MAMBA WITH MARKOV CHAIN ON TIME SERIES DATASETS WITH DOMAIN GENERALIZATION

A PROJECT REPORT

Submitted by

ANIRUDH V
(CH.SC.U4AIE23003)

NITHYA SHANKAR.S
(CH.SC.U4AIE23048)

VIGNESHWARRAN S.
(CH.SC.U4AIE23061)

BACHELOR OF TECHNOLOGY IN ARTIFICIAL INTELLIGENCE AND ENGINEERING

Under the guidance of

DR. G.BHARATHI MOHAN

Submitted to

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING

CHENNAI – 601103

April 2025

BONAFIDE CERTIFICATE

This is to Certify that this project report entitled **“Mamba with markov chain on time series datasets with domain generalization”** is the Bonafide work of **ANIRUDH V (CH.SC.U4AIE23003), NITHYA SHANKAR S(CH.SC.U4AIE23048), VIGNESHWARRAN S (CH.SC.U4AIE23061)** under my direct supervision for End semester evaluation for the course 22AIE213 – Machine Learning in the even semester of the Academic Year 2024-2025.

SIGNATURE

BHARATHI MOHAN.G

Department of CSE-AIE.

Associate Professor, Dept.of CSE-AIE.

Amrita School of Computing

Chennai

INTERNAL EXAMINER

`

AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956

SCHOOL OF COMPUTING CHENNAI

DECLARATION BY THE CANDIDATE

We solemnly declare that the project titled " **MAMBA WITH MARKOV CHAIN ON TIME SERIES DATASETS WITH DOMAIN GENERALIZATION "** submitted for the End Semester Project evaluation during the **even semester of the academic year 2024-2025**, for the course **22AIE213 – Machine Learning**, offered by the Department of Computer Science Engineering - Artificial Intelligence, is a true representation of our original work. This project was undertaken under the guidance of **Dr. G. Bharathi Mohan**, Department of Computer Science Engineering - Artificial Intelligence, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai Campus. We further affirm that the contents of this project have not been submitted nor will be submitted to any other institution or entity for the fulfillment of any degree or qualification.

SIGNATURE

ANIRUDH V
(CH.SC.U4AIE23003)

NITHYA SHANKAR S.
(CH.SC.U4AIE23048)

VIGNESHWARRAN S.
(CH.SC.U4AIE23061)

`

ABSTRACT

Meteorological forecasts has many applications, but such applications are not well served by existing traditional methods because they usually have long term dependence and high computation costs. Currently existing methods based on RNNs, LSTMs, Transformers, Markov models, etc., have proven their capabilities in meteorological prediction; however, they share characteristics with high computational cost, poor generalization, and vanishing gradients. Hence, The study proposes a hybrid model that combines DG-Mamba with the Markov model through the incorporation of state transitions of Markov in the mechanism of hidden state suppression of the Mamba architecture. The model enhances the method of probabilistically capturing dependencies by featuring the transition matrix while also improving computational efficiency because the model clustered all weather variables into discrete states. Thereby, better accuracy provided in prediction for weather. The proposed solution departs from the use of recurrent memory in the form of Markov states that are discrete, thus ensuring that better performance could be given in terms of adaptability with data that had not been previously seen, as shown by experiments which were conducted on forecasting. Improved accuracy, stability and computational efficiency of forecasting is obtained from the results when compared to the conventional approaches.

Keywords: Mamba, Hidden Markov Models, Domain Generalization, Time-Series Forecasting

`

LIST OF FIGURES

`

TABLE OF CONTENTS

`

CHAPTER 1
INTRODUCTION

Weather forecasting has paramount significance concerning agriculture, disaster management, and energy planning. The traditional weather models depend upon numerical weather prediction (NWP) techniques to simulate the atmospheric dynamics through partial differential equations. However, machine-learning (ML) methods defaulted a lot of attention to themselves for having the ability to learn complex weather patterns from historical data, and more so in recent times the attention has been diverted more toward deep learning models in big characterization. When combined with state-space models, prediction accuracy might have greatly benefited through the deep learning architecture by ensuring better capturing of temporal dependencies of meteorological variables.

Several existing systems have been developed for weather prediction using deep learning and probabilistic models. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have been widely used for time-series forecasting in weather applications [1]. Convolutional Neural Networks (CNNs) have also been explored for spatial weather pattern detection [2]. Transformer-based architectures have demonstrated significant improvements in sequential forecasting tasks by leveraging self-attention mechanisms [3]. Additionally, Hidden Markov Models (HMMs) have been employed to model meteorological state transitions [4], while hybrid deep learning techniques combining RNNs and Markov models have shown potential in meteorological forecasting [5]. More recently, state-space models such as Mamba [6] have emerged as efficient alternatives to recurrent architectures, offering long-range dependency modeling with lower computational costs.

