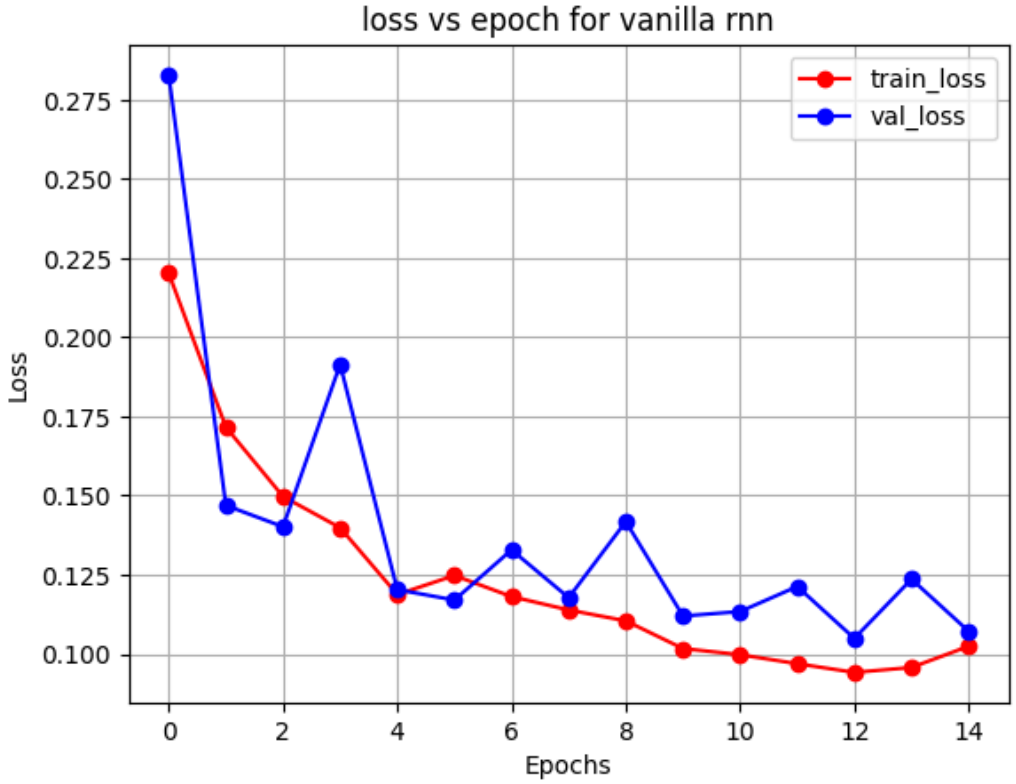
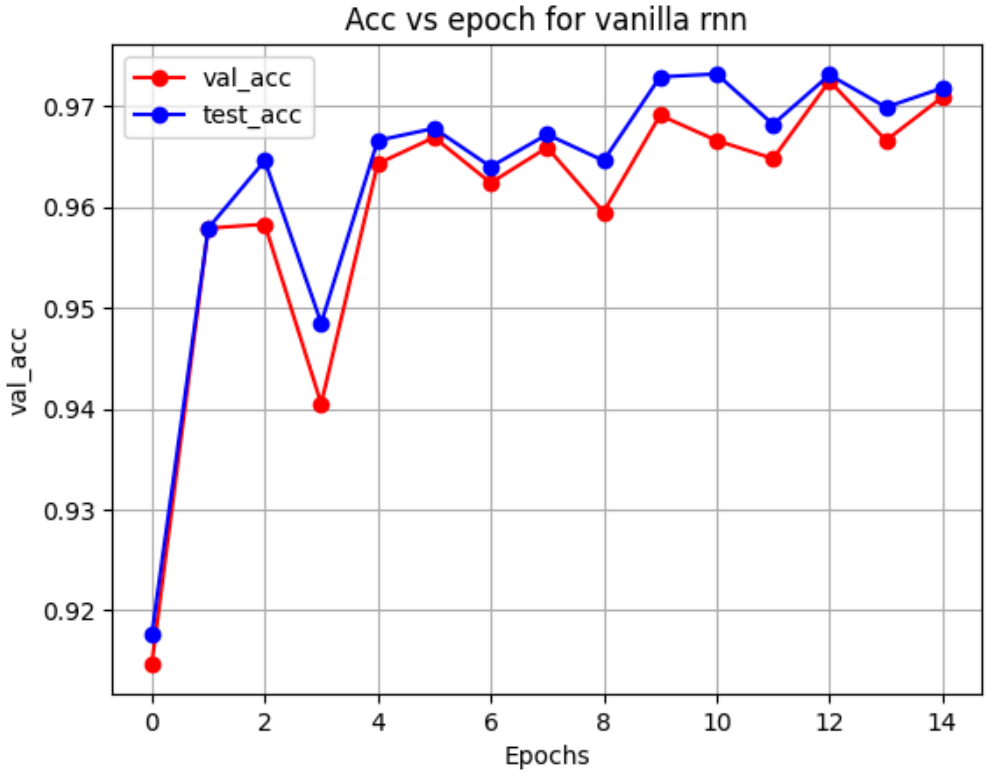
HARISH K N
EP22B026

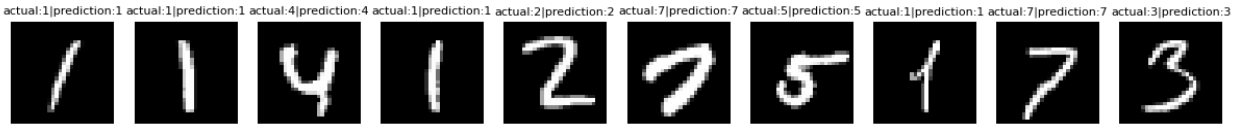
1. MNIST classification using RNN

- Vanilla RNN-(Best accuracy-97.32%)

