Software Requirements Specification

BHASA

Bayesian Hyperdimensional Adaptive Sequence Architecture

Version 1.0

Confidential Document

February 5, 2025

1 Executive Summary

BHASA will be India's first Large Language Model (LLM), combining the MAMBA architecture with Bayesian principles and hyperdimensional computing. With 20M parameters, it will deliver enterprise-grade performance while minimizing computational overhead. BHASA aims to position India as a leader in AI, following the success of DeepSeek and ChatGPT globally.

1.1 Maybe Attention Isn't All You Need

The decision to choose MAMBA over Transformers or DeepSeek's R1 for BHASA is based on several key advantages backed by research. Transformers, despite their success, are computationally expensive due to the quadratic scaling of the attention mechanism (Vaswani et al., 2017). While R1 offers some improvements, it still retains the inefficiencies of Transformers. In contrast, MAMBA utilizes state space modeling, which captures long-term dependencies more efficiently, reducing complexity and improving scalability (Cheng et al., 2021). Additionally, MAMBA integrates hyperdimensional computing, offering a robust framework for handling high-dimensional data efficiently, which is not inherently supported by Transformers (Feng et al., 2020). Finally, the incorporation of Bayesian principles allows BHASA to handle uncertainty in predictions, something that traditional transformer models lack (MacKay, 2003). Empirical evidence supports that state space models and hyperdimensional computing can outperform Transformers on resource-limited tasks, making MAMBA the ideal choice for building BHASA's efficient, scalable, and robust architecture.

2 Project Team and Responsibilities

Core Development Team	
Vishesh Yadav	• Architecture Implementation Lead • Research & Innovation Lead • Back-end Computing Infrastructure • Mamba SSM Integration • Performance Optimization • Model Training Pipeline
Ayush Soni	• Dataset Engineering Lead • Data Storage Architecture • ETL Pipeline Development • Expertize Dataset Management • Storage Optimization
Tushar Rana	• ML Operations Lead • Deployment Architecture • CI/CD Pipeline Setup • Monitoring Systems
Biswajit Mohapatra	• Model Architecture Design • Bayesian Implementation • Technical Documentation • Research Publications • Performance Benchmarking

3 Team Communication Structure

Daily Standups	• 9:30 AM IST • Progress updates • Blocker resolution • Task allocation
Weekly Reviews	• Architecture review (VY + BM) • Data pipeline review (AS) • Infrastructure review (TR) • Research sync (All members)
Documentation	• GitHub Wiki • Technical Documentation • Research Notes • Implementation Guides

4 Team Deliverables Timeline

Team Member	Key Deliverables	Timeline
Vishesh Yadav	• Mamba Core Implementation • Training Infrastructure Setup • Model Optimization Framework	Weeks 1-12
Ayush Soni	• Dataset Creation Pipeline • Storage Architecture • Data Processing Systems	Weeks 1-8
Tushar Rana	• Deployment Architecture • Monitoring Setup • Performance Testing Framework	Weeks 6-16
Biswajit Mohapatra	• Architecture Design Document • Research Publications • Innovation Framework	Weeks 1-20

5 Technical Architecture

Core Architecture Components	
Base Architecture	Mamba SSM with selective state retention
Parameter Distribution	• Token Embedding: 3M params • SSM Layers: 15M params • Output Head: 2M params
Architectural Innovations	• Bayesian State Updates • Hyperdimensional Vector Quantization • Adaptive Sequence Compression • Selective Memory Retention

6 Model Specifications

Model Parameters	
Total Parameters	20 million (19.8M trainable)
Layer Configuration	• 12 Mamba blocks • 768 hidden dimensions • 16 SSM states per block • 4 attention heads
Context Window	2048 tokens (expandable to 4096)
Vocabulary	32,000 tokens (SentencePiece tokenizer)
Model Dimension	$d_{\text{model}} = 768$, $d_{\text{state}} = 16$, $d_{\text{conv}} = 4$

7 Training Infrastructure

Component	Specification	Cost (INR)
Primary Computing	2x RTX 4060 8GB (Training)	140,000
Secondary Computing	1x RTX 3060 (Testing)	35,000
Development System	• AMD Ryzen 9 5900X • 64GB DDR4 RAM • 2TB NVMe SSD	125,000
Cloud Resources	• AWS S3 Storage (2TB) • MongoDB Atlas • GitHub Enterprise	45,000
Development Tools	• PyTorch Enterprise • Weights & Biases Pro • VS Code Enterprise • Docker Enterprise	35,000
Infrastructure	• UPS System • Networking Equipment • Cooling Solution	40,000
Miscellaneous	• Testing Resources • Documentation Tools • Monitoring Systems	30,000
	Total	**450,000**

8 Development Phases

Phase	Activities	Duration
Data Engineering	• Data Collection • Cleaning & Preprocessing • Validation Set Creation • Tokenizer Training	4 weeks
Architecture Development	• Mamba Implementation • Bayesian Layer Design • HD Computing Integration • Architecture Testing	6 weeks
Training Pipeline	• Pre-training Setup • Optimization Strategy • Monitoring Systems • Checkpointing Logic	4 weeks
Model Training	• Initial Training • Hyperparameter Tuning • Performance Analysis • Model Pruning	8 weeks
Deployment	• Model Optimization • API Development • Documentation • Performance Testing	4 weeks

9 Performance Metrics

Metric	Description	Target
Perplexity	Language modeling quality	< 12
Inference Speed	Token generation latency	< 30ms/token
Memory Footprint	Runtime memory usage	< 1.5GB
ROUGE-L	Text generation quality	> 0.40
BLEU	Translation accuracy	> 0.35
Response Quality	Human evaluation score	> 4.0/5.0

10 Technical Innovations

- **Bayesian State Updates:**
  - Uncertainty-aware state transitions
  - Probabilistic sequence modeling
  - Adaptive learning rates
- **Hyperdimensional Computing:**
  - Vector symbolic architectures
  - Distributed representations
  - Robust pattern recognition

- **Adaptive Sequence Processing:**
  - Dynamic context window
  - Memory-efficient processing
  - Selective attention mechanism

11 Risk Management

Risk	Impact	Probability	Mitigation
Training Divergence	High	Medium	• Gradient monitoring • Early stopping • Learning rate scheduling
Memory Constraints	High	Low	• Gradient accumulation • Mixed precision training • Model sharding
Performance Issues	Medium	Medium	• Regular profiling • Optimization sprints • Hardware upgrades

12 Deployment Strategy

Deployment Specifications	
Container Solution	Docker with CUDA support
API Framework	FastAPI with async support
Serving Infrastructure	• Kubernetes cluster • Load balancing • Auto-scaling
Monitoring	Prometheus + Grafana