Iterations									
			10	20	30	40	50	60	70	80	90	100
LSTM	MAML	RMSE	447.51	456.11	332.76	370.97	428.42	480.09	302.94	194.95	308.37	204.55
		MAE	390.89	351.26	269.47	317.03	369.77	445.66	245.01	134.87	264.83	148.49
		R^2	0.41	0.39	0.68	0.60	0.46	0.32	0.73	0.89	0.72	0.88
	Reptile	RMSE	233.80	238.95	226.21	223.95	199.96	489.65	305.79	359.89	371.40	281.54
		MAE	185.69	191.11	177.71	175.88	146.64	451.77	265.46	318.47	332.34	240.94
		R^2	0.83	0.83	0.85	0.85	0.88	0.29	0.72	0.62	0.59	0.76
GRU	MAML	RMSE	445.69	326.15	202.41	283.11	203.96	359.64	187.42	151.13	180.62	217.27
		MAE	366.86	286.19	141.98	242.65	149.44	317.74	129.67	158.89	120.05	166.73
		R^2	0.41	0.68	0.87	0.76	0.87	0.62	0.89	0.86	0.90	0.86
	Reptile	RMSE	220.04	222.88	225.68	231.44	240.02	242.52	243.17	244.36	245.32	246.09
		MAE	170.25	173.61	176.85	183.29	192.42	195.00	195.71	196.96	197.93	198.69
		R^2	0.85	0.85	0.85	0.84	0.83	0.82	0.82	0.82	0.82	0.82
FFNN	MAML	RMSE	248.55	248.97	249.24	249.71	250.37	251.31	251.92	251.00	252.31	252.56
		MAE	185.62	188.37	189.41	190.74	192.25	193.35	193.91	195.09	195.93	196.37
		R^2	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
	Reptile	RMSE	256.85	256.86	256.87	256.92	256.92	256.93	256.94	256.95	256.96	256.97
		MAE	202.78	202.82	202.87	202.96	202.97	202.98	202.99	203.01	203.03	203.84
		R^2	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88

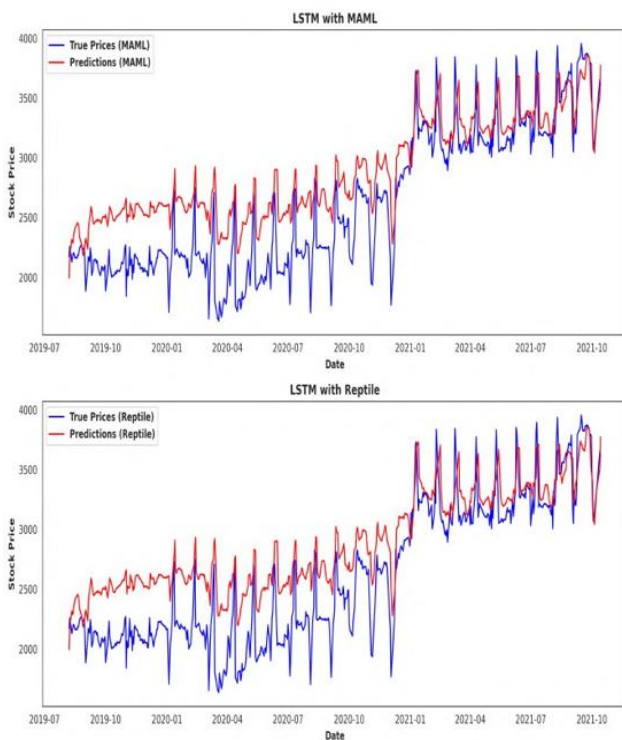
Fig. 3 and Table 2 together illustrate the comparative performance of meta-learning strategies: MAML and Reptile, when integrated with the LSTM, GRU, and FFNN models to forecast TATASTEEL stock prices. The visual analysis through charts/graphs shows that meta learning methods particularly when combined with recurrent architectures like LSTM and GRU, significantly give better plots than traditional models in catching the intricacy as well as dynamic fluctuations of stock prices. While Reptile consistently doing well, MAML shows good improvements as the iteration count rises, especially with LSTM as base model, explaining superior adaptability with LSTM in learning temporal features over the time. The results confirm that GRU using with MAML is the best combination for getting best results, achieving the lowest RMSE (109.88), the lowest MAE (71.22), and the highest R^2 score (0.90) at 60 iterations. On the

other hand, FFNN observed to be less appropriate for modeling temporal patterns in stock prices, showing minimum improvement regardless of whether MAML or Reptile is used. In essence we can say that the combination of MAML with GRU consistently delivers superior predictive accuracy, affirming MAML's strength as a meta-learning strategy in handling time series forecasting tasks such as stock price prediction.

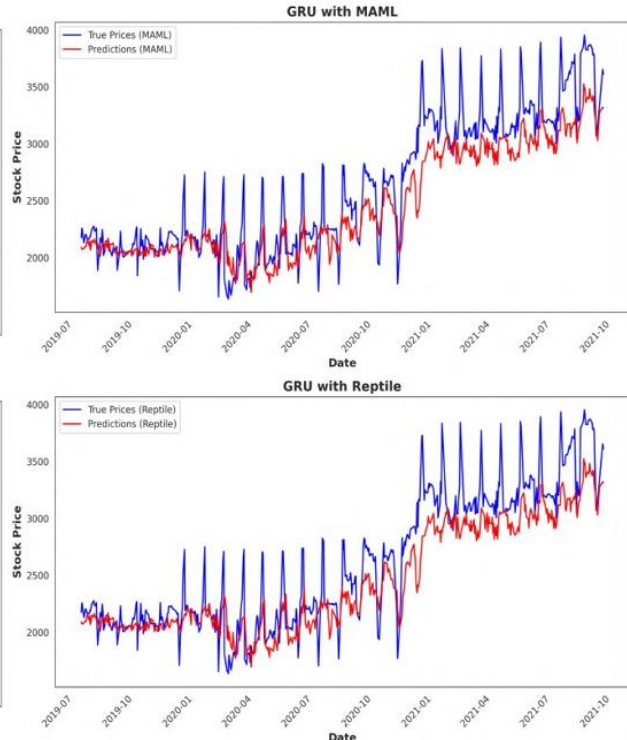
Table 2. Performance comparison of MAML(First Order) and reptile with different base models and iterations on tatasteel stock data