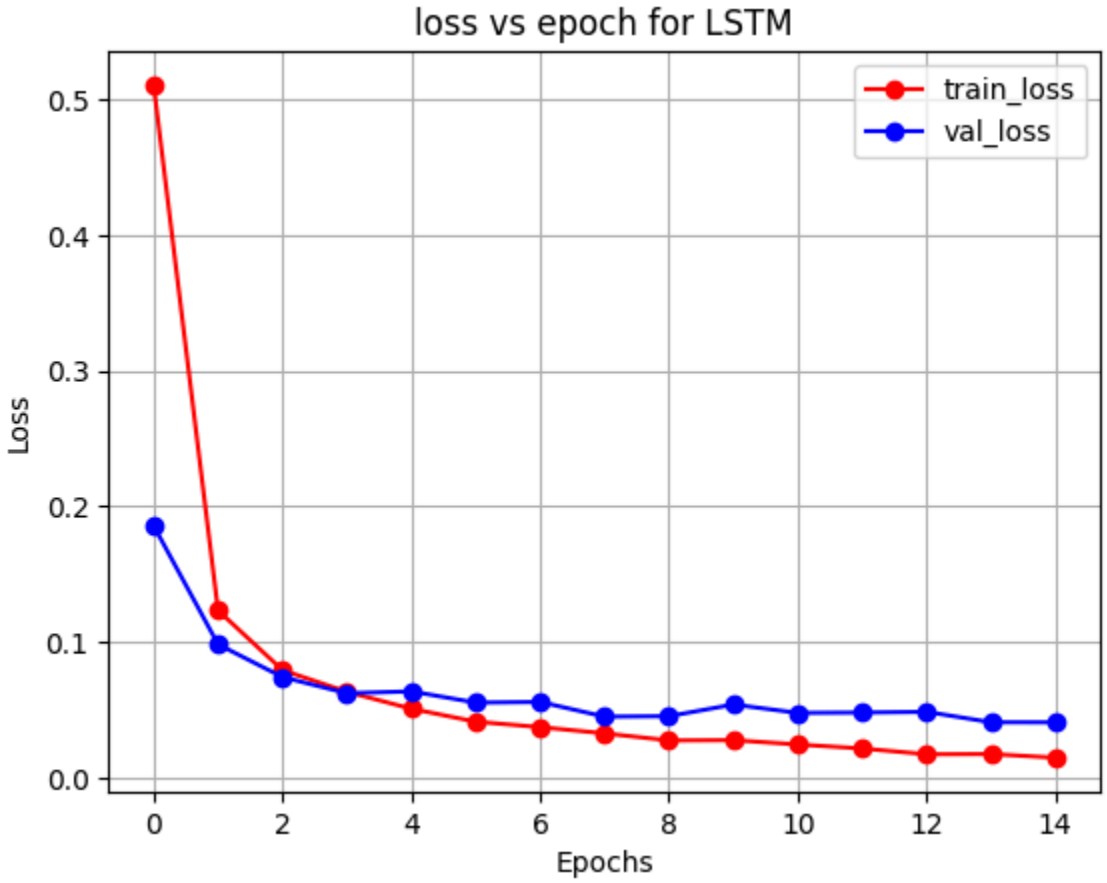




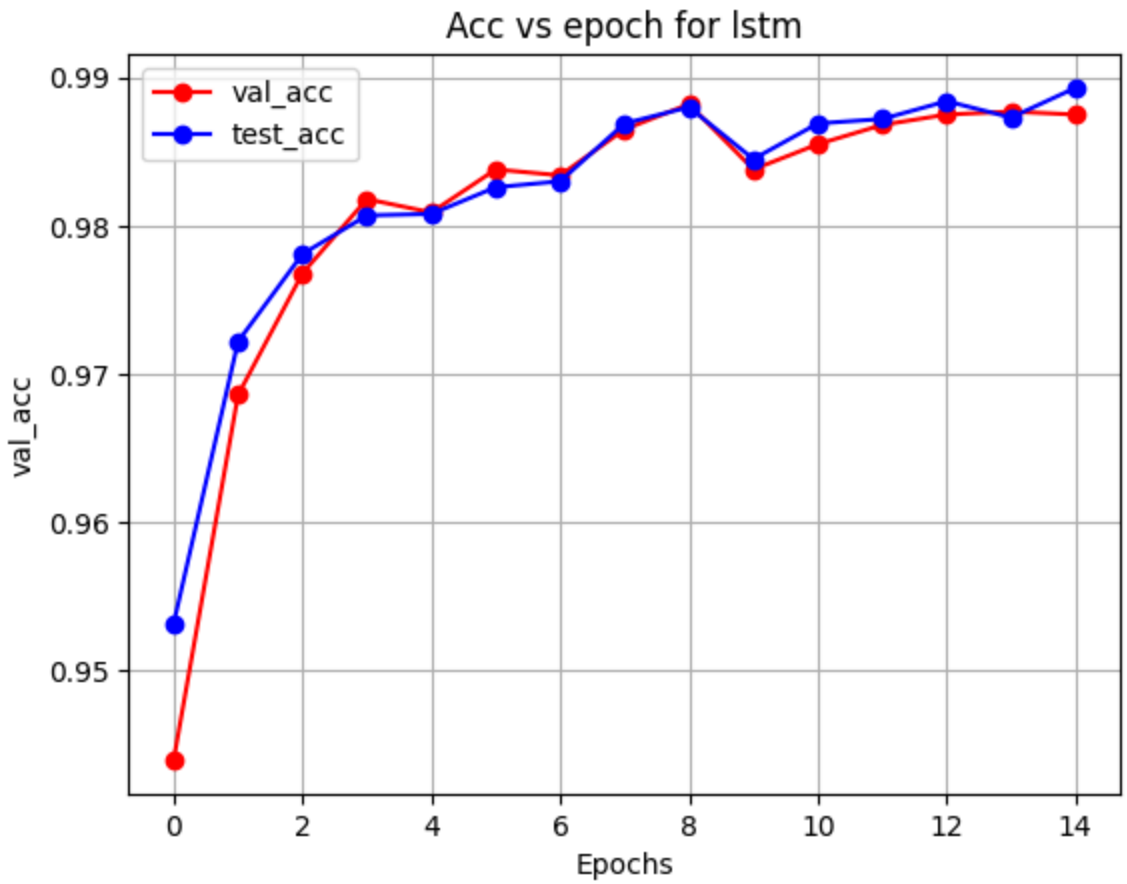
HARISH K N
EP22B026

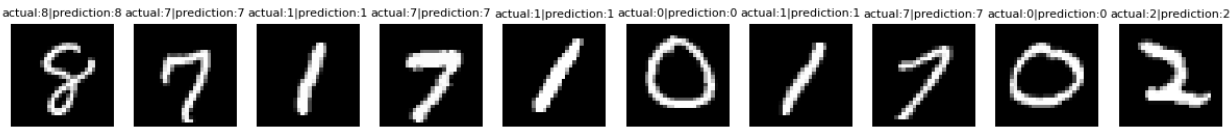


- LSTM -(Best accuracy-98.93%)



