A Comprehensive Survey on Mamba: Architectures, Challenges, and Opportunities

Abdus Salam, **Rasel Mahmud**, **and Tohedul Islam**, American International University-Bangladesh
Saddam Mukta, Lappeenranta-Lahti University of Technology
Swakkhar Shatabda, BRAC University

We comprehensively review a preliminary understanding of Mamba and its evolution. We present a taxonomy and provide a detailed architecture description, diving into reviews of existing architecture variants. We also highlight the challenges and open issues confronted by Mamba to inspire future research agendas.

With the advent of deep learning, transformer and attention modules have become prevalent in various applications, including natural language processing (NLP),[1] computer vision,[2] time-series analysis,[3] graph data,[4] and tabular data.[5] Despite the impressive performance and scalability of the transformer and its attention architecture, these models still face potential bottlenecks, requiring high-end GPUs due to their quadratic floating-point operations (FLOPs) and memory demands during both training and inference, which significantly limits their broader application.

In recent times, state-space models (SSMs) have attracted great attention due to decreasing computing costs and outstanding performance in modeling long-range dependencies. The SSM framework was initially proposed in control theory, computational neuroscience, etc. domains to model dynamic systems using state variables. Mamba, an advanced selective SSM, achieved

Digital Object Identifier 10.1109/MC.2025.3571322
Date of current version: 28 July 2025

cutting-edge performance by integrating time-varying parameters into the traditional SSM framework. It also incorporated an innovative, hardware-optimized algorithm for more efficient training and inference. Importantly, the model successfully addressed the challenge of high quadratic computational complexity caused by a self-attention mechanism.[6] The pipeline of the architecture for Mamba's core component (selective SSM) is shown in Figure 1. The architecture receives a sequence of input text and develops a time-varying selection mechanism that adjusts the context matrices based on the input sequence. The innovation enables contextual SSM models to filter out irrelevant information while preserving important details effectively. Precisely, the selection mechanism is defined by configuring the interval Δ, and the matrices B and C, enabling flexibility and dynamic content-aware modeling similar to the attention mechanism in transformer.[7]

Despite several review articles that have examined various aspects of the Mamba architecture, which serves as a core inference engine for capturing long-range dependencies, several areas remain unexplored. This research aims to provide extensive information by thoroughly examining, summarizing, and assessing Mamba's advancements across diverse fields. Our study provides a thorough overview of Mamba, highlights significant milestones, and explains the Mamba core architecture and its variants. Finally, this work identifies open issues, limitations, and challenges, and suggests future research directions for overcoming Mamba's shortcomings. We present a well-organized, in-depth, and reader-friendly analysis of Mamba and its variants, which distinguishes it from previous Mamba literature.

LITERATURE REVIEW

Mamba can be interpreted as a hybrid of recurrent neural networks and convolutional neural networks, highlighting its effective computation through either recurrence or convolution while obtaining linear or near-linear scalability for sequence modeling.[8] The architecture employs selective SSMs to automatically identify and prioritize crucial information within the input data. Unlike the traditional multihead self-attention mechanism, it eliminates the need for explicitly attending to every element.

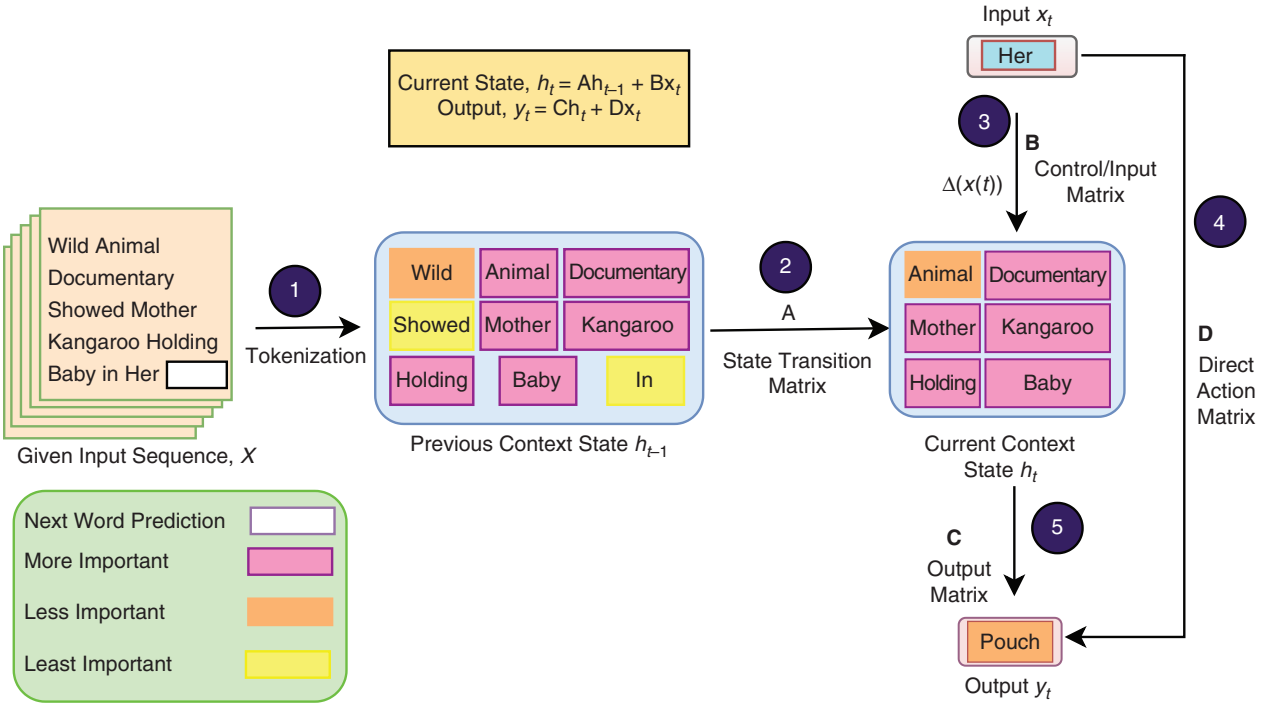

FIGURE 1. Simplified illustration of the overall working process of Mamba's core component (selective SSM) for predicting the next word from a sequence of text input.