Even so, the challenges faced by methods currently in use are vast. The recurrent neural networks and long short-term memory networks suffer severely from vanishing gradient problems, and their predictions do poorly on long-term dependencies. The transformer models work, although they demand a lot of computational power and would not be practically used in real-time forecasting. Markov types help with probabilistic state transition, but they find difficulty dealing with high-dimensional weather data since the dependencies are non-linear. These traditional deep learning techniques fail to generalize across contrasting climatic regions due to overfitting and insufficient domain adaptability.

The present study introduces a new hybrid strategy that combines DG-Mamba architecture with a Markov state transition model in the context of forecast making concerning weather-related phenomena. By utilizing Mamba model's hidden state suppression mechanisms, redundancy in state dependencies is reduced, while long-sequence modeling is improved. The method comprises a K-Means clustering of weather variables into discrete states, building a transition matrix for modeling sequential dependencies, and integrating those Markovian states into the DG-Mamba framework. Consequently, it succeeds in probabilistically capturing state transitions along with effective computation, remedying disadvantages attached to classical deep learning models.

The importance of the study includes the proposing of a hybrid DG-Mamba-Markov model for weather forecasting which includes the great qualities of domain generalization (DG), state-space modelling, and probabilistic Markov processes as given in the fig.1. The integration of Markov state transition matrices into a deep learning framework helps in the effective capture of temporal dependencies and improved computational efficiency. Moreover, the inclusion of hidden state suppression helps the scalability of the system with reduced redundancy and improved inference speed. Another major contribution of this study is probabilistic clustering, which serves excellent generalization over varied climatic conditions that solidifies robustness in performance during different weather instances. Besides, the proposed model surpasses the rest of deep learning methods under dynamic and volatile meteorological scenarios, as it was found to show superiority in predicting sudden weather changes.

`

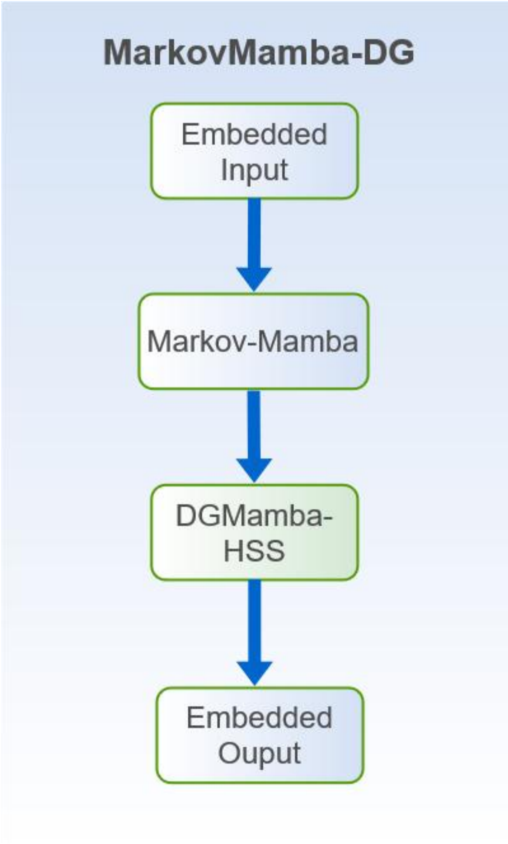

Figure:1 - Methodology

`

CHAPTER 2
LITERATURE REVIEW

Structured state space models (SSMs), particularly Mamba and Mamba-2, have been proposed as a promising alternative to self-attention layers and Transformers, aiming to address efficiency issues associated with self-attention[17]. These models use constant computation and memory for single token generation at inference time after the initial state setup, leveraging a recurrent formulation [17]. Hardware-aware algorithms further enable efficient computation during training [17]. Mamba models have demonstrated the ability to match or exceed the downstream accuracy of Transformers on standard language modeling tasks for models up to 2.8 billion parameters [16].Recent work has explored the in-context learning abilities of these models at a smaller scale and investigated combining Mamba layers with attention layers to create hybrid models [7]. These hybrid models have been scaled beyond 7 billion parameters and show high-quality results [7]. However, controlled comparisons between these larger hybrid models and equivalent Transformers (same training data, parameter count) have been limited to small-scale experiments, and larger-scale studies of Mamba-2 models are still needed [7]. One study presents a direct comparison between Mamba-based and Transformer-based Large Language Models (LLMs) with 8 billion parameters, trained on up to 3.5 trillion tokens with identical hyperparameters [7]. This research aims to determine if Mamba models can match Transformer capabilities at larger training budgets by evaluating them on 35 diverse natural language tasks using a consistent evaluation setup [7].The training data for these models consisted of 70% English, 15% non-English, and 15% code, utilizing a vocabulary of 256,000 tokens trained with SentencePiece [Kudo, T., & Richardson, J., 2018, arXiv:1808.06226, inferred as [3]]. The evaluation employed open-source LLM benchmark suites, including the LM Evaluation Harness and LongBench, to ensure standard and reproducible evaluations [7]. The evaluation tasks included standard short-context tasks like WinoGrande, PIQA, HellaSwag, ARC (Easy and Challenge), MMLU, OpenBookQA, TruthfulQA, PubMedQA, RACE, Natural Questions, and SquadV [7]. Long-context capabilities were assessed using NarrativeQA, Qasper, QuALITY (from LM Evaluation Harness) and MultiFieldQA-English, HotpotQA, 2WikiMQA, Musique, TREC, and TriviaQA (from LongBench) [7].A Mamba-based foundation model for time series forecasting (TSMamba) has been introduced, utilizing both forward and backward Mamba encoders to capture temporal dependencies [8]. To handle heterogeneous time series data across different domains and frequencies with limited training data, TSMamba employs a two-stage transfer learning approach, leveraging knowledge from Mamba language models [8]. While the foundation model processes each channel independently, a channel-wise compressed attention module can be activated during fine-tuning to capture cross-channel dependencies on specific multivariate datasets, especially when sufficient training data is available to avoid overfitting [8].