TATASTEEL Stock		Metric	No of Iterations									
			10	20	30	40	50	60	70	80	90	100
LSTM	MAML	RMSE	302.86	167.51	156.03	151.88	142.06	135.45	120.36	174.57	123.57	146.43
		MAE	210.51	100.59	100.88	106.19	99.06	90.66	73.64	116.52	79.38	106.14
		R ²	0.28	0.78	0.81	0.82	0.84	0.85	0.88	0.76	0.88	0.83
	Reptile	RMSE	144.96	143.66	138.39	116.98	119.72	139.76	273.67	279.38	382.77	381.58
		MAE	108.28	110.61	105.84	86.51	89.11	105.47	209.73	198.88	212.72	205.12
		R ²	0.83	0.83	0.85	0.89	0.88	0.84	0.41	0.39	0.28	0.29
GRU	MAML	RMSE	303.41	136.31	130.99	129.53	134.31	109.88	118.43	125.52	134.02	124.59
		MAE	216.18	83.49	83.02	78.47	79.46	71.22	77.06	80.02	99.76	84.18
		R ²	0.28	0.85	0.86	0.86	0.85	0.90	0.89	0.87	0.85	0.87
	Reptile	RMSE	147.79	125.41	118.04	120.22	120.89	119.04	119.52	117.01	119.79	122.47
		MAE	119.64	91.58	79.68	83.79	85.23	82.42	83.81	80.64	85.48	89.50
		R ²	0.83	0.87	0.89	0.88	0.88	0.88	0.88	0.89	0.88	0.88
FFNN	MAML	RMSE	277.71	272.52	273.95	274.25	288.45	298.56	298.28	297.98	298.38	298.76
		MAE	185.23	182.08	182.88	182.92	190.62	196.21	195.89	195.53	195.48	195.66
		R ²	0.39	0.42	0.41	0.41	0.35	0.30	0.30	0.30	0.30	0.30
	Reptile	RMSE	216.98	233.09	233.17	233.25	233.53	233.54	233.55	233.56	233.57	233.59
		MAE	182.81	205.28	205.42	205.56	205.93	205.95	205.97	206.00	206.03	206.07
		R ²	0.63	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57