Several related works[9,10,11] have investigated various aspects of the Mamba architecture, shedding light on its remarkable advancements and profound contributions to downstream tasks. Liu et al.[9] and Wang et al.[12] each presented a comprehensive review study on Mamba, its variants, and also demonstrated that Mamba is effective for high-resolution remote sensing and medical image processing. The last review[12] illustrated in-depth insights into the dynamic nature of SSM research and its impact in various domains. Later, Yu and Wang[10] explored the ways to evaluate the performance of Mamba in several datasets of visual tasks. Xu et al.[11] highlighted a detailed investigation of several representative visual Mamba backbone networks: pure mamba and hybrid mamba, along with an introductory overview of Mamba and SSMs. Patro and Agneeswaran[13] explored various applications of SSMs and conducted a comparative analysis of their performance. They identified the necessity of optimizing SSMs by addressing challenges related to computational complexity and generalization. Liu and Li[14] conducted a detailed investigation into the generalization of SSMs in sequence modeling, deriving a data-dependent limit to substantiate their findings. They found that the performance of sequence modeling has been significantly enhanced by introducing an initialization scheme and a regularization method.

However, the previous review studies are beset by the following limitations: 1) their taxonomy of mamba variants is incomplete because they ignored several recent architectures, 2) they omitted visual representation of the variants, and 3) they lack in-depth performance comparison. Consequently, a comprehensive yet concise synthesis of critical information on Mamba is necessary for further research. Our article presents an extensive and up-to-date survey of Mamba architecture for practitioners and researchers, which offers a rich tapestry of information on Mamba, delving deeper to completely understand the field and singling out our study from others. Table 1 summarizes existing survey studies related to Mamba architecture.

TABLE 1. Comparison between state-of-the-art research.

Authors	A	B	C	D	E
Liu et al.[9]	X	X	X	✓	Strength of Mamba in vision tasks
Wang et al.[12]	X	X	X	✓	Bibliometric review of SSMs research
Yu and Wang[10]	X	X	X	✓	Analysis of Mamba in vision tasks
Xu et al.[11]	X	X	X	✓	Extensive review of visual Mambas
Patro and Agneeswaran[13]	✓	X	X	✓	Broad review of SSMs research
Liu and Li[14]	X	X	X	X	Assessment of SSMs
Ours	✓	✓	✓	✓	Detailed review of Mamba variants

A: Taxonomy of diverse domains; B: Domain specific discussion; C: Visual representation of Mamba variants; D: Performance of Mamba variants; E: Scope.

METHODOLOGY

This review work concentrates on properly utilizing the preferred reporting items for systematic reviews and meta-analyses (PRISMA) technique[15] to analyze the various aspects of Mamba architecture. We assembled a comprehensive collection of research articles from esteemed academic databases, such as Google Scholar, IEEE Xplore, ScienceDirect, ACM Digital Library, Springer Link, and patents. We enclose a detailed search with the following sample terms: "Mamba AND machine learning OR deep learning OR models", "Variants of Mamba AND machine learning OR deep learning", "Mamba AND natural language processing OR NLP", "Mamba AND time series analysis", and "Mamba AND machine learning OR deep learning OR tools". In the PRISMA procedure, we started with the initial search, resulting in a total of 275 articles. The number of articles was reduced to 207 after excluding repeated, low-quality, and nonpertinent articles. The titles and abstracts were further undertaken to determine their alignment with the study, which resulted in the scaling down of the research, leaving 164 articles. In the final screening process, PRISMA's inclusion and exclusion criteria were applied with a set of conditions to qualify and disqualify articles that narrowed the number of articles to 131. These articles are exploited to write the relevant contents of this article. However, we have selected the 20 references in the bibliography based on pertinence, citation count, latest publication date, and publishers' reputations.

HISTORY OF MAMBA

The development of Mamba can be traced back to the foundational ideas of artificial neural networks introduced

by Warren McCulloch and Walter Pitts in the 1940s. This was followed by the creation of rule-based language models in the 1950s and 1960s, and the advent of statistics-based language models in the 1980s and 1990s. The early 2000s saw the introduction of machine learning frameworks, such as TensorFlow and PyTorch, which revolutionized neural network model development. In 2015, significant advancements in deep learning set the stage for transformer-based models, like BERT and GPT, introduced in 2018, which significantly improved NLP capabilities. In 2021, the Structured State Space for Sequence Modeling (S4) was introduced that leverages linear SSMs and the HiPPO framework for efficiently modeling long sequences. In 2022, research made further advancements in the development of the S4 and S5 models. In 2023, the latest state-of-the-art Mamba (S6) model was launched, offering cutting-edge features, setting a new benchmark for the industry, and making a significant impact in various fields.[8]

TAXONOMY

We have introduced a precise taxonomy of Mamba variants that is reader-friendly, enabling easier understanding and exploring relevant information for scholars. It demonstrates an insightful and logical taxonomy of each category to encompass various task domains. The Mamba taxonomy has been presented according to the downstream application scenarios or tasks, as shown in Figure 2, which we will discuss later in the section on "Mamba Variants."