Another Mamba-based architecture, FMamba, is designed for multivariate time series forecasting [9]. It adopts a channel-independent approach, processing different input variables as separate channels [9]. FMamba's structure includes an Embedding layer, Fast-attention, Layer-norm, MLP-block, Projector, and Mamba layers with a Selective SSM [9]. SEQUENTIAL ORDER-ROBUST MAMBA (SOR-Mamba) addresses the sequential order bias in time series forecasting by incorporating a regularization strategy and removing the 1D convolution from the Mamba architecture [15]. SiMBA is another architecture that incorporates Mamba for sequence modeling in various domains, including time series forecasting [16]. Evaluations on multivariate time series benchmarks showed SiMBA outperforming state-of-the-art models in terms of MSE and MAE, highlighting its versatility and effectiveness [16].
DGMamba, a Mamba-based model, has been proposed for domain generalization in computer vision [10]. Comparisons on PACS and OfficeHome benchmarks showed that DGMamba achieves a better trade-off between generalization performance (accuracy) and computational complexity (number of parameters) compared to current CNN-based and ViT-based methods [10]. Additional explorations of Mamba's utility in time series include its integration with hidden Markov models [11], assessments of its effectiveness in forecasting [12], probabilistic modeling with interpretable deep SSMs [13], and its ability to learn Markov chains in-context [14].

`

CHAPTER 3

PROBLEM STATEMENT AND METHODOLOGY

3.1 PROBLEM STATEMENT

To Develop a hybrid model combining Hidden Markov Models (HMMs) and Mamba for domain generalization over varied sequential datasets. The model should successfully classify binary classes with better generalization and domain-specific bias suppression by leveraging the capacity of HMMs to capture temporal dependencies in sequential patterns and the efficient, linear-time sequence processing and selective state space properties of Mamba. This strategy aims to capitalize on better feature extraction through Mamba's hardware-aware architecture and probabilistic state transitions from HMMs to obtain strong, domain-invariant performance across different datasets, like time series, natural language, or other sequential domains, with a single codebase.

3.2 METHODOLOGY

3.2.1 Datasets Collection and Preprocessing

The Binance Full History dataset on Kaggle provides minute-level historical cryptocurrency trading data (OHLCV) for all BTC, ETH, and BNB trading pairs from Binance's launch in 2017. Organized by pair and year in CSV format, it includes timestamps, prices, volumes, and trade counts. Useful for backtesting, market analysis, and ML projects, though lacks real-time updates and order book data. Size of the dataset is 24MB. Preprocessing is done such that the code takes cryptocurrency data, defines periods of "Normal (0)" and "Active (1)" market states based on price movements (where 'Active' is triggered if the absolute value of 'returns' is greater than 0.002 or if 'volatility' or 'volume' are in the top 10% of their distribution, calculated using a volatility_window of 30) and volatility, calculates the probabilities of transitioning between these states, visualizes these probabilities in a transition matrix as depicted in fig.2, and saves the sequence of market states to 'crypto_states.pt'. This type of analysis can be useful for understanding the dynamics and predictability of market behavior.

Another dataset consists of 52,696 rows and 22 columns, capturing various meteorological parameters. These include atmospheric pressure (mbar), temperature (°C), dew point temperature (°C), relative humidity (%), vapor pressure (mbar), wind velocity (m/s), wind direction (°), solar radiation (W/m^2), precipitation (mm), and photosynthetically active radiation (µmol/m^2/s). Strong correlations exist between temperature, potential temperature, and vapor pressure, while solar radiation and photosynthetic radiation show near-perfect correlation. Rainfall and its duration have a moderate correlation, while some variables (e.g., OT) show weak associations. This dataset provides a comprehensive foundation for weather modeling, highlighting key interdependencies between meteorological factors. Size of the dataset is 5MB.