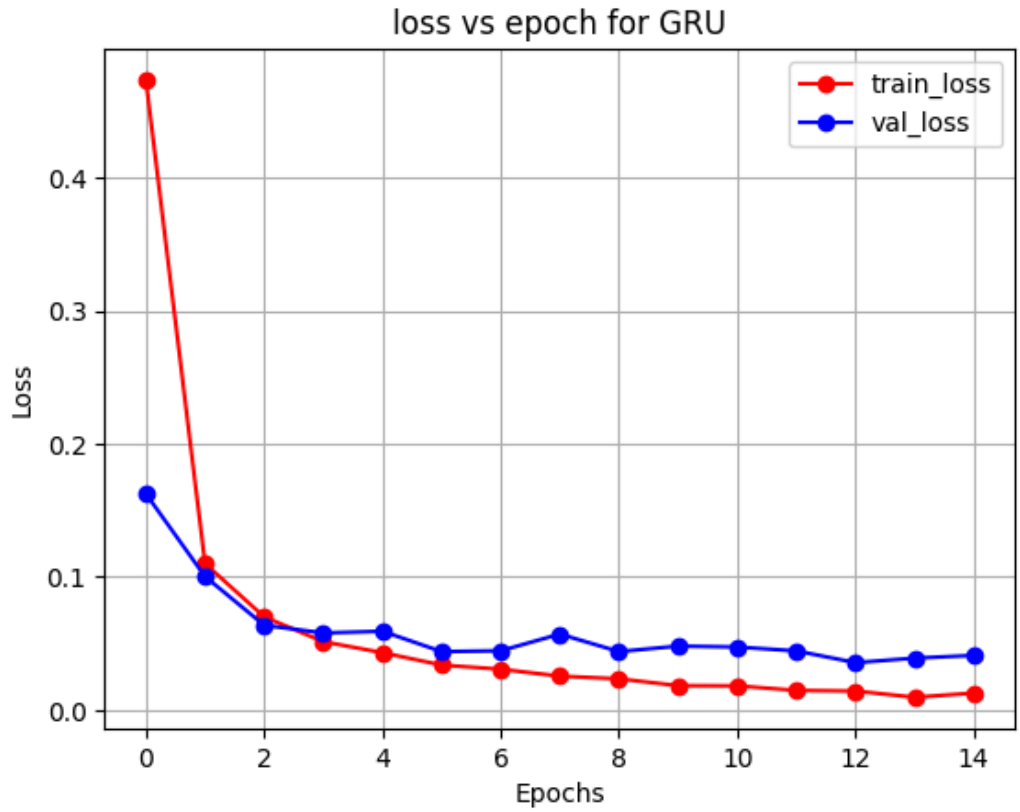
HARISH K N
EP22B026

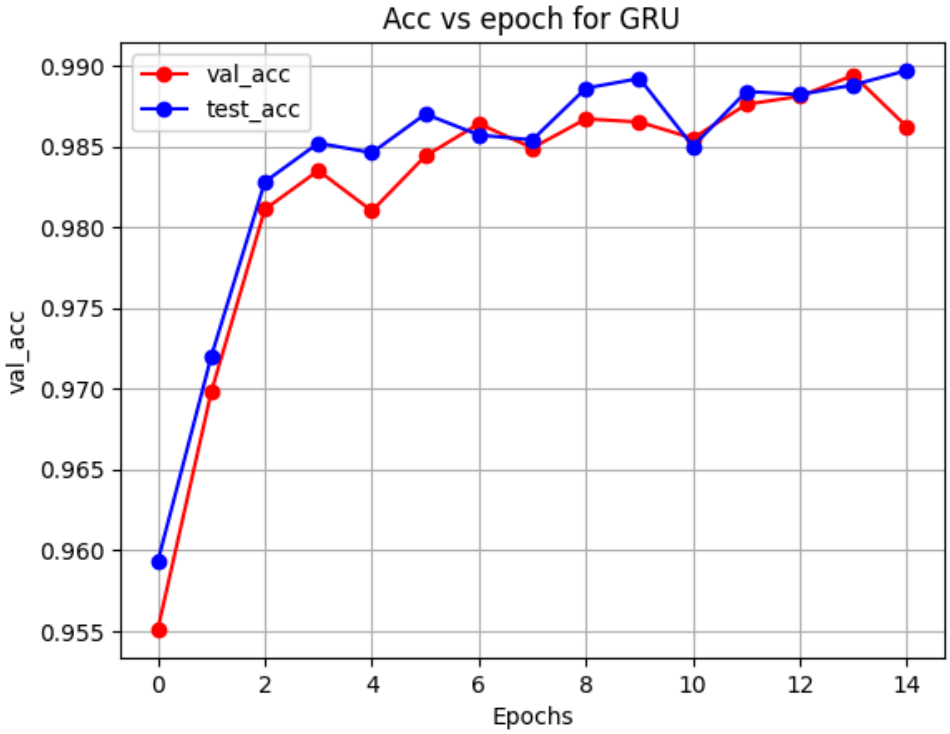




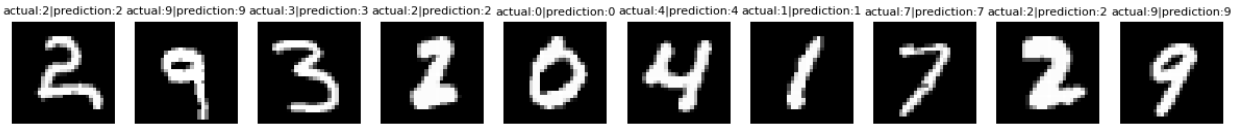
- GRU -(Best accuracy-98.92%)

HARISH K N
EP22B026

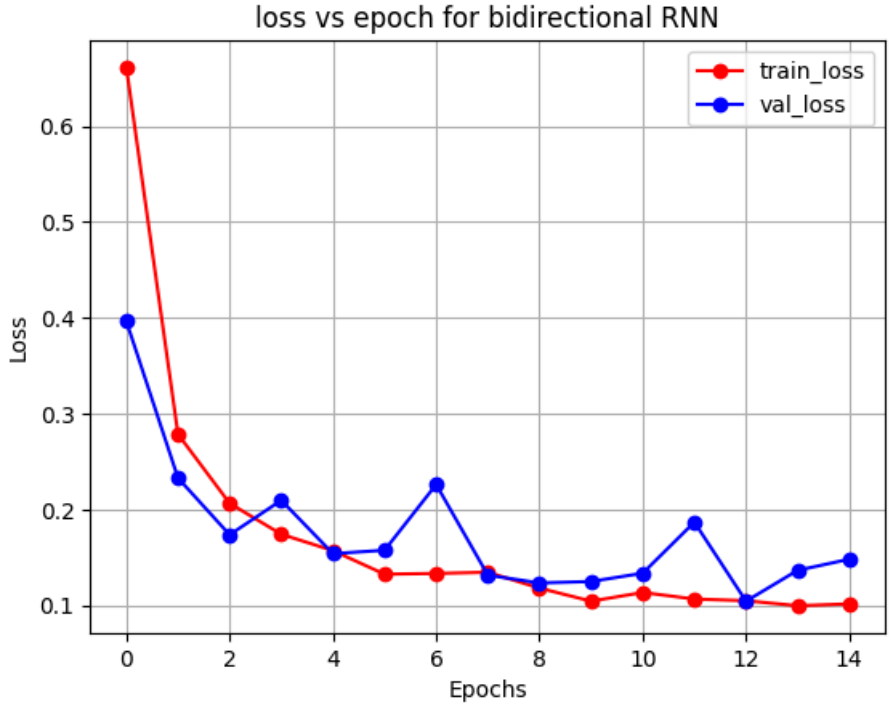


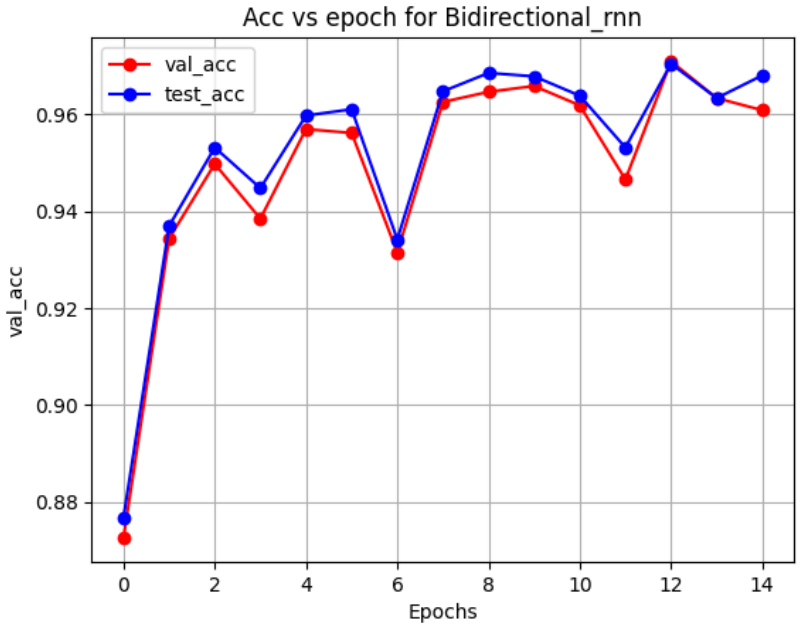


HARISH K N
EP22B026

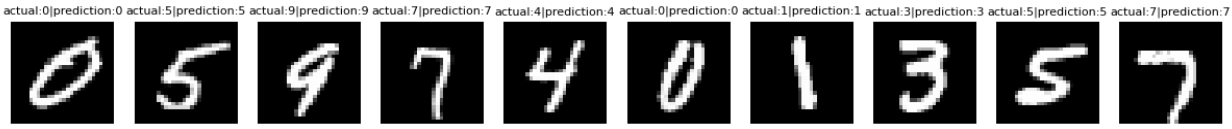


- Bidirectional RNN-(Best accuracy-97%)