ARCHITECTURE

Transformers exhibit efficacy in terms of capturing distant relationships and parallelizing computation through the utilization of feedforward neural networks in multilayer perceptron (MLP) blocks, allowing each token to interact explicitly with others in the sequence, but poses significant computational challenges in long inputs.[1] The Mamba architecture combines the hierarchical modeling of H3 block with the gated MLP, specifically designed to manage protracted sequences while maintaining extensive representation by leveraging memory capabilities of the selective SSM. Uniform repetition of Mamba blocks throughout the network layer further advances the consistency and reliability of input processing, simplifying the overall architecture. It uses sigmoid-weighted linear unit (SiLU)/Swish activation function to improve the model's nonlinearity. This design approach enables prioritizing content pertinent information and filtering out less relevant features, positioning Mamba as a powerful alternative to transformers to interpret a prolonged stream.[16] Nevertheless, this advent may inadvertently discard subtle yet important features or misprioritize ambiguous inputs. Figure 3(a) represents the architecture of a Mamba model. In the following sections, we discuss the architectural components in detail, considering x as the input sequence $(x = x_1, x_2, \ldots, x_n)$ and y as the output.

Linear projection (W_{in})

The Mamba model's processing pipeline embarks with a 1D input and is advanced through a linear projection layer.[17] The input is usually represented as a sequence of vectors, with each element characterized by many features. The sequence could be tokens, time series, or any type of sequential data. This layer implements a linear transformation to the vectors, represented by a weight matrix that projects the vectors from their original feature space into a new one and qualifies the data for deeper processing by the subsequent layers.

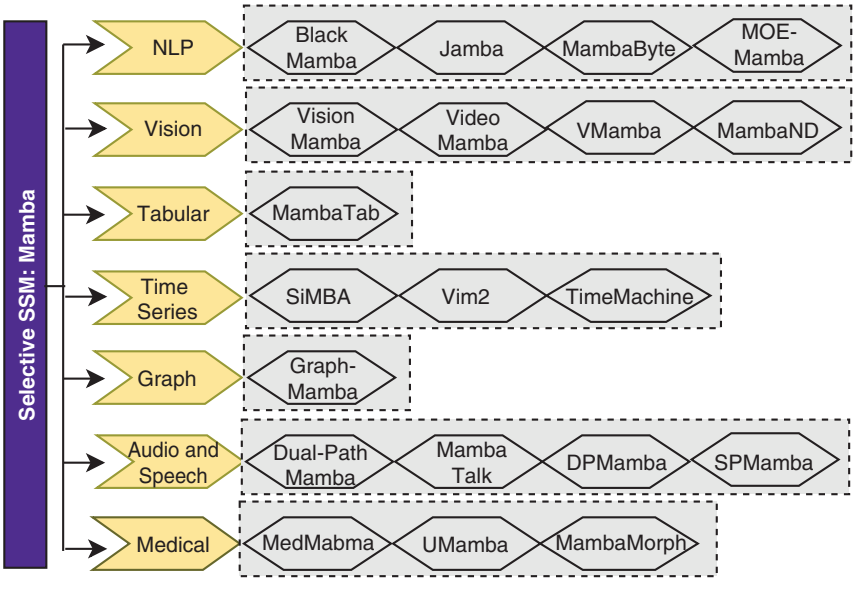

FIGURE 2. Taxonomy of Mamba models.