Stock Price Prediction using LSTM with Meta-Learning



Stock Price Prediction using GRU with Meta-Learning



Stock Price Prediction using FFNN with Meta-Learning

Stock Price Prediction using Traditional Algorithms

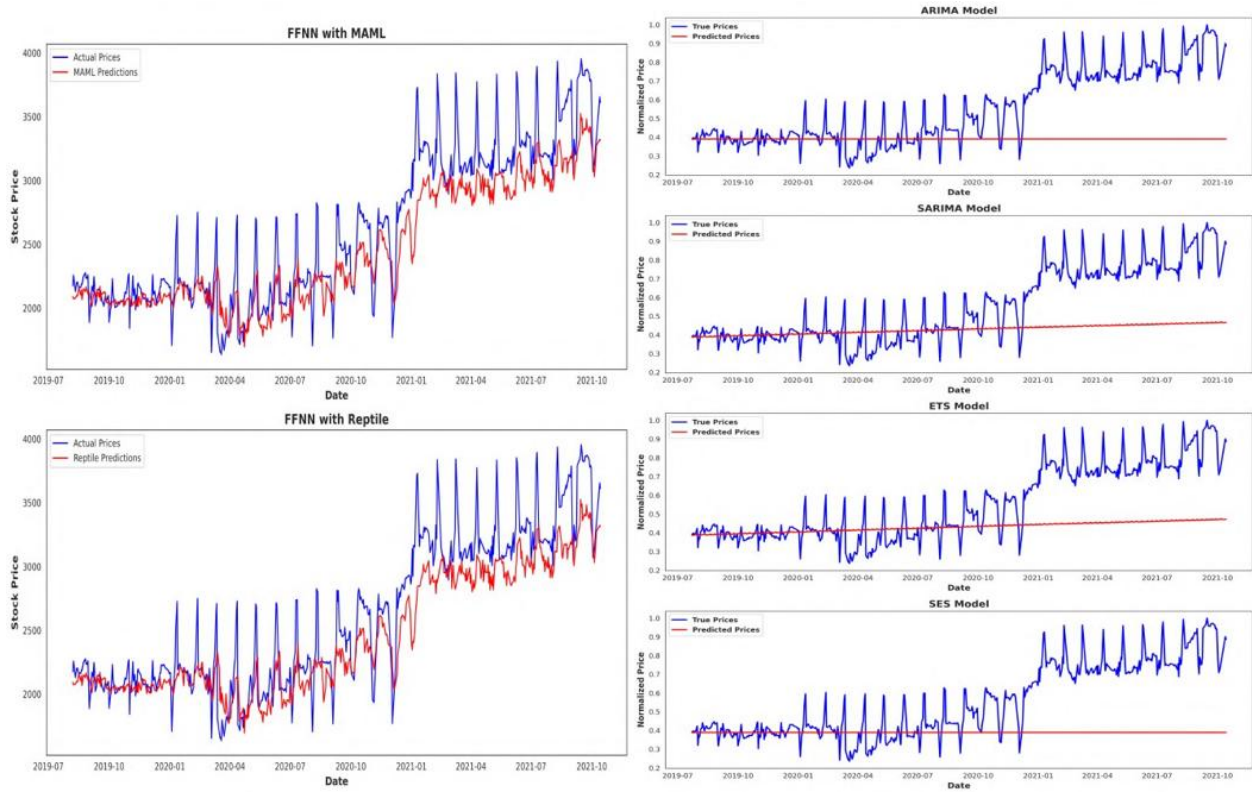
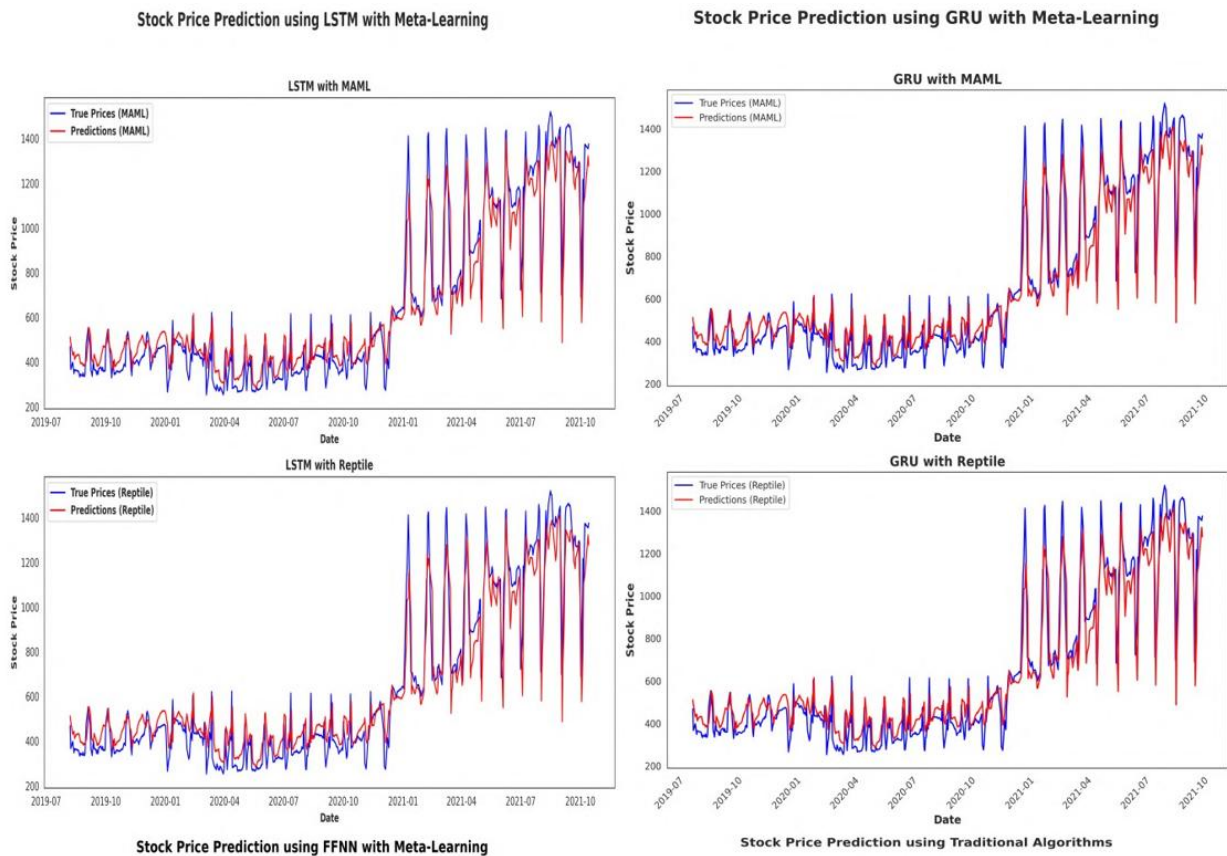


Fig.2. Performance comparison of MAML(First Order) and reptile with different base models on tcs stock data