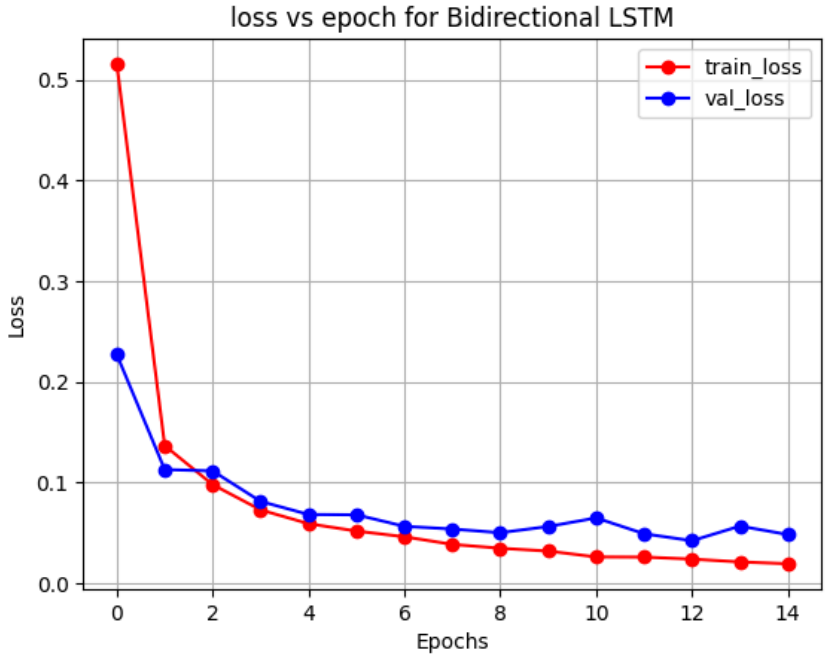


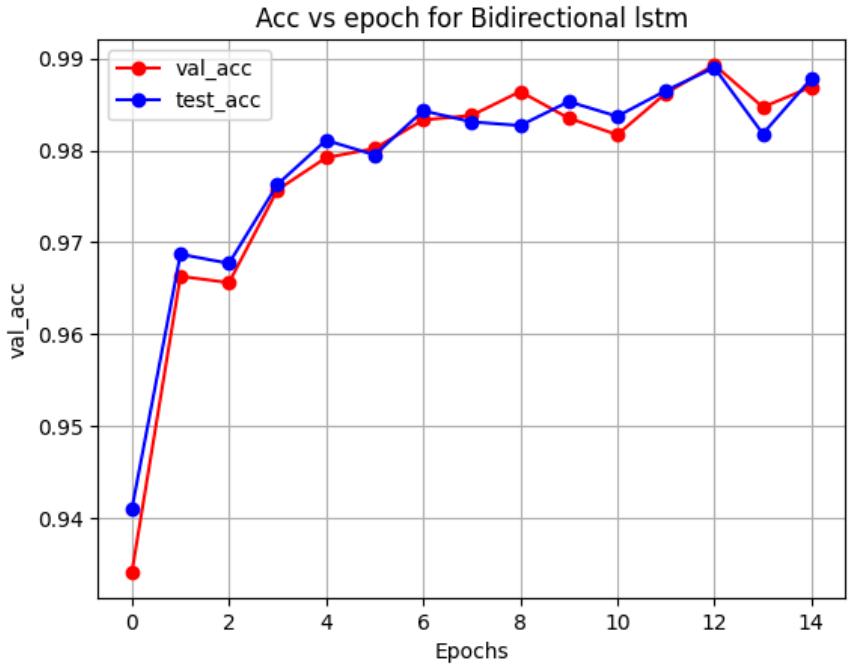


HARISH K N
EP22B026

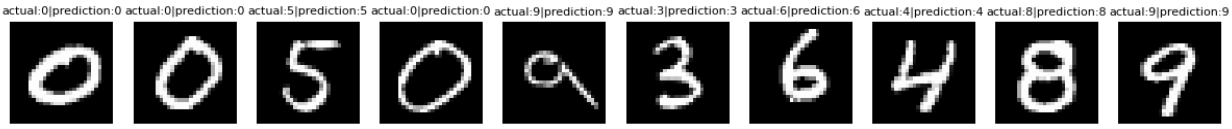


- Bidirectional LSTM-(Best accuracy-98.77%)





HARISH K N
EP22B026



Among all the models GRU has the best but GRU, LSTM, Bidirectional LSTM all have almost the same accuracy.

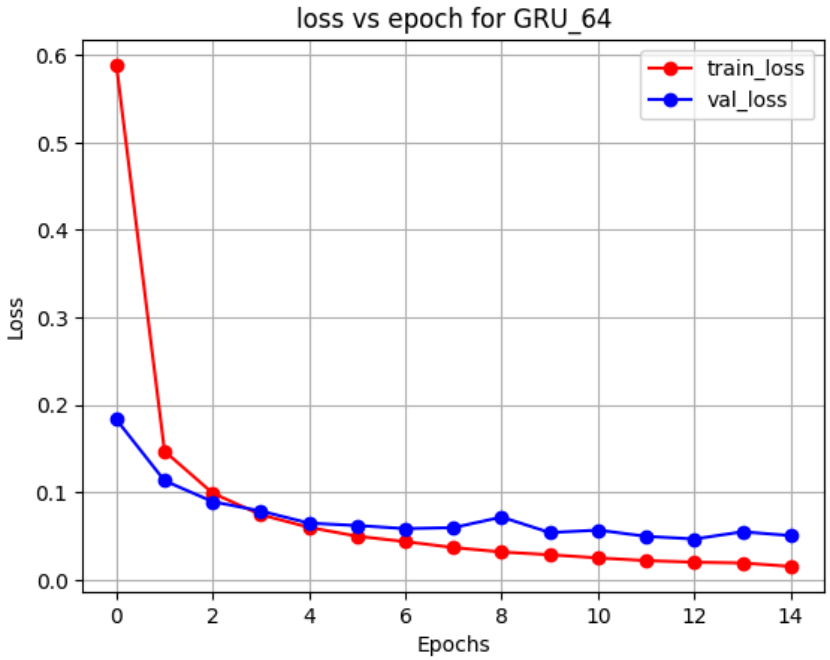
- Hidden state size variation for GRU :

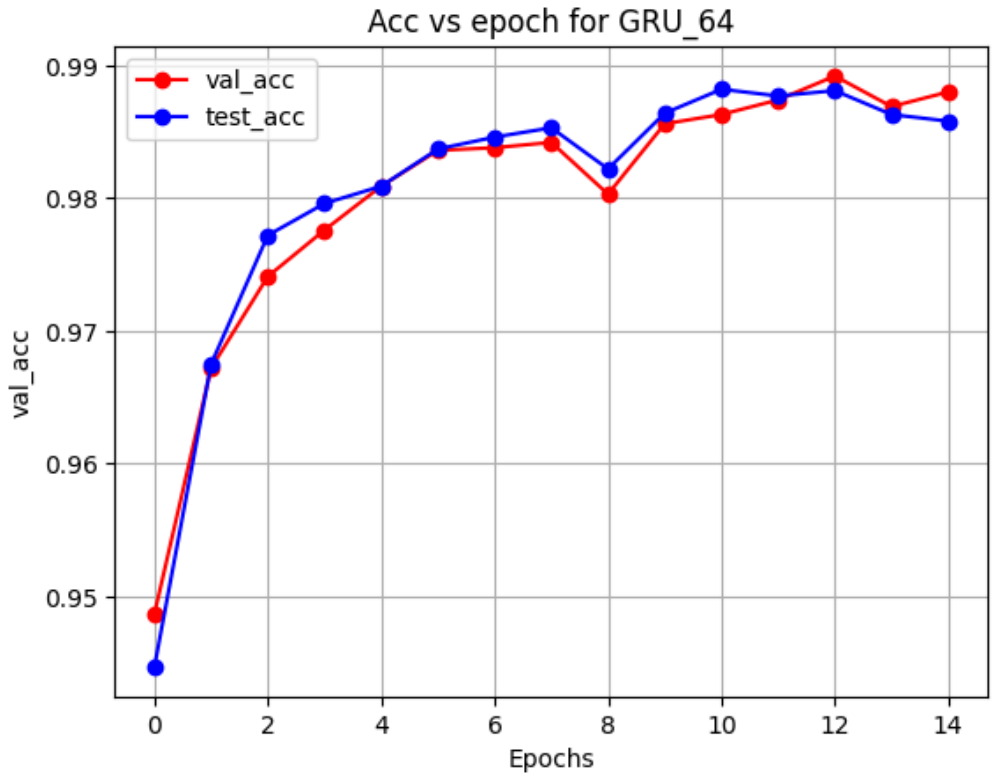
S.No	Hidden_state_size	Best_test_accuracy
1.	64	98.82%
2.	128	98.92%
3.	256	99.12%