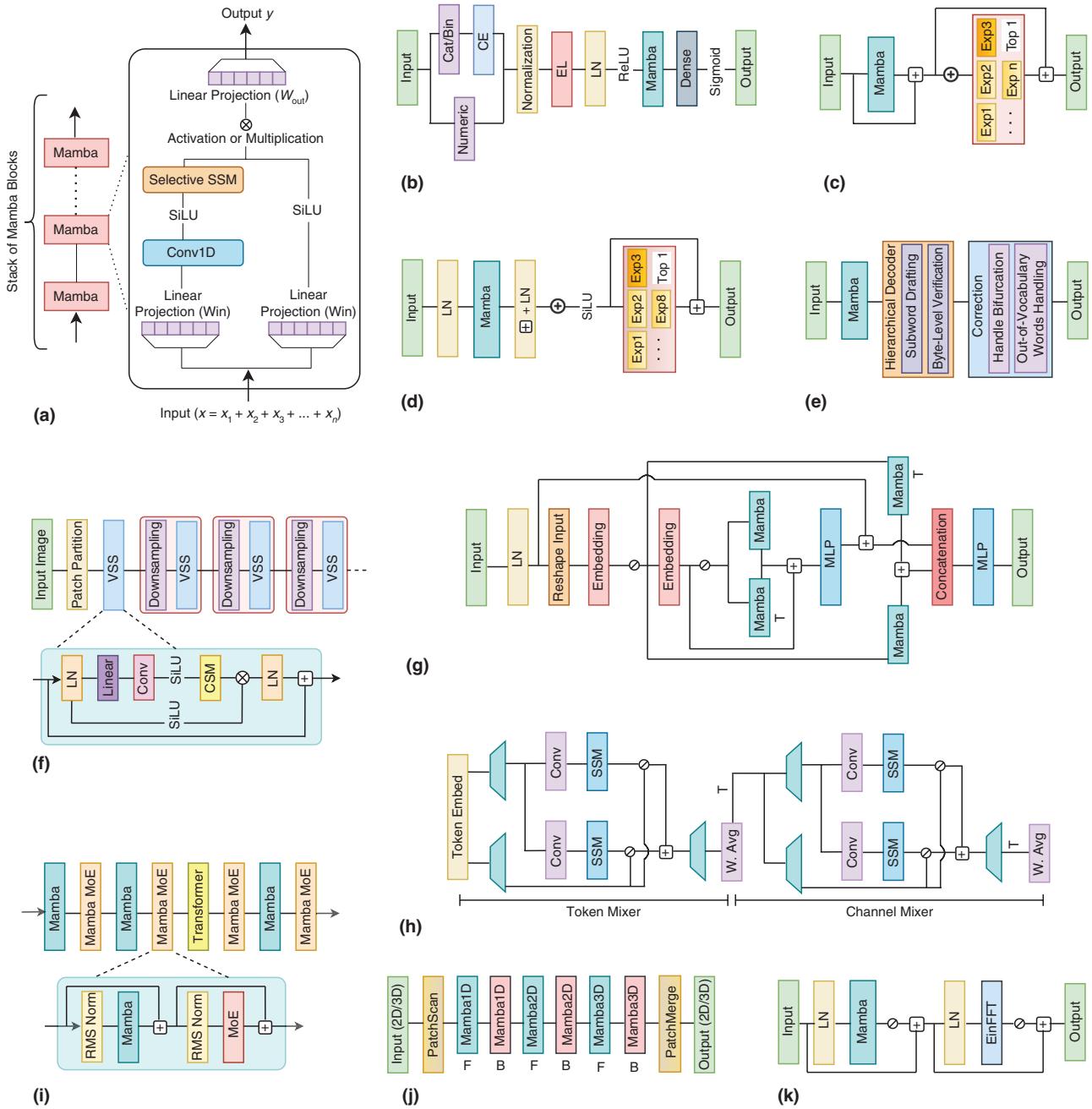

FIGURE 3. (a)–(k) The architectures of Mamba and its variants. B: backward; Bin: binary; Cat: categorical; CE: categorical encoder; CSM: cross-scan module; EL: embedding learning; Exp: expert; F: forward; LN: layer normalization; MoE: mixture of experts; Norm: normalization; T: transpose; VSS: visual state space; W Avg: weighted average; ⊘: dropout; ⊕: routing; MLP: multilayer perceptron; SSM: state space model; EinFFT: einstein fast fourier transform.

Conv1D

The input to Conv1D is a series of vectors from the linear projection (W_{in}). This layer entails sliding convolutional filters all through the sequence to identify correlations or short-term interdependence, capturing local patterns in the data.[16] This procedure yields modified sequences (series of vectors) with more detailed, locally relevant characteristics at each time step (or token).

SiLU activation

SiLU is a pointwise nonlinear activation function that combines linear and sigmoid functions to independently transform each element in the sequence. The nonlinear transformation of SiLU helps the model to capture detailed patterns and learn sophisticated representations. After processing the output of the Conv1D layer, this function sends the resultant data to the selective SSM layer [see (1)]. Simultaneously, SiLU receives data from linear projection (W_{in}), processes it, and sends the outcome to the nonlinearity component [see (2)]. The output of this layer remains a sequence of vectors but with increased representational capability and expressiveness that assist in the later layers

$$a = \text{SiLU}(\text{Conv1D}(\text{Linear}(x))) \quad (1)$$

$$b = \text{SiLU}(\text{Linear}(x)). \quad (2)$$

Selective SSM

SSMs are the foundational basis of the Mamba architecture, which provides a mathematical framework for modeling the evolution of hidden states over time. They use an N-D latent to transform input into output. Mamba integrates a selection mechanism with SSM, referred to as *selective SSM*, that transforms the traditional time-invariant parameters to time-varying.[7] The output of the linear projection (W_{in}) + Conv1D + SiLU pipeline is further processed by Mamba's selective SSM layer to capture longer-range dependencies.[3] The selective SSM processes temporal information using an internal state-space technique, whereas standard approaches find retaining information over long sequences difficult. This selective attention mechanism empowers the model to remember and focus on obvious segments of the sequence, employing a hidden state, improving the model's concentration, and making it proficient at encoding both short- and long-term dependences.

Nonlinearity

This module handles two data streams: one from the linear projection (W_{in}) + SiLU path and the other from the linear projection (W_{in}) + Conv1D + SiLU + selective SSM path. The module mixes local and global information by multiplying these two streams [see (3)]. The output of this module is an extensive and symmetrical representation of the sequence that nonlinearly combines local and long-range dependencies. This mechanism makes the data more appropriate for jobs requiring knowledge of complicated, hierarchical relationships within the sequence

$$c = \text{SelectiveSSM}(a) \otimes b. \quad (3)$$

Linear projection (W_{out})

Linear projection (W_{out}) completes the processing pipeline by generating the final output of the Mamba model [see (4)]. This component maps the nonlinear results from the preceding step into the intended output dimensions using a learned linear transformation much like the original linear projection (W_{in}). The output is usually a sequence of vectors. Invaluable information accomplished from the network's multilayered processing is apprehended in the output sequence

$$y = \text{Linear}(c). \quad (4)$$

MAMBA VARIANTS

In this section, we present variants of Mamba to address varying requirements and tradeoffs in computational efficiency, memory usage, and performance across diverse application domains. MambaTab,[5] Graph Mamba,[4] SiMBA,[18] and TimeMachine[3] models have demonstrated Mamba's effectiveness across tabular, graph, and time series data, respectively. Mamba variants like MambaND,[6] VMamba,[2] MedMamba,[19] and MambaMixer[16] have become effective techniques for computer vision tasks, capturing long-range dependencies across both spatial and temporal dimensions. Similarly, MedMamba[19] incorporates SSMs with convolutional layers, achieving superior performance in medical image processing and paving the way for effectively segmenting and identifying target lesions and other anomalies in this research space. BlackMamba,[7] Jamba,[1] MambaByte,[17] and MoE-Mamba[20] utilize Mamba architecture to process and evaluate sequential text data where the last three models employ the mixture of experts (MoE) framework. Detailed specifications of Mamba variants based on the downstream task scenarios, strength, computational efficiency, scalability, context length, and hardware requirements are outlined in Table 2.

PERFORMANCE EVALUATION

Mamba has attracted widespread attention, demonstrating superior performance with lower memory and computational burden in diverse tasks,

TABLE 2 Summary of existing Mamba variants.