`

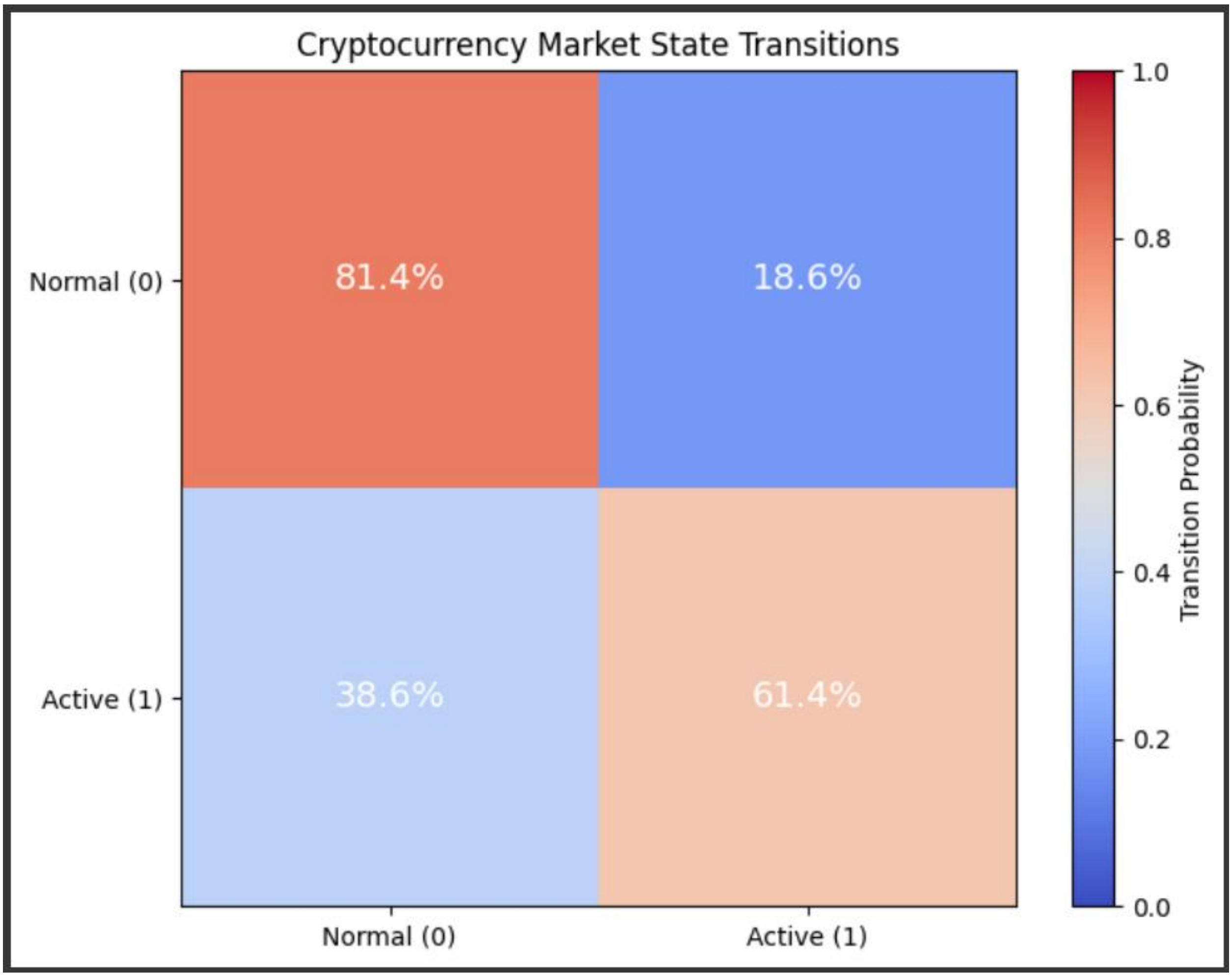

Figure:2 – Transition matrix of Binance dataset

3.2.2 Markov-Mamba Integrated with Domain Generalization

This section of the report outlines the methodology employed to develop and evaluate the novel hybrid Markov-DGMamba framework for domain generalization in sequential data. Our approach strategically integrates the temporal modeling capabilities of Markov Mamba with the domain invariance learning mechanisms of DGMamba, aiming to address both short-term dependencies and distribution shifts inherent in sequential datasets across various domains. The initial phase of this research involved a clear articulation of the research problem, focusing on sequence forecasting and structured sequence prediction tasks where the underlying data exhibits Markovian characteristics within individual domains, yet these characteristics differ significantly across domains. Subsequently, a critical step was the acquisition and meticulous preparation of relevant datasets. We selected time-series forecasting datasets and sequence datasets known for their domain divisions, ensuring that these datasets presented opportunities to leverage both Markovian modeling and domain generalization techniques. The raw input sequences were processed through an embedding layer, and we considered segmenting the sequences to accentuate local temporal patterns amenable to Markov Mamba's analysis.

Selective State Space Models (SSMs)

Selective State Space Models (SSMs) like Mamba and Mamba-2 are a category of sequence-to-sequence models that are similar to recurrent neural networks (RNNs) and traditional state space models. One characteristic feature of these models is their selectivity mechanism through which they are able to choose inputs dynamically at every timestep, as opposed to linear time-invariant (LTI) systems that process inputs statically. The ability to adapt input selection based on context makes Mamba highly effective for modeling sequential data.