- OBSERVATION:

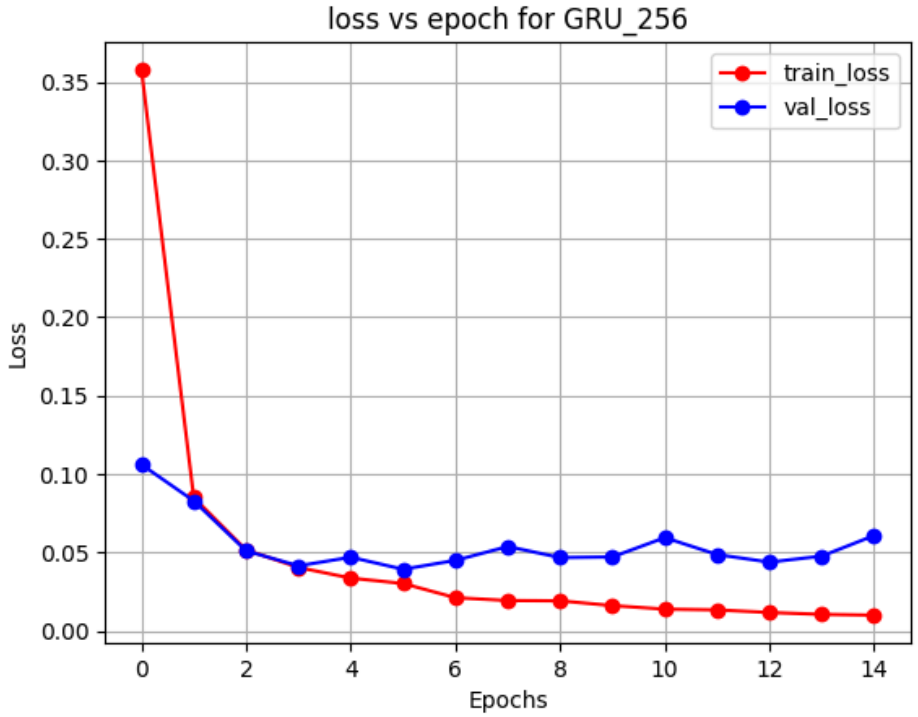
As the hidden state size increased from 64 to 256, the test accuracy improved from 98.82% to 99.12%. This suggests that a larger hidden state provides the model with greater **capacity**, allowing it to learn, remember, and model more complex relationships and patterns within the 28-step image sequences.

1. Hidden state=64

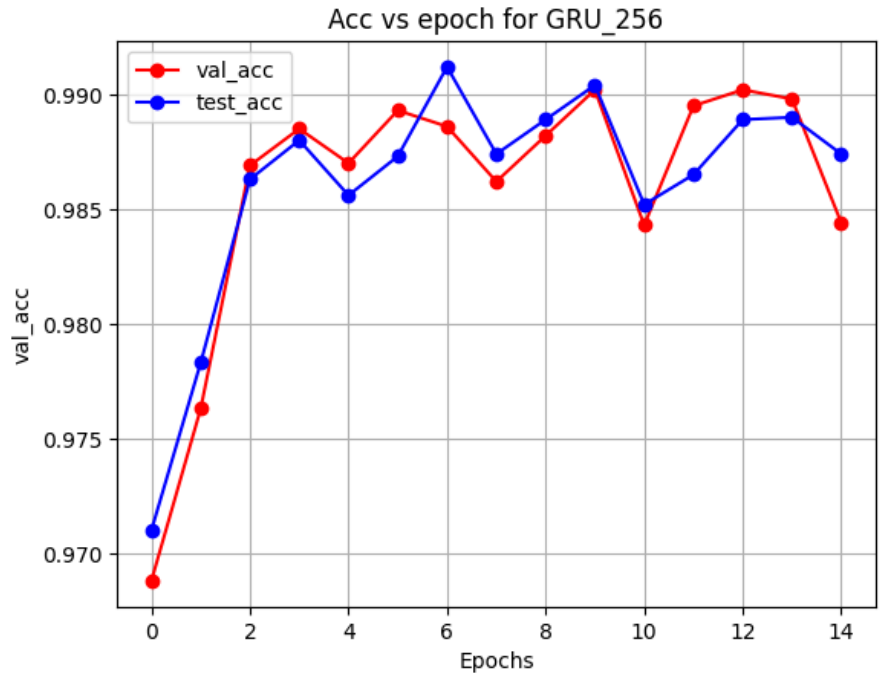




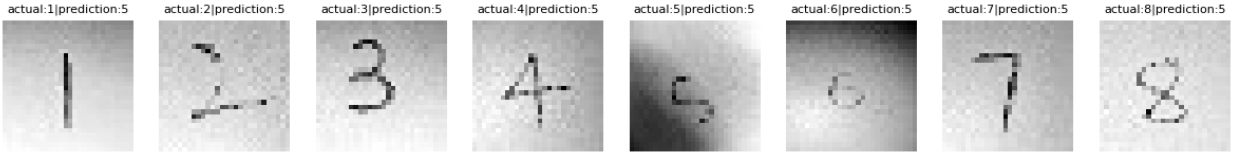
2. **Hidden state=128(refer 3rd set of images)**
3. **Hidden state=256**



HARISH K N
EP22B026



- **Hand written digit recognition:**



Our model failed due to domain shift. It was only trained on clean, white-on-black, standardized MNIST digits, so it never learned to ignore features like background noise, light gray backgrounds, shadows, or different handwriting styles. When it sees your custom images, which are a completely different "domain," it gets confused by these unexpected features and cannot generalize.