Variant name	Computational efficiency	Context/ sequence length	Scalability	Hardware requirement	Domain application: Strength
MambaND, Figure 3(j)	Multidirectional scanning captures the spatial and temporal relationships. Optimized memory and reduced parameters without sacrificing performance.	2D (H × W), 3D (T × H × W)	Linear complexity, efficient memory usage, and flexible scalability for large-scale tasks.	Nvidia A100 GPU	Computer vision and forecasting: Unravels images across different dimensions.
VMamba, Figure 3(f)	Admirable performance while maintaining lower FLOPs with fewer parameter counts.	224^2, $1{,}024^2$	Cross-scan enforces multidirectional scanning. Capture global context ensuring high adaptability.	Nvidia A100 GPU	Image processing: proposes a four-way 2D selective scan.
MoE-Mamba, Figure 3(c)	Specific token-level patterns for both unconditional and conditional text processing. 2.35× faster training step than Mamba and better log perplexity than transformer.	1,024 tokens	Evenly distribute tokens to most appropriate experts for load balancing.	Nvidia A100 GPU	Language processing: Explores the interleaving Mamba and MoE layers.
Black-Mamba, Figure 3(d)	Optimizes memory and computational costs. Achieves training FLOPs rate 5 and 1.25 times more efficient than Pythia, OPT and Mamba.	2,048 tokens	Minimal memory footprint and reduced FLOPs maintain computational efficiency. Appropriate expert selection mechanism.	Nvidia A100 GPU	Language processing: incorporates Mamba-MoE with Sinkhorn-based expert selection.
Mamba-Tab, Figure 3(b)	Dynamically identifies effective and new features. Lightweight and uniform preprocessing approach also preserves numerical representations.	32 tokens	Comparable performance with fewer parameters depicts high scalability.	Versatile CPUs and GPUs	Tabular data processing: Assesses feature incremental learning in Mamba.
TimeMachine, Figure 3(g)	Integration of four Mamba blocks to learn global and local contexts. Channel mixing and channel independence strategies for accurate forecasting.	96 tokens	Hierarchical scalable modeling approach. Efficient data processing and hyperparameter tuning.	NVIDIA 4XV100 GPU	Forecasting: Single model handles both channel-mixing and independent cases.
Jamba, Figure 3(i)	96.8% and 87.5% reduction in key-value cache memory usage compared to LLAMA-2 and Mistral, Mixtral, with lower memory overhead.	256K tokens	Hybrid attention-Mamba layers. Efficient memory management and mixer of expert layer.	NVIDIA H100 GPU	Language processing: hybrid design combines Transformer-Mamba architectures.
MambaMixer, Figure 3(h)	Selective token, channel mixer, multidirectional scanning and weighted averaging of earlier features. Lower MSE across multiple datasets. Strong performance in both box and mask average precision.	Versatile input length	Superior performance due to their selective channel mixer and hierarchical architectures. Competitive performance despite having fewer parameters.	NVIDIA A100 GPU	Computer vision: Introduces selective token and channel mixer mechanism.
MambaByte, Figure 3(e)	Handle long sequences with 2.6× faster inference than Mamba and 4.6× faster in sequence generation than MegaByte.	8,192 tokens	Maintains efficient computational cost with lower complexity.	A100 80GB PCIe GPU	Language processing: Adopts Mamba for byte-level modeling.
SiMBA, Figure 3(k)	Discretization to transform and process continuous parameters. Ensuring efficiency with lower parameter counts and FLOPs. Lowest mean squared error and mean absolute error compared to other models.		Integrates EinFFT with structured (SSMs) for channel modeling. Varying configurations and architectural modules.	NVIDIA A100 GPU	Image processing and forecasting: Uses EinFFT to manage the instability issues.

such as classification, detection, generation, understanding and many more. We summarize the contrasting outcomes into three major tasks: time series, image processing, and NLP. This research presents the top-performing Mamba models for each task and compares them with their leading contenders for brevity. We have compared the results of the models only if they are evaluated using the same datasets and performance metrics. Tables 3–5 show the comparisons where Mamba variants and their competitors are separated and categorized based on performance metrics, such as top-1 accuracy, mean intersection over union (mIoU), average precision (AP), log perplexity, bits-per-byte (BPB), mean squared error (MSE), and mean-absolute error (MAE) across datasets.

Time series

The performances of three Mamba variants (TimeMachine, SiMBA, and TSMambaMixer) are compared with their top competitors (iTransformer, PatchTST, and SAMFormer). Performance metrics MSE and MAE are evaluated across five datasets (Weather, Traffic, Electricity, ETTh1, and ETTm1) for two token lengths: 96 and 720. Notably, TimeMachine was trained across 100 epochs using Softplus activation.[3] Table 3 shows data for the top two models and three datasets. Mamba models have exhibited superior performance in most cases, revealing their strengths on time series tasks. For token length 720, TSMambaMixer displayed the lowest MSE values 0.449, 0.422, and 0.407 on the Traffic, ETTh1, and ETTm1 datasets, whereas SiMBA presented the lowest MAE value 0.297 on the Traffic dataset. In all other settings, TimeMachine mostly achieved the best performance, showing the lowest MSE and MAE values. Exceptionally, for token length 720 on the Weather dataset, SAMFormer showed the lowest MSE value (0.334), slightly better than TSMambaMixer (0.337). Similarly, iTransformer and TSMambaMixer achieved close MSE values of 0.395 and 0.396 for token length 96.

Computer vision

The experimental results in Table 4 are displayed for three computer vision tasks on separate datasets: image classification on ImageNet-1K, image segmentation on ADE20K, and object detection on COCO. Vision Mamba models (for example, VMamba, MambaMixer, and SiMBA) were trained on ImageNet-1K by leveraging activation functions, such as SiLU, Softplus, and a custom EinFFT, with training epochs of 300–310. For ImageNet-1K, Mamba variants achieved competitive accuracy. However, the models required lower FLOPs, indicating their computational efficiency.[9] The mIoU scores of the Mamba models illustrated their outstanding performance over their competitors on image segmentation tasks. The metrics AP^b and AP^m measure the object detection and instance segmentation performance. AP^b and AP^m values of VMamba on different settings specify that it outperforms others in the object detection task on the COCO dataset when trained for the longer schedule (3× or 36 epochs). In contrast, SiMBA achieved incredible results with fewer parameters.