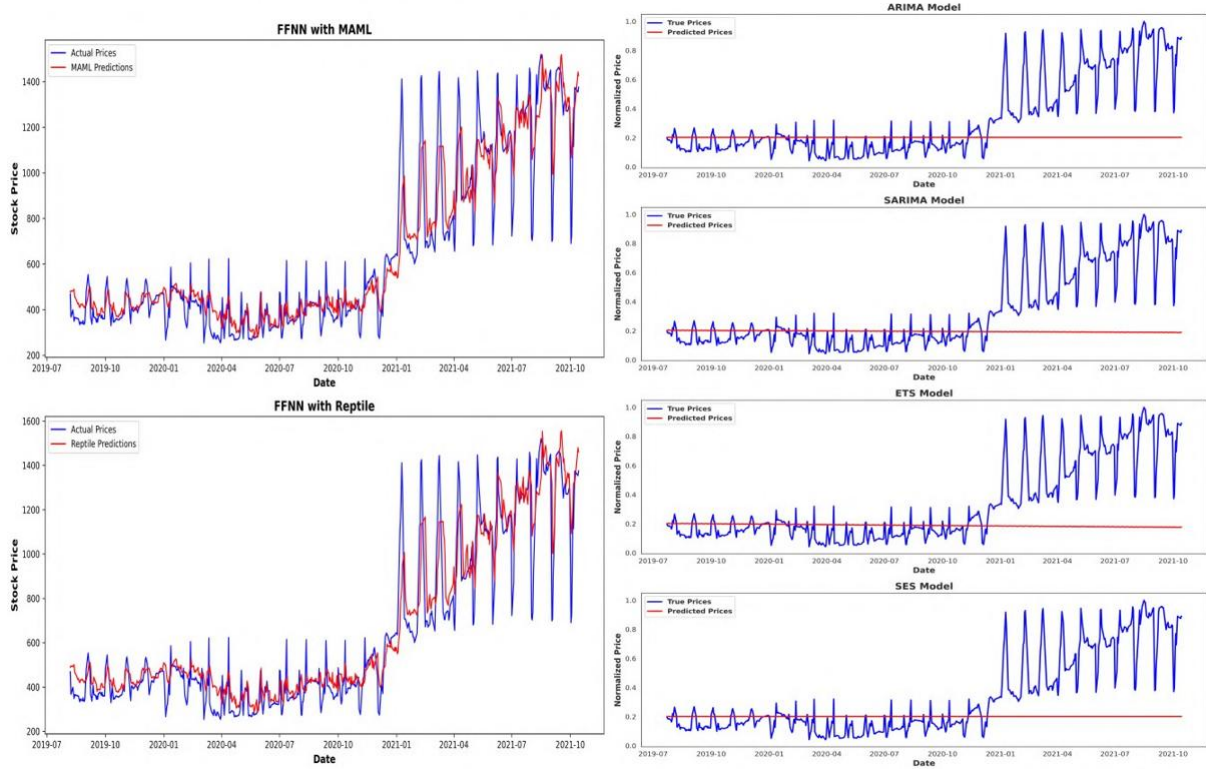


Fig.3. Performance comparison of MAML(First Order) and reptile with different base models on TATASTEEL stock data

Table 3. Performance comparison of MAML(First Order) and reptile with different base models and iterations on GRASIM stock data

GRASIM Stock		Metric	No of Iterations									
			10	20	30	40	50	60	70	80	90	100
LSTM	MAML	RMSE	622.94	295.44	316.18	323.42	229.94	178.42	253.17	113.21	169.53	112.60
		MAE	543.98	239.36	273.09	283.92	195.44	152.12	223.28	81.55	134.22	80.67
		R ²	-20.32	0.22	0.18	0.59	0.75	0.50	0.90	0.77	0.98	0.98
	Reptile	RMSE	111.96	109.29	108.20	106.93	105.38	119.56	104.22	105.00	105.55	112.06
		MAE	80.47	76.82	76.11	75.55	75.11	93.71	74.68	76.47	77.78	84.93
		R ²	0.90	0.90	0.90	0.91	0.91	0.88	0.91	0.91	0.91	0.90
GRU	MAML	RMSE	355.11	296.57	162.47	142.84	519.06	218.08	117.13	100.22	110.97	106.38
		MAE	310.79	262.42	135.44	112.71	466.24	195.22	76.76	64.22	75.49	75.14
		R ²	0.02	0.31	0.79	0.84	-1.08	0.63	0.89	0.92	0.98	0.91
	Reptile	RMSE	105.25	109.25	113.00	124.21	129.12	131.60	132.56	129.95	134.24	135.24
		MAE	75.48	71.95	85.93	97.86	102.75	105.57	106.93	104.98	109.66	109.74
		R ²	0.91	0.90	0.90	0.88	0.87	0.86	0.86	0.86	0.86	0.85
FFNN	MAML	RMSE	161.31	190.73	179.60	181.08	181.43	181.57	182.33	198.89	246.42	245.19
		MAE	116.91	143.68	140.62	145.84	146.56	146.77	147.91	163.99	192.53	191.71
		R ²	0.79	0.71	0.75	0.74	0.74	0.74	0.74	0.60	0.52	0.53
	Reptile	RMSE	268.98	287.97	293.19	300.21	309.13	319.88	332.65	335.96	335.97	336.01
		MAE	229.52	247.01	251.78	258.13	266.19	275.95	287.69	298.74	290.75	290.78
		R ²	0.43	0.35	0.33	0.30	0.25	0.20	0.14	0.12	0.12	0.12

Fig. 4 and Table 3 present the comparative results of MAML(First Order) and Reptile meta-learning algorithms with LSTM, GRU, and FFNN as base learner models for predicting GRASIM stock prices. The visual plots and metrics collectively show the superior learning capabilities of MAML, particularly when used with recurrent models like GRU and LSTM, in capturing the temporal dependencies within financial time series data. LSTM with MAML demonstrating

a good improvement in performance with increased iteration counts, achieving the best results at 100 iterations with an RMSE of 112.60, MAE of 80.67, and R^2 of 0.98. On the other hand, Reptile maintains continuously stable performance across various iterations, but does not able to reach the peak accuracy of MAML. Similarly, GRU with MAML gives excellent predictive performance, reaching at 80 iterations with an RMSE of 100.22, MAE of 64.22, and R^2 of 0.92, strengthening its ability to model sequential data effectively. Conversely, the model using FFNN as base neural architecture performs poorly with both MAML and Reptile, showing higher error and worse R^2 values over all the iterations, showing its flaws in handling time-dependent patterns. Traditional methods such as ARIMA, SARIMA, ETS, and SES are also shown in Fig. 4 for baseline comparison but fall short of matching the accuracy and responsiveness of meta learning enhanced deep learning models. These findings suggest that MAML-based meta-learning significantly boosts the adaptability and accuracy of deep models, particularly LSTM and GRU, for dynamic stock forecasting tasks.