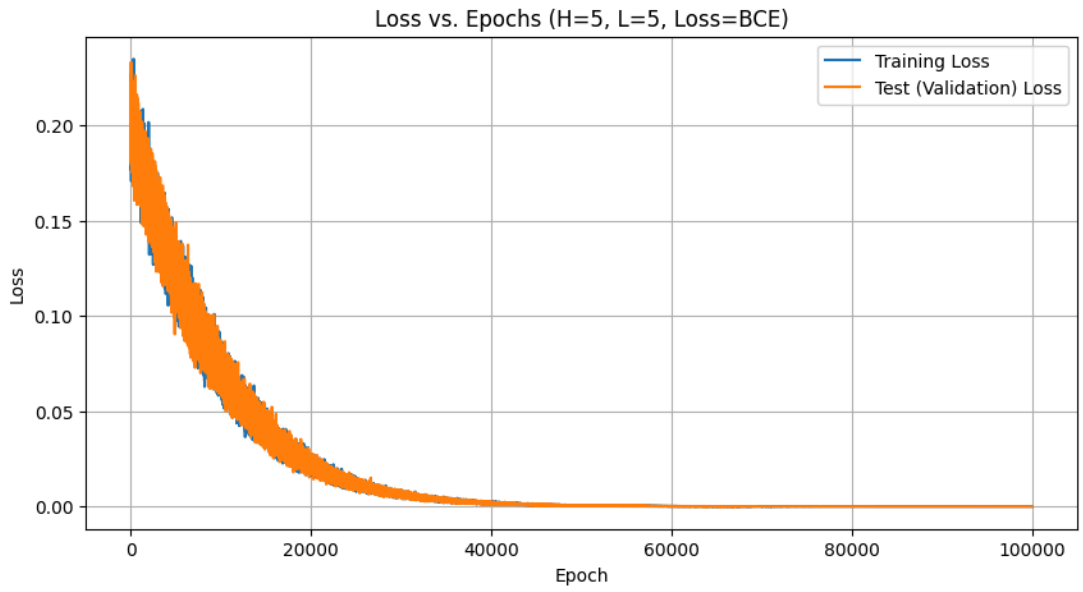
2. ADDING TWO BINARY STRINGS
lr=1e-4

S. N	L_train	L_test	Train_loss	Test_Loss	hidden size	Avg Bit Accuracy (L=1 to 20)	Epochs (conv epoch)	Batch_ Size	loss_fn
1	5	5	0	0	5	99.999	100000 (40000)	128	BCE
2	5	5	0.0346	0.0330	5	91.722	50000	128	MSE

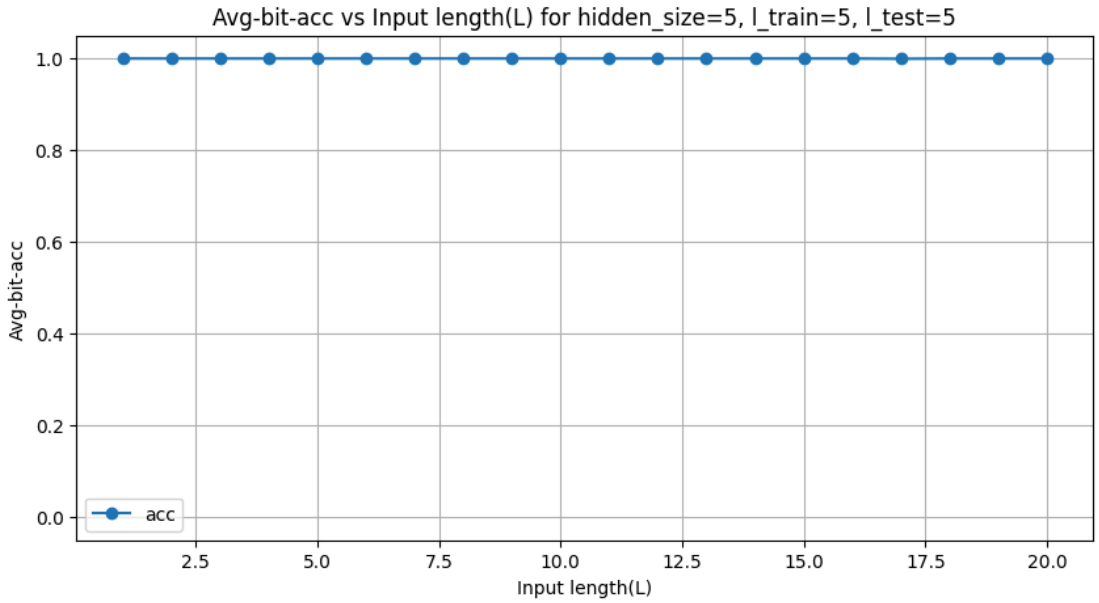
HARISH K N
EP22B026

3	5	5	0	0	10	99.999	50000 (35000)	128	BCE
4	5	5	0	0	10	99.843	50000 (25000)	128	MSE
5	5	5	0	0	100	81.303	22000 (21000)	128	BCE
6	5	5	0	0	100	81.097	20000 (15000)	128	MSE
7	3	5	0	0	10	99.995	50000 (30000)	128	BCE
8	10	5	0	0	10	100	50000 (30000)	128	BCE

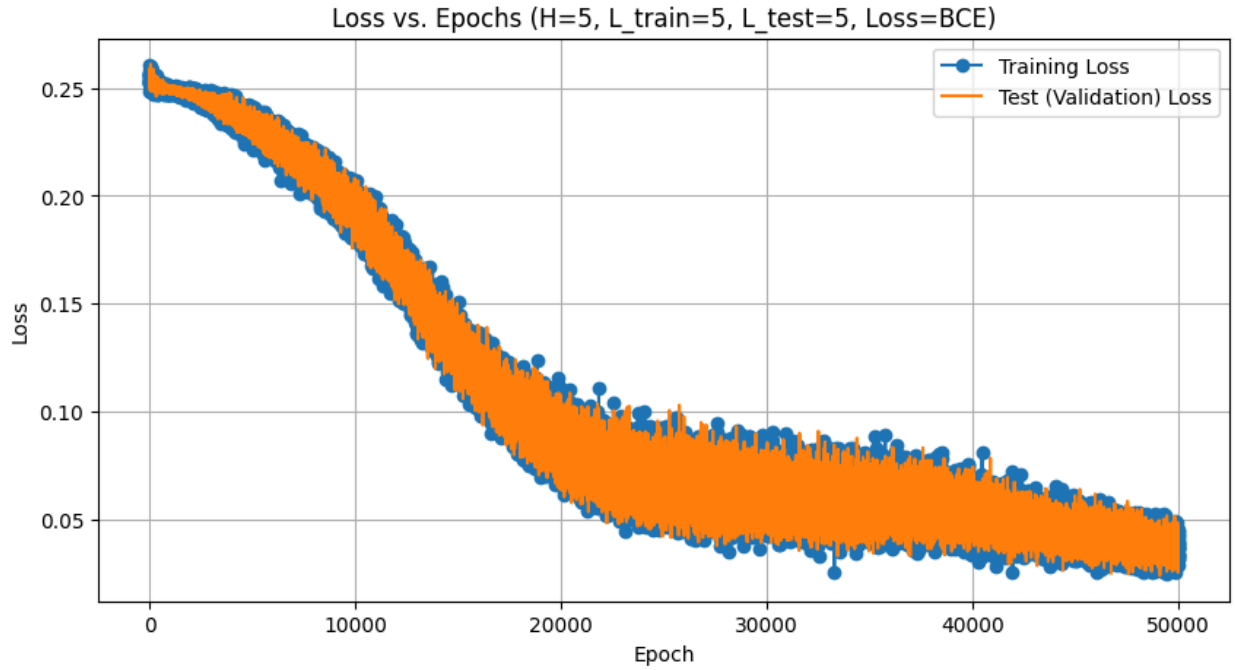
1.