Natural language processing

The outstanding performance of Mamba models in NLP tasks are demonstrated in Table 5. The lower log perplexity value of MoE-Mamba shows its dominance. Smaller BPB values for all datasets mean MambaByte performed better than MegaByte even with only 353 M parameters. In the reasoning tasks, Jamba and BlackMamba

TABLE 3. Mamba performance comparison for time-series forecasting.

	Long-term forecasting						
	Datasets →	Weather		Electricity		Traffic	
Models	Prediction length ↓	MSE	MAE	MSE	MAE	MSE	MAE
TimeMachine[3]	96	0.164	0.208	0.142	0.236	0.397	0.268
	720	0.342	0.343	0.207	0.298	0.467	0.3
SiMBA[18]	96	0.176	0.219	0.165	0.253	0.468	0.268
	720	0.350	0.349	0.214	0.305	0.564	0.297
iTransformer[3]	96	0.174	0.214	0.148	0.240	0.395	0.268
	720	0.358	0.349	0.225	0.317	0.467	0.302
PatchTST[18]	96	0.171	0.23	0.159	0.268	0.583	0.319
	720	0.365	0.367	0.215	0.317	0.601	0.341

achieved competitive scores in most datasets by employing SwiGLU activation, illustrating the efficiency of Mamba models.[1,7] However, Jamba's lower scores for ARC-E (ARC-Easy), ARC-C (ARC-Challenge), and NQ (Natural Question) datasets indicate that further focus is needed to produce better results in reasoning tasks. The results also show that Mamba models obtained these excellent performances using a competitive number of parameters.

DISCUSSION

The Mamba model has been widely explored and utilized in diverse applications; however, research in this domain is still in its preliminary stages. To help readers understand the latest advancements, this article highlights several intriguing research avenues worthy of attention.

Challenges and open issues

- *Scalability issues:* While Mamba offers linear complexity, managing large-scale genomic, chemical, and video data, it is still an ongoing issue to efficiently handle increasing data sizes without compromising performance. The stability configurations of Mamba are ambiguous when scaling to large networks, affecting performance and optimization of the model for large datasets.[11,18,16]
- *Efficiency tradeoffs of Mamba versus transformer:* The GPU utilization for Mamba does not consistently show a reduction compared to the transformer model. The bidirectional scanning method and multiple scanning directions reduce the model performance and limit the benefit of Mamba's linear complexity.
- *Memory constraints:* Despite Mamba's inherent efficiency in processing lengthy sequences, memory constraints restrict the

TABLE 4. Mamba performance comparison for image processing.

Models	Image size	Parameters	FLOPs						
Image classification on ImageNet-1K				Top-1 Accuracy					
EffNet-B6[18]	528^2	43 M	19.0G	84					
SVT-H-L[18]	224^2	54.0 M	12.7G	85.7					
VMamba-B[2]	224^2	89 M	15.4G	83.9					
SiMBA-L (EinFFT)[18]	224^2	36.6 M	7.6G	84.4					
Image segmentation on ADE20K				mIoU (single-scale)			mIoU (multiscale)		
ConvNeXt-S[18]	512^2	82 M	1,027G	48.7			49.6		
ConvNeXt-B[2]	512^2	122 M	1,170G	49.1			49.9		
SiMBA-S[18]	512^2	62 M	1,040G	49			49.6		
VMamba-B[2]	512^2	122 M	1,170G	51			51.6		
Object detection on COCO				AP^b	AP^b50	AP^b75	AP^m	AP^m50	AP^m75
PVTv2-B3[18]		65 M	397G	47	68.1	51.7	42.5	65.7	45.7
ConvNeXt-S[2]		70 M	348G	47.9	70	52.7	42.9	66.9	46.2
SiMBA-S (1× SS)[18]		60 M	382G	46.9	68.6	51.7	42.6	65.9	45.8
VMamba-S (3× MS)[2]		70 M	384G	49.9	70.9	54.7	44.2	68.2	47.7

1× SS: 1× (12 epochs) with single-scale; 3× MS: 3× (36 epochs) with multiscale.

sequence length that the model can manage.

- *Challenges in Mamba deployment:* Mamba has shown remarkable results in various domains of multimodal applications. However, training and deploying Mamba models requires substantial computational resources, hindering widespread adoption.
- *Lack of robustness:* Ability to generalization across diverse domains, Mamba models yet to evolve without requiring extensive retraining remains an open research area.[9]
- *Optimization for real-time data:* Model optimization is an ongoing research focus for applying Mamba to domains that require real-time analysis, such as video processing.
- *Spatial loss in scanning:* When Mamba's selective scanning is applied to 2D data, it loses spatial information. To overcome this issue, different methods, such as spatial-first and temporal-first scanning, expose image patches from various directions, allowing spatial data to be integrated across multiple dimensions.[6]
- *Lack of interpretability:* The black-box nature of Mamba makes it hard to interpret the decision-making process. Several studies have presented empirical insights to explain the mechanisms behind the NLP and Vision Mamba models, but their internal representations and decision-making process remain challenging to interpret.[9]
- *Limited datasets and benchmarks:* Mamba models typically require large, high-quality datasets for training. The models struggle in domains where such datasets are scarce. Moreover, the lack of standardized benchmarks makes comparing the performance against other state-of-the-art models difficult.[7,9]
- *Ethical issues in the model:* A critical ethical concern in Mamba

TABLE 5. Mamba performance comparison for NLP tasks.