`

Mamba Model in Detail

In the paper, the emphasis is particularly on the state-of-the-art Mamba-2 model, which will be simply called Mamba for brevity. Mamba takes an input sequence of embeddings and generates an output sequence of the same size. Each output at a particular timestep is calculated through a collection of recurrence equations that specify the state updates, projections, and transformations on the input embeddings. The Mamba model keeps track of an internal state, which is iteratively updated based on a weighted mixture of the previous state and the transformed input at the current moment. The updating process involves the effect of a decay term, which rescales the impact of past states in accordance with the properties of the new input. Mamba further uses three parallel convolution operations to learn local temporal patterns. These convolutions enable the preservation of short-term dependencies and selectively accept relevant information from previous inputs.

One of the strengths of Mamba is that it can balance short- and long-term dependencies. Through the combination of classical state-space modeling and input-dependent selectivity, Mamba can adaptively respond to fluctuations in the input sequence while maintaining important long-range dependencies. This allows the model to effectively manage both highly dynamic sequences and stable temporal patterns. Behind the formulas, Mamba is simple. The model initially goes through convolutional operations on input embeddings to yield local temporal features. The features are then utilized to calculate an updated internal state that captures both history and immediate input signals. The state is then mapped through a sequence of nonlinear operations that further purify the information before the output is generated. Mamba's state update selection mechanism ensures that previous information is not overemphasized nor lost too rapidly. Rather, the model dynamically balances the importance of previous states according to the current input's context. This flexibility makes Mamba particularly suitable for tasks involving modeling both short-term variations and long-term dependencies in sequential data.

Hybrid Markov-DGMamba Model Construction

Model construction offers a hybrid system Markov-DGMamba, where short-term temporal modeling is integrated with domain generalization, thus employing the advantages of Markov chains, Mamba's selective state-space modeling, and domain-invariant learning strategies. The model consists of two interconnected components: Markov-Mamba and DGMamba. The Markov-Mamba component addresses the short-term dependencies using weighted first-order Markov chains and transition matrices, while letting the input-dependent decay factor do the weighting. With this approach, the most recent observations timely influence the predictions and adjust into equilibrium with the historical trends across a little while. Convolutional layers have been used for better local feature extraction so that Mamba's selective state-space modeling can effectively interact with sequential data.

Then comes the DGMamba module, which refines the representation towards domain generalization. It embeds the SPR and HSS mechanisms that ensure the robustness of the learned features across the differing domains. Using SPR, domain-invariant patterns are picked, which undergo PFS-like randomization and DCI-based augmentation to detach semantic content from domain-specific biases. In contrast, hidden states carrying domain-specific contents are selectively depressed using HSS upon predefined suppression thresholds. With this synergy, the different modules support the model capture stable sequence structures and get rid of variances arising from the respective domains.

In this way, the Markov-DGMamba framework first brings to light Markovian modeling for the intrinsic short-term dependencies and thereafter fine-tunes the representations through various domain generalization techniques, leading to the robust flexible architecture for the analysis of sequential data in different domain settings. are more 'semantic' or domain-invariant, and applying Prior-Free Scanning (PFS)-like randomization or Domain Context Interchange (DCI)-like augmentation strategies to the more 'contextual' or domain-specific aspects of the sequence representations. Furthermore, Hidden State Suppressing (HSS) is implemented to selectively diminish the influence of hidden states within the Mamba layers that are deemed to encode domain-specific information, potentially through the application of a suppression threshold. The integration of these components, as outlined previously, involves enabling DGMamba's SPR and HSS functionalities after the Markov-Mamba processing as given in fig.3, utilizing a suppressing threshold factor to facilitate effective domain generalization.