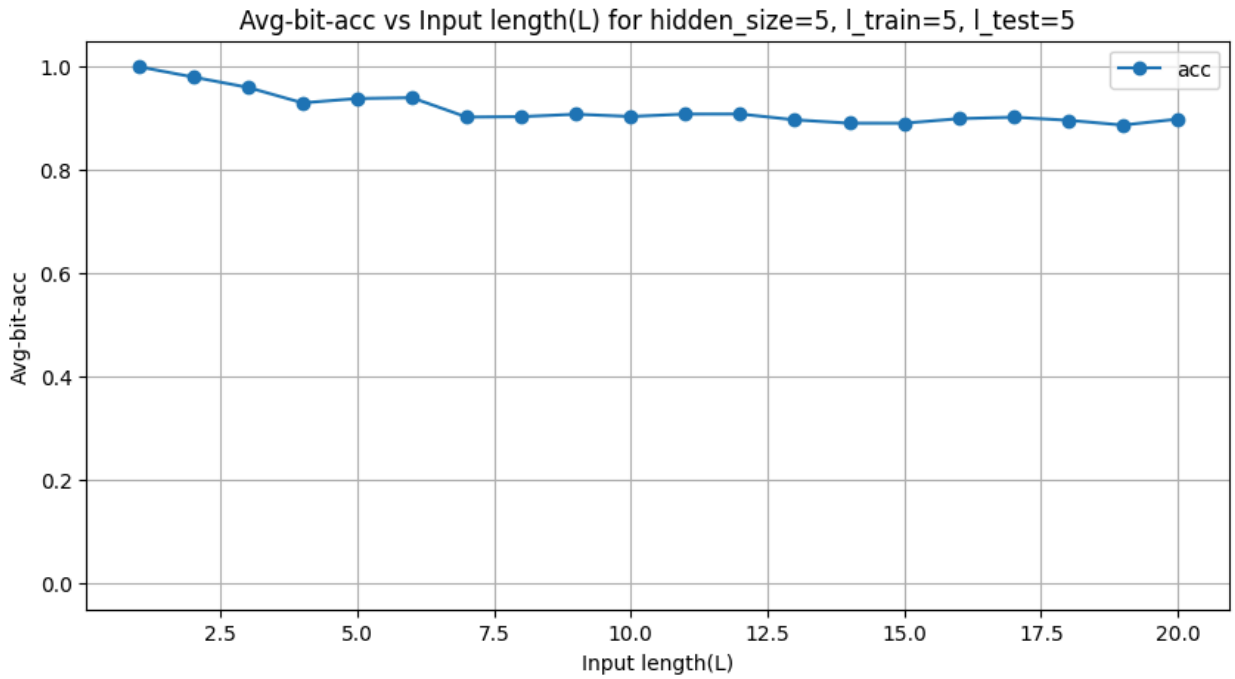
HARISH K N
EP22B026



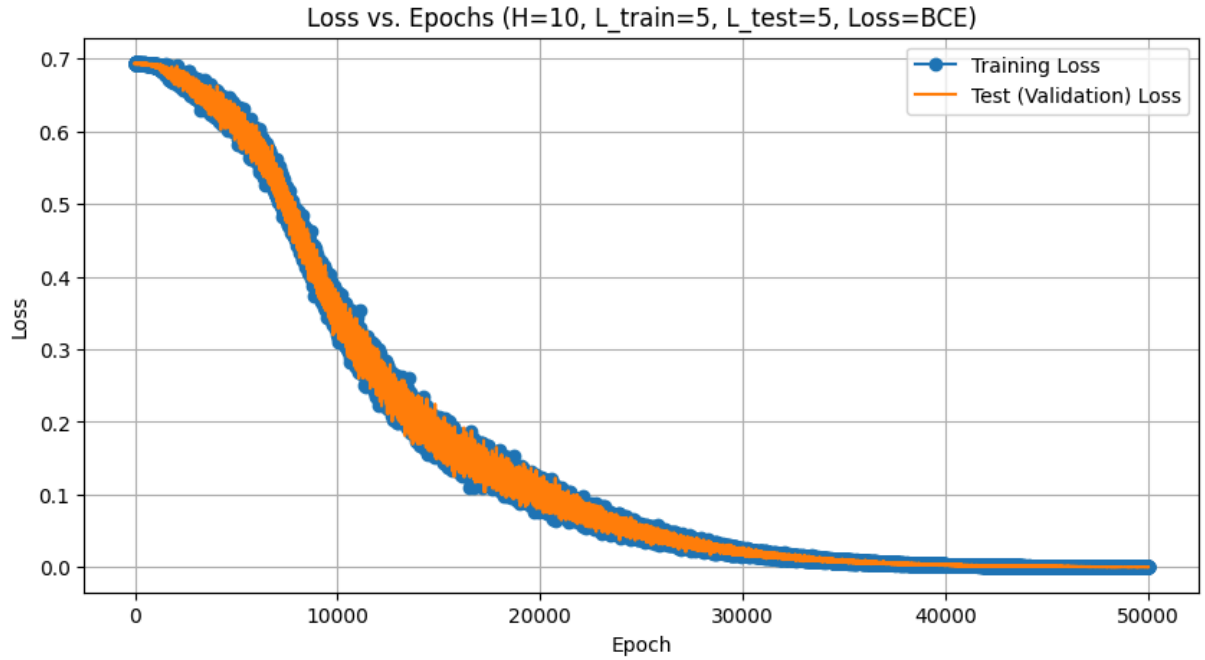
2.