Models	Datasets and evaluation metrics							
	Log perplexity							
	Datasets →	Colossal clean crawl corpus						
Transformer-MoE$_{100M}$[20]		2.88						
MoE-Mamba$_{100M}$[20]		2.81						
	Bits-per-byte							
	Datasets →	PG19	Stories	Books	ArXiv	Code		
MegaByte$_{758M+262M}$[17]		1	0.978	1.007	0.678	0.411		
MambaByte$_{353M}$[17]		0.93	0.908	0.966	0.663	0.396		
	Reasoning							
	Datasets →	HellaSwag	WinoGrande	ARC-E	ARC-C	PIQA	NQ	TQA
Pythia$_{2.8B}$[7]		45.1	61.2	62.9	28.8	73.7		
Llama-2$_{70B}$[1]		85.3	80.2	80.2	67.3	82.8	46.9	44.9
BlackMamba$_{2.8B}$[7]		39.7	52.1	60.3	24.5	71.2		
Jamba[1]		87.1	82.5	73.5	64.4	83.2	45.9	46.4

ARC-C: ARC-Challenge; ARC-E: ARC-Easy; NQ: natural questions; TQA: TruthfulQA; PIQA: Physical Interaction: Question Answering.

models is that they do not mitigate the biases in training data, particularly in sensitive domains like health care and genomics. Handling such data in Mamba models requires robust privacy and security measures to prevent misuse and ensure compliance with regulations.[1,7]

result, the model cannot capture the relationships between feature maps, which restricts its capacity to capture global information in multidimensional datasets, such as multivariate time series and images.

- *Model vulnerability:* Models like VMamba are prone to white-box attacks, in which the attacker is fully aware of the model's parameters and design.[11] The experiment revealed that some parameters are vulnerable and perturbing them can significantly affect the model, making it more susceptible to adversarial attacks.

WE OBSERVED THAT MAMBA MODELS DEMONSTRATED OUTSTANDING RESULTS IN VARIOUS APPLICATION DOMAINS, SUCH AS NLP, COMPUTER VISION, TIME-VARYING PREDICTION, AND MANY MORE.

Limitations

- *Attention prioritization:* Mamba's focused attention may prioritize certain input components over others, potentially overlooking crucial contextual information and hindering overall task performance.[9]
- *Model overfitting:* Mamba's hidden states can accumulate domain-specific data, which increases the chance of overfitting and decreases the model's capacity for generalization.[9] The requirement to handle domain-agnostic information is frequently but not sufficiently addressed by Mamba's current scanning algorithms.
- *Difficulty in understanding global context:* Mamba does not consider interactions between several channels because it independently applies the selective SSM to each channel.[2] As a

This research presented an in-depth review of Mamba and its variants architecture, performance, challenges, open issues, and limitations. We observed that Mamba models demonstrated outstanding results in various application domains, such as NLP, computer vision, time-varying prediction, and many more. Nevertheless, Mamba is in its initial development stage and has significant potential for improvement compared to a more refined transformer-based approach. We outlined various potential challenges and offered potential research to drive further progress as follows.

For data efficiency and utilization:

- To improve the generalization capabilities of Mamba models, methods like data augmentation, transfer learning, and few-shot learning can be applied to limited data scenarios.
- Applying appropriate optimization techniques to the Mamba model is essential for avoiding computational overhead, specifically while handling high-resolution data.
- Mamba can be improved by developing robust frameworks, like cross-modal transfer learning and joint representations, to understand and integrate complex and multimodal data.[9]

For algorithmic enhancements:

- *Efficient scanning mechanism:* Adapting scanning techniques similar to grid scanning and hierarchical scanning in the Mamba model are required to effectively capture the spatial and temporal information in 2D or 3D visual data. The scanning mechanism may minimize the spurious associations and retain important image structure information.
- *Hybridization with transformer architecture:* To enhance Mamba's performance, attention mechanisms, hierarchical fusion techniques, and feature integration methods from architectures like transformer can be effective. Mamba's sparse architecture and lack of attention mechanisms led to limited token interactions, restricting its capability to capture comprehensive details.

Although attention mechanisms consider all tokens equally during scanning, Mamba focuses only on local scan. Hybridizing Mamba with transformers presents a potential solution to address these limitations.
- *Algorithmic optimization:* Optimization of hardware-friendly algorithms with exploratory methods in Mamba models can produce optimal performance while considering the computational overhead.

For interpretability and explainability:

- *Enhancing model interpretability:* Improving Mamba's interpretability by including attention mechanisms and visualizing state transitions can provide insights into the model's decision-making process. Interpretability can be enhanced through techniques like salience mapping, modular architecture, and diagnostic evaluation of input/output mappings based on feature importance.
- *Model explainability:* Incorporating explainable artificial intelligence (XAI) techniques, like developing tools for interactive exploration of model behavior, is necessary to enhance the transparency and trustworthiness of Mamba. Developing XAI in Mamba requires careful consideration, such as scalability, domain specific constraints, and tradeoff between performance and efficiency.

For ethical and societal implications:

- Efficient identification, quantification, and mitigation of biases in the training data and Mamba models' predictions are essential for developing fair and ethical applications.
- *Differential Mamba models:* Similar to transformer-based differential models, Mamba can be fine-tuned by emphasizing specific layers determined from their gradient contributions and applying regularization through optimized gradient dynamics. However, these approaches may increase computational overhead, make gradient balancing difficult, demand significant resources, and reduce scalability. Despite these limitations, differential learning techniques can improve user data protection and strengthen trust through robust, privacy-preserving methods.

For expanding application domains:

- *Integration with other approaches:* Mamba models should make necessary advancements to analyze complex time series, long-term dependencies, and multimodal data, allowing models to make accurate predictions and recommendations in various domains, including medical and finance.
- *Utilization in decision support system:* Mamba models can be utilized efficiently in decision-making tasks by refining their capabilities, such as autonomous navigation, robotic process automation, customizable prediction, and resource efficiency by integrating multiple sensory inputs and managing temporal dependencies.
- *Incorporating Mamba in large language models:* Mamba could replace transformers in large language models by utilizing its selective SSM architecture to handle sequential data and minimize attention overhead. However, this approach may require theoretical advancements to effectively address the challenges of capturing long-range dependencies.