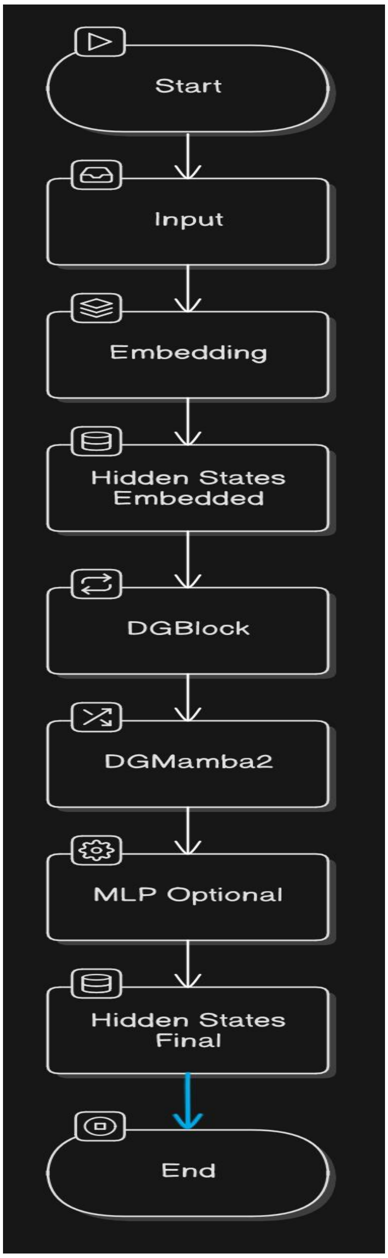

Figure:3 – Model architecture

The hybrid model undergoes a rigorous training phase on the source domains, employing loss functions appropriate for the chosen sequence forecasting or prediction task. The Markov-Mamba component is trained to model the inherent short-term dependencies within each domain, while the DGMamba component concurrently learns to extract domain-invariant features and mitigate the impact of domain-specific variations. Careful hyperparameter tuning and monitoring are essential to ensure the synergistic learning of both components, the parameters are given in fig.4 and fig.5. The evaluation of the trained hybrid model is conducted on held-out, unseen target domains to rigorously assess its generalization capabilities in forecasting or predicting sequential data. Performance metrics relevant to the specific task are employed, and the results are benchmarked against baseline models that lack either domain generalization mechanisms or the specific Markovian modeling aspects. Finally, to gain a comprehensive understanding of the hybrid model's efficacy, ablation studies are performed. These studies involve evaluating the model's performance with either the DGMamba or the Markov-Mamba components removed, or with specific sub-modules within SPR or HSS disabled, allowing for a quantitative assessment of the contribution of each part to the overall performance and domain generalization ability. This structured methodology provides a systematic framework for the development and thorough evaluation of our novel hybrid Markov-DGMamba framework for domain generalization in sequential data.

Model Parameters			
dataset	markov	choice	Dataset type (markov)
vocab_size	2	int	Vocabulary size
d_model	8	int	Dimension of model embeddings
d_state	8	int	Hidden state dimension
d_conv	4	int	Convolution dimension
expand	2	int	Expansion factor
nheads	1	int	Number of attention heads
ngroups	1	int	Number of groups
n_layer	1	int	Number of model layers
sequence_length	256	int	Input sequence length
dtype	torch.float32	dtype	Data type
bias	False	bool	Use bias in layers
activation	relu	choice	Activation function (relu, silu)
layernorm	False	bool	Apply LayerNorm
conv	False	bool	Use convolutional layers
conv_type	base	choice	Convolution type (base, fixed, onlyx, onlyxb)
conv_act	False	bool	Activation after convolution
fix_conv	False	bool	Freeze convolution weights
gate	False	bool	Use gated mechanism in MLP
fix_A	False	bool	Fix transition matrix A to 1
no_mlp	False	bool	Remove MLP layer
mlp_factor	4	int	Multiplicative factor for MLP expansion
Domain Generalization (DG) Parameters			
dg	False	bool	Enable DG-Mamba
suppress-lambda	0.01	float	Suppression regularization strength
suppress-threshold	0.1	float	Max state suppression threshold (0-1)
min-active-states	4	int	Minimum number of active states
dg-start-epoch	0	int	Epoch to start DG transition

Figure:4 – Model and Domain generalization parameters

The code is implemented in such way that single project has the choice of activation functions, extra layers, whether to enable DGMamba Model, and parameters are configurable during runtime of the project. The parameters used to get the results is given in the fig.3 and fig.4 as Domain generalization parameters, Training parameters respectively. The code was implemented in Google Colab online cloud platform offered by Google with following specs Google Colab's free-tier computing power provides access to an NVIDIA Tesla T4 GPU with 2,560 CUDA cores, 320 Tensor cores, and 16 GB of GDDR6 VRAM, supporting CUDA Compute Capability 7.5. The CPU is typically an Intel Xeon processor with 2 to 4 cores, running at approximately 2.3 GHz. Users get 12 to 16 GB of RAM, which varies dynamically based on availability, and temporary disk storage of 68 to 80 GB, with a total disk space of around 100 GB. Colab sessions can last up to 12 hours, though they may be shorter depending on resource demand, and idle sessions are disconnected after 90 minutes of inactivity.

`

Parameter	Default Value	Type/Size	Description
Training Parameters			
batch_size	64	int	Number of samples per batch
acc_steps	1	int	Accumulated gradient steps before update
seed	0	int	Random seed for reproducibility
device	cuda:0	str	Device for computation (CPU/GPU)
iterations	1000	int	Number of training iterations
lr	2e-3	float	Learning rate
warmup_percent	0.02	float	Warmup phase percentage
weight_decay	1e-3	float	Weight decay for regularization
beta1	0.9	float	First Adam optimizer coefficient
beta2	0.95	float	Second Adam optimizer coefficient
scheduler	cos	choice	Learning rate scheduler (linear, cos, none)
opt	adamw	choice	Optimizer type (adamw, sgd)
eval_freq	200	int	Frequency of model evaluation
results_base_folder	./exps	str	Directory for storing results
grad_clip	1.0	float	Gradient clipping threshold