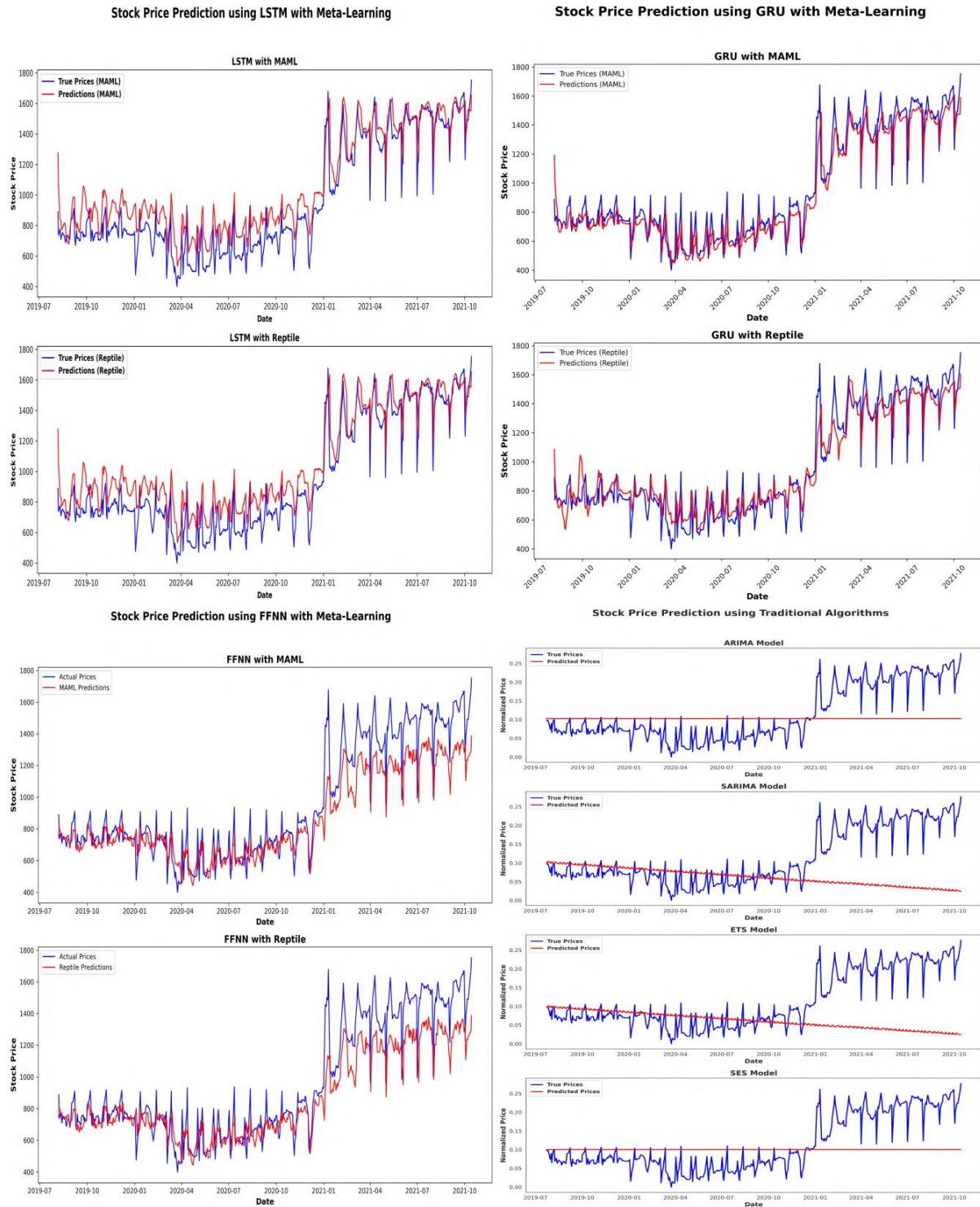


Fig.4. Performance comparison of MAML(First Order) and reptile with different base models on GRASIM stock data