For benchmarking and standardization:

- *Standard metrics and datasets:* Establishing standardized metrics and datasets for evaluating performance against other state-of-the-art methods can help to increase trust in Mamba models.[12]
- *Hyperparameter tuning and optimization:* Creating guidelines for Mamba's architectural design, hyperparameter tuning, and performance evaluation is necessary to standardize the training, evaluation, and deployment.

REFERENCES

1. O. Lieber et al., "Jamba: A hybrid transformer-Mamba language model," 2024, *arXiv:2403.19887*.
2. Y. Liu et al., "VMamba: Visual state space model," *Adv. Neural Inform. Process. Syst.*, vol. 37, pp. 103,031–103,063, Dec. 2024.
3. M. A. Ahamed and Q. Cheng, "TimeMachine: A time series is worth 4 Mambas for long-term forecasting," in *Proc. 27th Eur. Conf. Artif. Intell. (ECAI)*, 2024, vol. 392, pp. 1688–1695.
4. C. Wang, O. Tsepa, J. Ma, and B. Wang, "Graph-Mamba: Towards long-range graph sequence modeling with selective state spaces," 2024, *arXiv:2402.00789*.
5. M. A. Ahamed and Q. Cheng, "MambaTab: A plug-and-play model for

ABOUT THE AUTHORS

ABDUS SALAM is an associate professor of the Computer Science Department at the American International University-Bangladesh, Dhaka 1229, Bangladesh. His research interests include artificial intelligence, machine learning, and data science. Salam received his Ph.D. in explainable artificial intelligence from Macquarie University, Australia. Contact him at abdus.salam@aiub.edu.

RASEL MAHMUD is pursuing a Master's degree from American International University-Bangladesh, Dhaka 1229, Bangladesh. His research interests include deep learning, computer vision, and natural language processing. Mahmud received his B.S. in computer science and engineering from American International University-Bangladesh. Contact him at 20-43867-2@student.aiub.edu.

TOHEDUL ISLAM is an assistant professor in the Computer Science Department at the American International University-Bangladesh, Dhaka 1229, Bangladesh. His research interests include machine learning, data mining, and data science. Islam received his M.Sc. in computer science from the University of Trento, Italy. Contact him at islamtohedul@aiub.edu.

SADDAM MUKTA is a postdoctoral researcher with the Software Engineering Department of Lappeenranta-Lahti University of Technology, 53850 Lappeenranta, Finland. His research interests include natural language processing, social computing, and machine learning. Mukta received his Ph.D. in artificial intelligence from Bangladesh University of Engineering and Technology, Bangladesh. Contact him at saddam.mukta@lut.fi.

SWAKKHAR SHATABDA is a professor in the Computer Science and Engineering Department at BRAC University, Dhaka 1212, Bangladesh. His research interests include optimization, machine learning, and computational biology. Shatabda received his Ph.D. from the Institute for Integrated and Intelligent Systems, Griffith University. Contact him at swakkhar.shatabda@bracu.ac.bd.

learning tabular data," in *Proc. IEEE 7th Int. Conf. Multimedia Inf. Process. Retrieval (MIPR)*, Piscataway, NJ, USA: IEEE Press, 2024, pp. 369–375.

6. S. Li, H. Singh, and A. Grover, "Mamba-ND: Selective state space modeling for multi-dimensional data," in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland: Springer-Verlag, 2025, pp. 75–92.
7. Q. Anthony, Y. Tokpanov, P. Glorioso, and B. Millidge, "BlackMamba: Mixture of experts for state-space models," 2024, *arXiv:2402.01771*.
8. A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv:2312.00752*, 2024.
9. X. Liu, C. Zhang, and L. Zhang, "Vision mamba: A comprehensive survey and taxonomy," 2024, *arXiv:2405.04404*.
10. W. Yuand and X. Wang, "MambaOut: Do we really need Mamba for vision?," 2024, *arXiv:2405.07992*.
11. R. Xu, S. Yang, Y. Wang, B. Du, and H. Chen, "A survey on vision Mamba: Models, applications and challenges," 2024, *arXiv:2404.18861*.
12. X. Wang et al., "State space model for new-generation network alternative to transformers: A survey," 2024, *arXiv:2404.09516*.
13. B. N. Patro and V. S. Agneeswaran, "Mamba-360: Survey of state space models as transformer alternative for long sequence modelling: Methods, applications, and challenges," 2024, *arXiv:2404.16112*.
14. F. Liu and Q. Li, "From generalization analysis to optimization designs for state space models," in *Proc. 41st Int. Conf. Mach. Learn.*, PMLR, 2024, vol. 235, pp. 31,383–31,405.
15. M. J. Page et al., "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *BMJ*, vol. 372, Mar. 2021, Art no. n71.
16. A. Behrouz, M. Santacatterina, and R. Zabih, "MambaMixer: Efficient selective state space models with dual token and channel selection," 2024, *arXiv:2403.19888*.
17 J. Wang, T. Gangavarapu, J. N. Yan, and A. M. Rush, "MambaByte: Token-free selective state space model," 2024, *arXiv:2401.13660*.
18. B. N. Patro and V. S. Agneeswaran, "SiMBA: Simplified Mamba-based architecture for vision and multivariate time series," 2024, *arXiv:2403.15360*.
19. Y. Yue and Z. Li, "MedMamba: Vision Mamba for medical image classification," 2024, *arXiv:2403.03849*.
20. M. Pióro et al., "MoE-Mamba: Efficient selective state space models with mixture of experts," 2024, *arXiv:2401.04081*.