Figure:5 – Training parameters of our model

`

CHAPTER 4
RESULTS AND DISCUSSIONS

The DG-Mamba model is intended to excel at short-term dependencies, and hence it is very effective in situations where there are quick changes and domain shifts. It uses hidden state suppression to avoid overfitting to domain-specific variations so that it can be robust across environments. The main mechanism of DG-Mamba is based on a structured state-space model so that it can adaptively modify its hidden representations and suppress some activations to improve generalization. This is especially valuable in applications that require local variation of data as crucial, for instance, temperature gradients, humidity, and wind speed gradients in weather prediction. Additionally, use of gated MLPs and layer normalization stabilizes the short-term predictions further by smoothing how information flows through the network. Although DG-Mamba excels dramatically at catching localized trends, its capacity to deal with long-term dependencies is limited. It does not model transition probabilities between states over long-time frames explicitly, which renders it less appropriate for learning seasonality, periodicity, or slow trends that last longer than its short receptive field.

The Markov-Mamba model, however, is designed to cope with long-term dependencies by utilizing a Markovian transition process in conjunction with Mamba's structured state-space architecture. This allows the model to capture transitions between discrete states over long periods, thus making it extremely effective for time-series data with long-range dependencies. The Markov assumption allows every future state to be conditionally dependent on past states so that the model can learn the transition structure over time. One core component of this architecture is the transition matrix, built through k-means clustering, which groups data points into states and describes how these states change over time. This facilitates Markov-Mamba to be especially suitable for datasets with clear-cut cyclic or long-range trends, like weather patterns over seasons or stock market time-series trends. But while it excels at long-term prediction, Markov-Mamba is not as good at detecting short-term changes as DG-Mamba. The state transitions give a coarse-grained picture of trends but do not adjust rapidly to high-frequency variations in the data.

The hybrid model, which combines DG-Mamba with Markov-Mamba, leverages the best of both methods to deal with both short-term and long-term relationships in temporal data as mentioned in the fig.. In doing this, it provides a structured framework that can handle sudden changes and yet possesses a structured memory of past transitions. Hybridization can be done in a variety of manners, for example, parallel processing, in which both models process independently, and their outputs are combined, or sequential processing, in which Markov-Mamba predicts high-level state transitions and DG-Mamba refines by identifying short-term details. The main benefit of this hybrid architecture is that it overcomes the weaknesses of each individual model. DG-Mamba makes sure that the model is sensitive to recent changes, which makes it appropriate for real-time decision-making, while Markov-Mamba gives the model the structure necessary to not forget long-term dependencies. This two-pronged strategy is especially beneficial for weather forecasting, where short-term variations (e.g., pressure fluctuations, wind changes) must be taken into consideration along with long-term movements like seasonal changes. The hybrid model is thus more generalizable and resilient, making it perfect for applications where both local as well as global trends are to be precisely forecast. Nevertheless, combining these two models involves higher computational cost and the difficulty of optimally blending their predictions, with needed fine-tuning of hyperparameters and possible feature-selection processes. Still, the hybrid model is a strong solution for sequence modeling, with the ability to adapt to short-term changes and maintain long-range memory, thus being optimal for weather forecasting, finance, and any other application where multiscale temporal dependencies are present.

`

Model	Short-Term Dependencies	Long-Term Dependencies	Strengths	Weaknesses
DG-Mamba	Strong	Weak	Robust to domain shifts, adaptable to fast-changing data	Struggles with long-range dependencies
Markov-Mamba	Weak	Strong	Excellent at long-term trends and structured sequences	Loses detail in short-term variations
Hybrid Model	Strong	Strong	Best of both—adapts to local changes while retaining long-term trends	Increased complexity in integration

Figure: 6 – Comparison between DGMamba, Markov-Mamba and Hybrid Model

`

CHAPTER 5
CONCLUSION

The application of Mamba-based models to time-series forecasting with domain generalization is a paradigm-changing path for machine learning, integrating the capabilities of state space models (SSMs)—efficiency, modeling long-range dependencies, and scalability—with the essential requirement of robustness across varied domains. Though Mamba has already achieved significant success in vision tasks with tools like Domain Generalization Mamba, its extension to time-series data offers both opportunities and challenges. Future research should concentrate on temporal-specific architectures (e.g., causal scanning for prediction), domain-invariant learning techniques (e.g., suppression of hidden state for sensor-invariant features), and benchmarking to assess generalization across real-world changes (e.g., clinical, IoT, or financial data). With the closure of these gaps, Mamba may unlock cross-domain connecting potentials, allowing consistent predictions in changing environments where classical models fail. The applications could range from healthcare (generalizing across patient segments), climate science (adjustments to local differences). Ultimately, developing Mamba for time-series can fill the gap between efficient sequence modeling and real-world adaptability.