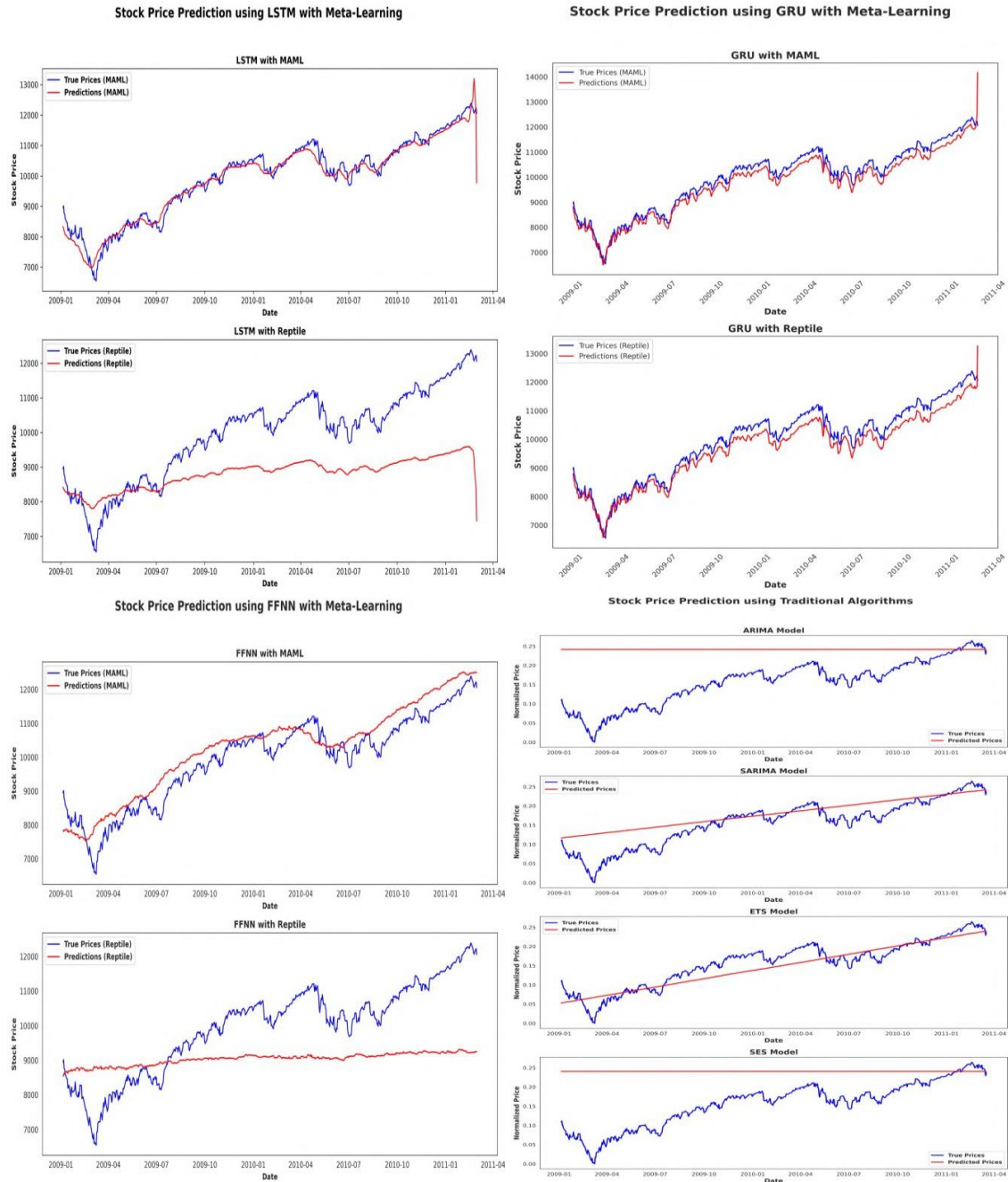


Fig.5. Performance comparison of MAML(First Order) and reptile with different base models on DJIAHD stock data

Fig. 5 and Table 4 explain the comparative analysis of MAML and Reptile meta-learning framework used with LSTM, GRU, and FFNN as base models for the prediction of DJIAHD stock dataset. The performance patterns show the notable adaptability of MAML, particularly when combined with LSTM, where the model exhibit very good improvements over successive iterations while reaching peak performance at 100 iterations with RMSE of 278.06, MAE of 184.07, and a high R^2 of 0.95. Similarly, GRU with MAML algorithm gives strong predictive accuracy at 30 iterations, boasting RMSE of 281.17, MAE of 206.03, and R^2 of 0.95, though its performance is less stable as compared to LSTM. On the other hand, Reptile shows relatively consistent performance across iteration counts but does not able to reach the best results achieved by MAML. For instance, here reptile with LSTM hovers around RMSE values of 309–394 with R^2 0.91, showing reliability but limited amount of improvement. Notably, FFNN performs bad with both meta-learning approaches, particularly with Reptile, where the R^2 remains around 0.01, indicating poor generalization. Various traditional models like ARIMA, SARIMA, ETS, and SES, plotted in Fig. 5, not adequately capture the market dynamics effectively which shows the superior temporal learning capability of deep learning models, especially when used with meta-learning. Overall, these findings shows that MAML provides significant improvements in learning efficiency and accuracy for stock price forecasting, especially when combined with recurrent architectures base models like LSTM and GRU.

Table 4. Performance comparison of MAML(First Order) and reptile with different base models and iterations on DJIAHD stock data

DJIAHD Dataset		Metric	No of Iterations									
			10	20	30	40	50	60	70	80	90	100
LSTM	MAML	RMSE	2197.38	1173.06	1122.69	609.41	757.18	495.14	1052.55	458.86	838.02	278.06
		MAE	1888.63	1038.35	982.19	522.98	653.68	406.62	995.52	387.45	783.38	184.07
		R ²	-1.94	0.16	0.23	0.77	0.65	0.85	0.32	0.87	0.57	0.95
	Reptile	RMSE	309.51	394.38	380.66	392.62	380.77	405.84	408.57	396.99	387.63	380.31
		MAE	238.83	331.33	321.82	335.19	325.47	349.37	351.73	341.08	331.64	324.62
		R ²	0.94	0.90	0.91	0.90	0.91	0.89	0.89	0.90	0.90	0.91
GRU	MAML	RMSE	1357.56	612.77	281.17	1165.88	286.24	1028.87	1007.02	1051.64	898.07	720.30
		MAE	1032.72	541.53	206.03	1130.88	225.37	1005.82	995.85	1040.99	886.68	699.25
		R ²	-0.12	0.77	0.95	0.17	0.95	0.35	0.38	0.32	0.50	0.68
	Reptile	RMSE	720.33	720.33	720.22	719.91	718.96	717.36	715.25	712.98	736.48	766.76
		MAE	699.73	700.51	701.77	703.04	703.57	702.99	701.62	699.73	725.42	756.16
		R ²	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.69	0.66	0.64
FFNN	MAML	RMSE	864.16	792.92	775.17	774.46	773.86	773.45	774.31	773.71	773.58	772.91
		MAE	657.36	615.16	602.86	602.21	601.66	601.17	600.13	599.61	599.27	598.74
		R ²	0.54	0.61	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.63
	Reptile	RMSE	1269.77	1270.22	1270.89	1271.78	1272.89	1274.26	1275.85	1277.69	1279.81	1282.13
		MAE	1116.65	1117.05	1117.64	1118.43	1119.42	1120.64	1122.04	1123.67	1125.54	1127.60
		R ²	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00

5. Conclusions

In this paper we investigated the potential of meta-learning algorithms, to be specific MAML(First Order) and Reptile algorithms, for time series forecasting problem in financial domain. Through a comprehensive evaluation using various real-world company stocks like TCS, Tata Steel, Grasim, and DJIAHD, we demonstrated(from visualization) that meta-learning approaches consistently perform better than traditional forecasting models used, especially in dynamic and volatile environments. Here MAML was proven to be effective in catching stable trends and adapting quickly to minor variation. On the other hand, reptile showed enhanced generalization as well as robustness across various stocks. While helpful in some situations, various traditional models such as ARIMA, SARIMA, etc. showed limits with regard to performance and flexibility across a variety of financial time series stock datasets. The comparative analysis through visual graphs further explains the pros of meta-learning methods in forecasting accuracy and learning efficiency. Overall, this strategic investigation validates the use of MAML and Reptile as effective alternative when compared against conventional time series prediction methods. Additionally, it creates new pathways for robust, adaptive, and task-efficient time series modeling in financial markets.