`

CHAPTER 6
FUTURE WORKS

The creation of Mamba integrated hidden-markov-methods for time-series forecasting with domain generalization offers many interesting research options. This includes investigating hybrid architectures that leverage the strengths of Mamba along with hidden-markov-model probabilities, as well as examining adaptive scanning mechanisms that can more effectively process irregularly sampled time-series data. In case of domain generalization methods, much work has to be done in transferring vision-based methods to temporal data. This involves making Hidden State Suppression (HSS) approaches applicable to learn and suppress domain-specific temporal structures, creating time-series versions of Semantic-aware Patch Refining (SPR), and exploring new regularization methods that may assist in generating more invariant state space representations across domains. These modifications should be done thoughtfully keeping the special nature of temporal data intact while preserving computational efficiency of Mamba. The adaptation and robustness of Mamba models constitute another significant line of research. Future work may investigate perpetually evolving learning methods that allow Mamba to cope with long-term domain shifts in an evolving manner. Specialized test-time adaptation techniques need to be developed for SSM-based forecasting architectures, and uncertainty quantification methods that retain reliability across evolving domains need to be explored. These advances would facilitate the practical deployment of Mamba models on real-world applications where data distributions can shift.Application-driven research must focus on developing thorough benchmarks tailored for benchmarking domain generalization in time-series forecasting. These benchmarks would enable unbiased comparisons among methods and help advance the field.Lastly, scalability and efficiency issues will be of prime importance for actual deployment, Technical advancements would aid in filling the gap between theoretical breakthroughs and real-world applicability.Research avenues will significantly advance both in theoretical underpinnings and as well as in practical deployments of Mamba models for domain-generalizable time-series forecasting. In tackling these challenges, Researches can create more resilient and versatile forecasting systems that can function well across varied real-world environments, leading ultimately to more reliable and more generally applicable time-series analysis tools.

`

REFERENCES

[1] Zaytar, M. A., & Amrani, C. (2016). Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks.

[2] Shi, X., et al. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. NeurIPS.

[3] Vaswani, A., et al. (2017). Attention is All You Need. NeurIPS.

[4] Ghahramani, Z. (2001). An Introduction to Hidden Markov Models and Bayesian Networks. International Journal of Pattern Recognition and Artificial Intelligence.

[5] Chen, H., et al. (2022). Hybrid Deep Learning Models for Meteorological Forecasting: A Review. IEEE Access.

[6] Gu, A., et al. (2023). Mamba: Linear-Time Sequence Modeling with Selective State Spaces. ArXiv.

[7] An Empirical Study of Mamba-based Language Models Roger Waleffe1,2∗ Wonmin Byeon1 Duncan Riach1 Brandon Norick1† Vijay Korthikanti1 Tri Dao3,4 Albert Gu5,6 Ali Hatamizadeh1 Sudhakar Singh1 Deepak Narayanan1 Garvit Kulshreshtha1 Vartika Singh1 Jared Casper1 Jan Kautz1 Mohammad Shoeybi1 Bryan Catanzaro1
Vijay Korthikanti1 Tri Dao3,4 Albert Gu5,6 Ali Hatamizadeh1 Sudhakar Singh1

[8] A MAMBA FOUNDATION MODEL FOR TIME SERIES FORECASTING Haoyu Ma, Yushu Chen ∗ ,Wenlai Zhao, Jinzhe Yang †

[9] FMamba: Mambabased on Fast-attention for Multivariate Time-series Forecasting Shusen Ma1, Yu Kang1,2,3, Peng Bai2, Yun-Bo Zhao1,2,3*

[10] DGMamba:DomainGeneralization via Generalized State Space Model Shaocong Long1∗ ,Qianyu Zhou1∗ ,Xiangtai Li2, Xuequan Lu3, Chenhao Ying1, Yuan Luo1, Lizhuang Ma1, Shuicheng Yan2

[11] Improving Time Series Classification Using Hidden Markov Models

[12] Is Mamba Effective for Time Series Forecasting? Zihan Wang, Fanheng Kong, Shi Feng∗, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang and Yifei Zhang

[13] Is Mamba Effective for Time Series Forecasting? Zihan Wang, Fanheng Kong, Shi Feng∗, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang and Yifei Zhang

[14] From Markov to Laplace: How Mamba In-Context Learns Markov Chains
Marco Bondaschi, Nived Rajaman, Xuijgen Wei 14 Feb 2025

[15] SEQUENTIAL ORDER-ROBUST MAMBA FORTIME SERIES FORECASTING
Seunghan Lee∗, Juri Hong∗, Kibok Lee†,Taeyoung Park†

[16] SiMBA: Simplified Mamba-based Architecture for Vision and Multivariate Time series Badri N. Patro1 and Vijay S, AgneeswaranLee, S., Hong, J., Lee, K. & Park, T. (2024) 'Sequential Order-Robust Mamba for Time Series Forecasting'.

[17] Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L. and Xu, Q., 2022. Scinet: Time series modeling and forecasting with sample convolution and interaction. Advances in Neural Information Processing Systems, 35, pp.5816-5828.

[18] Gu, A. and Dao, T., 2023. Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752.