Although the efficiency of meta-learning methods like MAML and Reptile for financial time series forecasting is the primary focus of the current work done, there are still a numerous other areas that might be investigated and studied in the future. One of the potential directions is the study of various other meta-learning algorithms and optimization approaches to try and further increase adaptability as well as convergence speed. Using specific domain expertise or external factors like macroeconomic indicators, sentiment analysis or news-based signals may increase forecasting accuracy, especially during high-volatility conditions. Furthermore, broadening the current framework to multi-step forecasting and testing across a wider range of asset classes, including cryptocurrencies, commodities and data-scarce emerging markets could validate the applicability of these methods and their potential to support robust portfolio diversification. Finally, real-time implementation of meta-learned models in a live trading environment could give us valuable insights into their practical feasibility and impact on making correct decisions in financial markets.

Author Contributions Statement

Pratik Zinjad– Conceptualization, Methodology, Implementation, Data Curation, and Writing – Drafted the initial manuscript.

Tushar Ghorpade– Supervision, Formal Analysis, Validation, and Writing – Review and Editing.

Vanita Mane– Supervision, Performance Evaluation, and Writing – Review and Editing.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

This research received no external funding.

Data Availability Statement

This study analyzed publicly available datasets (Nifty Fifty and DJIAHD). The datasets can be found here: “<https://www.kaggle.com/datasets/setseries/nifty50-stocks-dataset20102021>”, <https://www.kaggle.com/datasets/mnassrib/dow-jones-industrial-average>”, accessed on “20-05-2024”.

Ethical Declarations

The authors declare that no human subjects or animals were involved in this study.

Acknowledgments

We sincerely thank the experts for their professional evaluation and valuable recommendations, which have contributed to improving the quality of the experiment and the reliability of its results.

Declaration of Generative AI in Scholarly Writing

During the preparation of this work, the authors used ChatGPT in order to improve readability and grammar. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Abbreviations

The following abbreviations are used in this manuscript:

MAML - Model-Agnostic Meta-Learning
 FOMAML - First-Order Model-Agnostic Meta-Learning
 LSTM - Long Short-Term Memory
 GRU - Gated Recurrent Unit
 FFNN - Feed Forward Neural Network
 RMSE - Root Mean Square Error
 MAE - Mean Absolute Error
 ARIMA - Autoregressive Integrated Moving Average
 SARIMA - Seasonal Autoregressive Integrated Moving Average
 AI - Artificial Intelligence
 DL - Deep Learning

References

- [1] Subhrajit Samanta, Shubhangi Ghosh, and Suresh Sundaram, “A Meta-cognitive Recurrent Fuzzy Inference System With Memory Neurons (McRFIS-MN) and its Fast Learning Algorithm for Time Series Forecasting,” in *IEEE Symposium Series on Computational Intelligence SSCI*, 2018.
- [2] P. H. Li, D. X. Chen, H. Y. Huang, X. T. Wu, H. W. Jiang, and G. L. Ji, “A Novel Hybrid Scheme for Time Series Prediction Using LMS Filter and ISSA-based LSTM,” in *2022 IEEE Smartworld, Ubiquitous Intelligence Computing, Scalable Computing Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*, Haikou, China, 2022, pp.343-350,doi:10.1109/SmartWorld-UIC-ATC-ScalCom-DigitalTwin-PriCompMetaverse56740.2022.00070.
- [3] Chelsea Finn, Pieter Abbeel, and Sergey Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML’17)*, JMLR.org, 2017, pp. 1126–1135.
- [4] Alex Nichol, Joshua Achiam, and John Schulman, “On First-Order Meta-Learning Algorithms,” *ArXiv abs/1803.02999*, 2018.
- [5] Lo, A. W. (2004). The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective. *Journal of Portfolio Management*, 30(5), 15–29.
- [6] X. Wen and W. Li, “Time Series Prediction Based on LSTM-Attention-LSTM Model,” in *IEEE Access*, vol. 11, pp. 48322-48331, 2023, doi: 10.1109/ACCESS.2023.3276628.
- [7] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.
- [8] Shin-Hung Chang, Cheng-Wen Hsu, Hsing-Ying Li, Wei-Sheng Zeng, and Jan-Ming Ho, “Short-Term Stock PriceTrend Prediction Using Meta-Learning,” 2021, pp. 978-1-6654-4207-7/21.
- [9] Yanbing Song, Shaofei Zang, Jianwei ma, Huimin Li, and Jinfeng Lv, “Stock prediction based on Weighted meta extreme learning machine,” 2024, pp. 979-8-3503-7922-8/24.

- [10] M. Maya, W. Yu, and X. Li, "Time series forecasting with missing data using neural network and meta-transfer learning," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, Orlando, FL, USA, 2021, pp. 1-6, doi: 10.1109/SSCI50451.2021.9659864.
- [11] Terence L. van Zyl, "Late Meta-learning Fusion Using Representation Learning for Time Series Forecasting," <https://doi.org/10.48550/arXiv.2303.11000>.
- [12] Xiao Yao, Jianlong Zhu, Guanying Huo, Ning Xu, Xiaofeng Liu, and Ce Zhang, "Model-agnostic multi-stage loss optimization meta learning," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 2349–2363, 2021, doi: <https://doi.org/10.1007/s13042-021-01316-6>.
- [13] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey, "Meta-Learning in Neural Networks: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, September 2022.
- [14] Asit Barman, Swalpa Roy, Swagatam Das, and Paramartha Dutta, "Exploring the Horizons of Meta-Learning in Neural Networks: A Survey of the State-of-the-Art," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1-16, 2024, doi: 10.1109/TETCI.2024.3502355.
- [15] Małgorzata Gutowska, Suzanne Little, and Andrew Mccarren, "Constructing a meta-learner for unsupervised anomaly detection," *10.48550/arXiv.2304.11438*, 2023.
- [16] K. Noor and U. Fatima, "Meta Learning Strategies for Comparative and Efficient Adaptation to Financial Datasets," in *IEEE Access*, vol. 13, pp. 24158–24170, 2025, doi: 10.1109/ACCESS.2024.3516490.
- [17] M. Ma'sum, Md. Rasel Sarkar, Mahardhika Pratama, Savitha Ramasamy, S.G. Anavatti, Lin Liu, H. Habibullah, and Ryszard Kowalczyk, "Dynamic Long-Term Time-Series Forecasting via Meta Transformer Networks," *IEEE Transactions on Artificial Intelligence*, pp. 1-11, 2024, doi: 10.1109/TAI.2024.3365775.
- [18] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep. 1648, 2009.
- [19] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, and J. Tang, "MetaFSCIL: A meta-learning approach for few-shot class incremental learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14146–14155.
- [20] H. Zhuang, Z. Weng, R. He, Z. Lin, and Z. Zeng, "GKEAL: Gaussian kernel embedded analytic learning for few-shot class incremental task," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7746–7755.
- [21] Xianyun Wen and Weibang Li, "Time series prediction based on LSTM-attention-LSTM model," *IEEE Access*, pp. 1-1, 2023, doi: 10.1109/ACCESS.2023.3276628.
- [22] Jaehyeon Son, Soochan Lee, and Gunhee Kim, "When Meta-Learning Meets Online and Continual Learning: A Survey," *IEEE transactions on pattern analysis and machine intelligence*, pp. 1-1, 2024, doi: 10.1109/TPAMI.2024.3463709.
- [23] Polina Pilyugina, Svetlana Medvedeva, Kirill Mosievich, Ilya Trofimov, Alina Kostromina, Dmitry Simakov, and Evgeny Burnaev, "A Large-Scale Empirical Study of Aligned Time Series Forecasting," *IEEE Access*, pp. 1-1, 2024, doi: 10.1109/ACCESS.2024.3458391.
- [24] Anna Vettoruzzo, M.-R. Bouguelia, Joaquin Vanschoren, Thorsteinn Rognvaldsson, and Kc Santosh, "Advances and Challenges in Meta-Learning: A Technical Review," *IEEE transactions on pattern analysis and machine intelligence*, pp. 1-1, 2024, doi: 10.1109/TPAMI.2024.

Authors' Profiles



Mr. Pratik Zinjad was born in Mumbai, Maharashtra, India, in 1999. He received the B.E. degree in computer engineering from Xavier Institute of Engineering, Mumbai, India, in 2021. He is currently pursuing the M.Tech degree in computer engineering at Ramrao Adik Institute of Technology, D. Y. Patil Deemed to be University, Navi Mumbai, India. His major field of study is artificial intelligence, with a focus on meta-learning and time series forecasting.

He is currently working as a Teaching Assistant at Saraswati College of Engineering, Kharghar, since 2024.

Previously, he served as a Lab Assistant at K.C. College of Engineering, Thane (2022–2023), and as an Assistant System Engineer at Tata Consultancy Services, where he worked on the TATA Digital Project (2021–2022). He has authored two research papers and has attended one international conference held at TSEC, Bandra. His current research interests include meta-learning, deep learning for time series forecasting, and optimization techniques.

Mr. Zinjad is a UGC-NET Junior Research Fellowship (JRF) awardee (Dec 2024), and has also qualified the UGC-NET in Dec 2023 and June 2024 for Assistant Professor and Ph.D. He is a Gold Badge holder on HackerRank for Python problem solving. He has completed multiple NPTEL certifications from IIT Madras and IIT Kharagpur. He has also served as Presiding Officer in the Indian Assembly Election 2024 and was a Webcasting Operator in the 2019 election. He is a member of several academic communities and has received positive student feedback in various teaching labs.



Dr. Tushar Ghorpade received his BE in Computer Engineering from North Maharashtra University in June 2005, ME in Computer Engineering from University of Mumbai in June 2012 and PhD in Department of Computer Engineering from University of Mumbai. He is presently holding position of an Assistant Professor in Computer Engineering at the Ramrao Adik Institute of Technology, Nerul, Navi Mumbai. He has published about 45+ papers in international and national journals and conferences. He has guided 20+ PG students and two are ongoing. He is a life member of Indian Society for Technical Education (ISTE) also active faculty coordinator of Computer Society of India (CSI) in RAIT. His research area includes data science, deep learning, machine learning and natural language processing.



Dr. Vanita Mane received the M.E. and Ph.D. in Computer engineering from Mumbai University. She has been dedicatedly working as an Associate Professor in Computer Engineering Department of RAIT, Navi Mumbai, India for the past 22 years. She has published 95 research papers in International Journals and International Conferences. Additionally, she has 2 Patents and 4 Copyrights on her name. Her area of specialization includes Digital Forensics, Cyber Security, Image Processing and Pattern Recognition.

How to cite this paper: Pratik Zinjad, Tushar Ghorpade, Vanita Mane, "Meta-learning Approach for Time Series Forecasting: First-order MAML and Reptile", International Journal of Intelligent Systems and Applications(IJISA), Vol.18, No.2, pp.27-43, 2026. DOI:10.5815/ijisa.2026.02.03