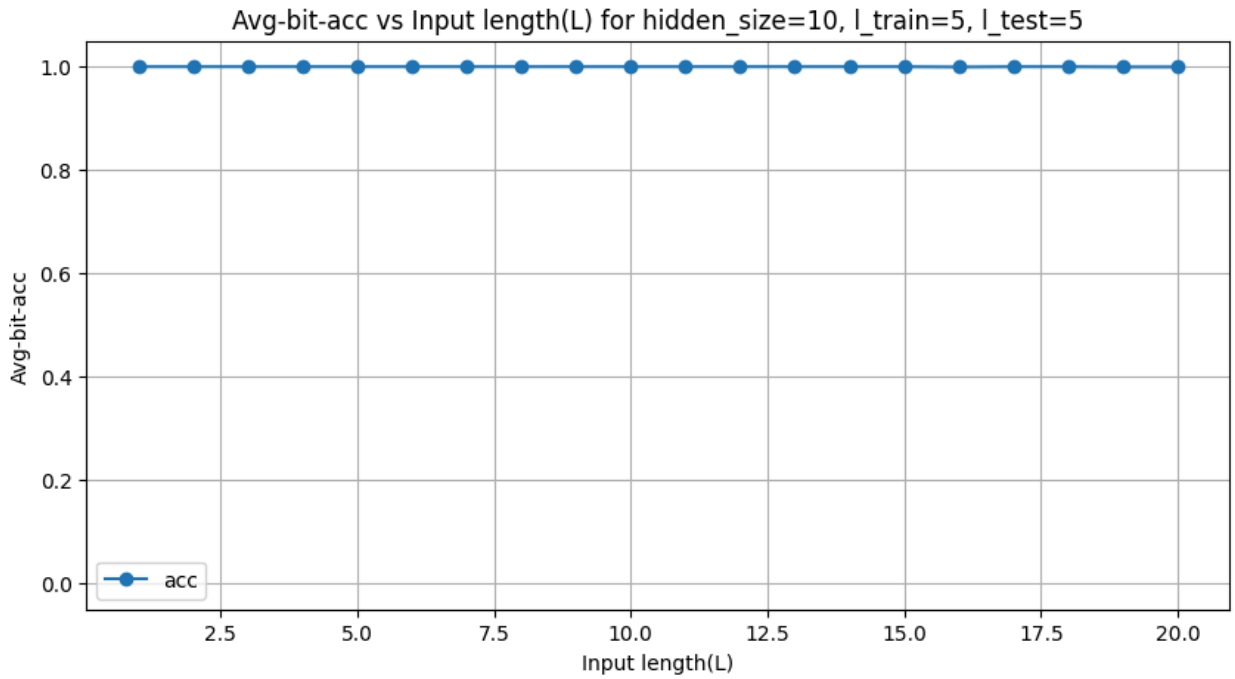
HARISH K N
EP22B026



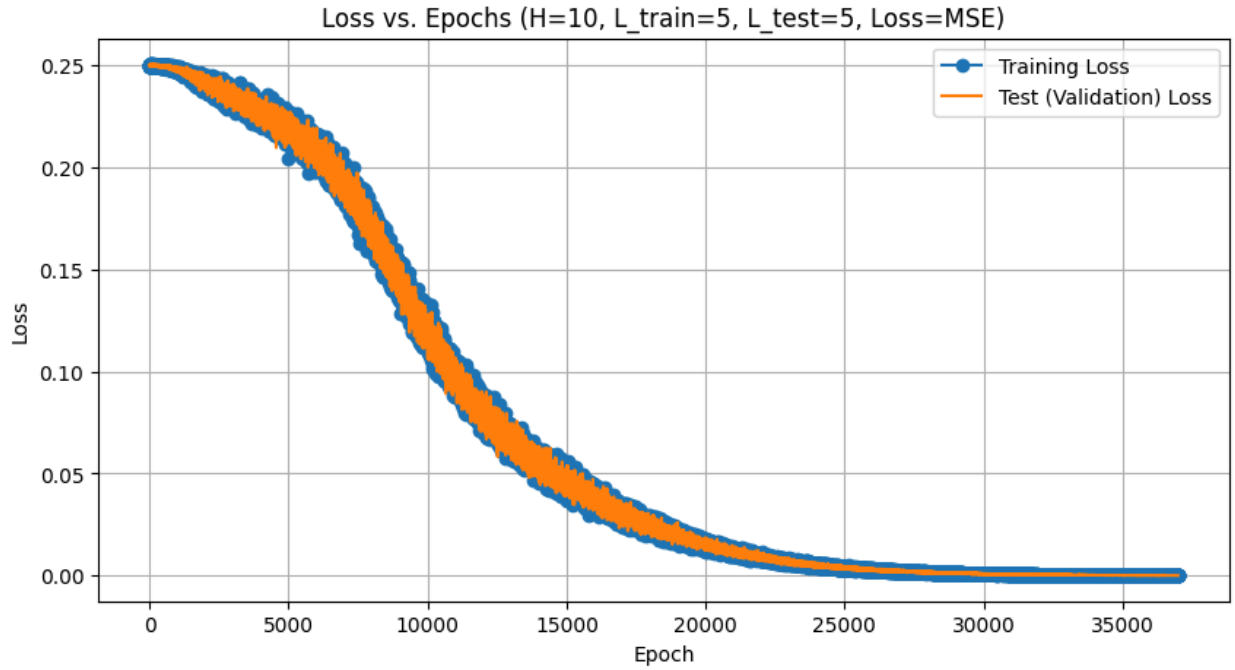
3.



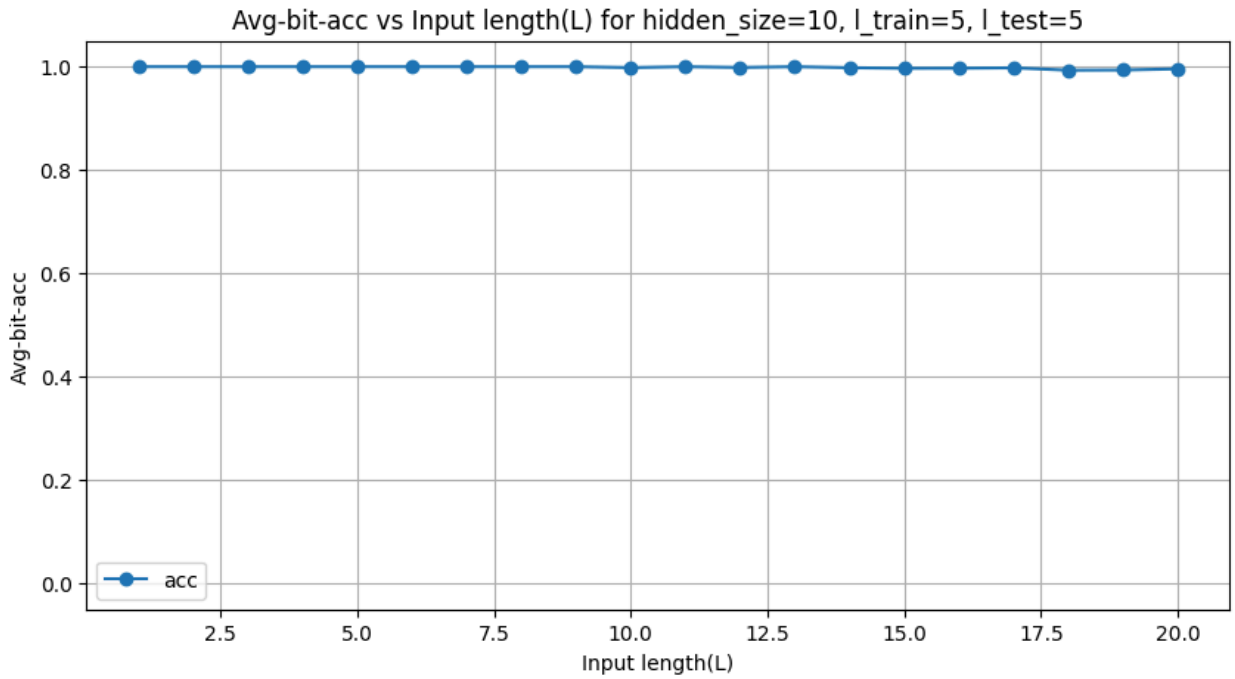
HARISH K N
EP22B026



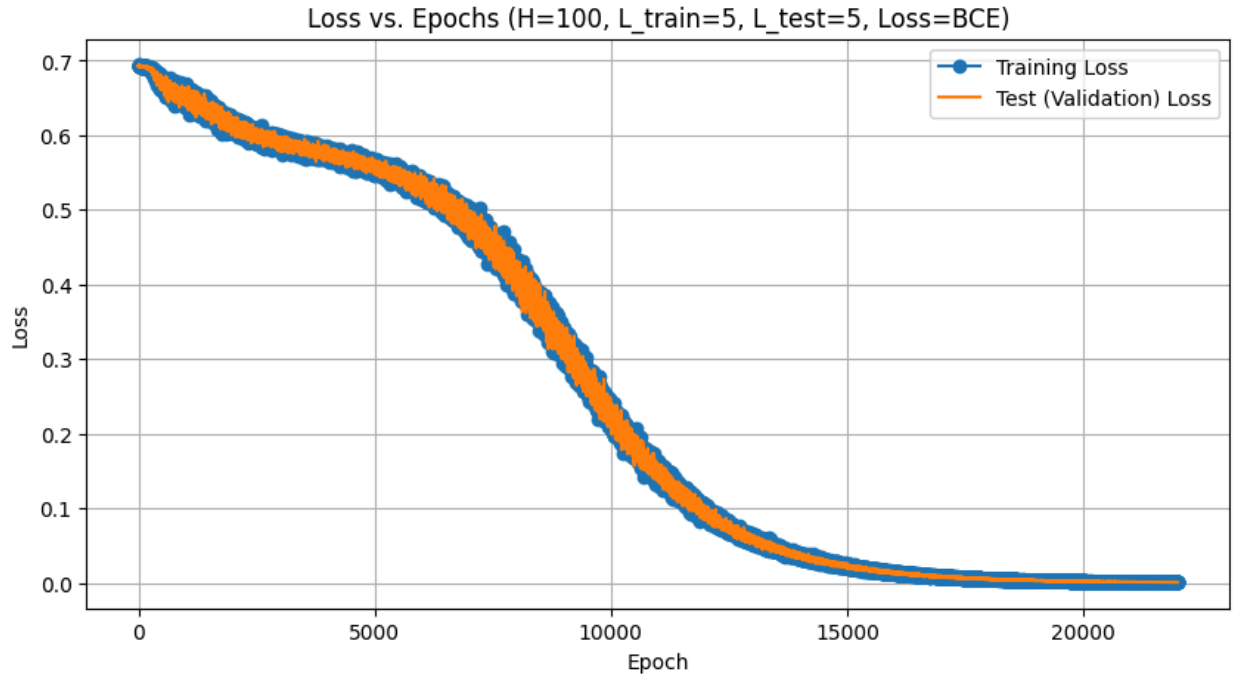
4.



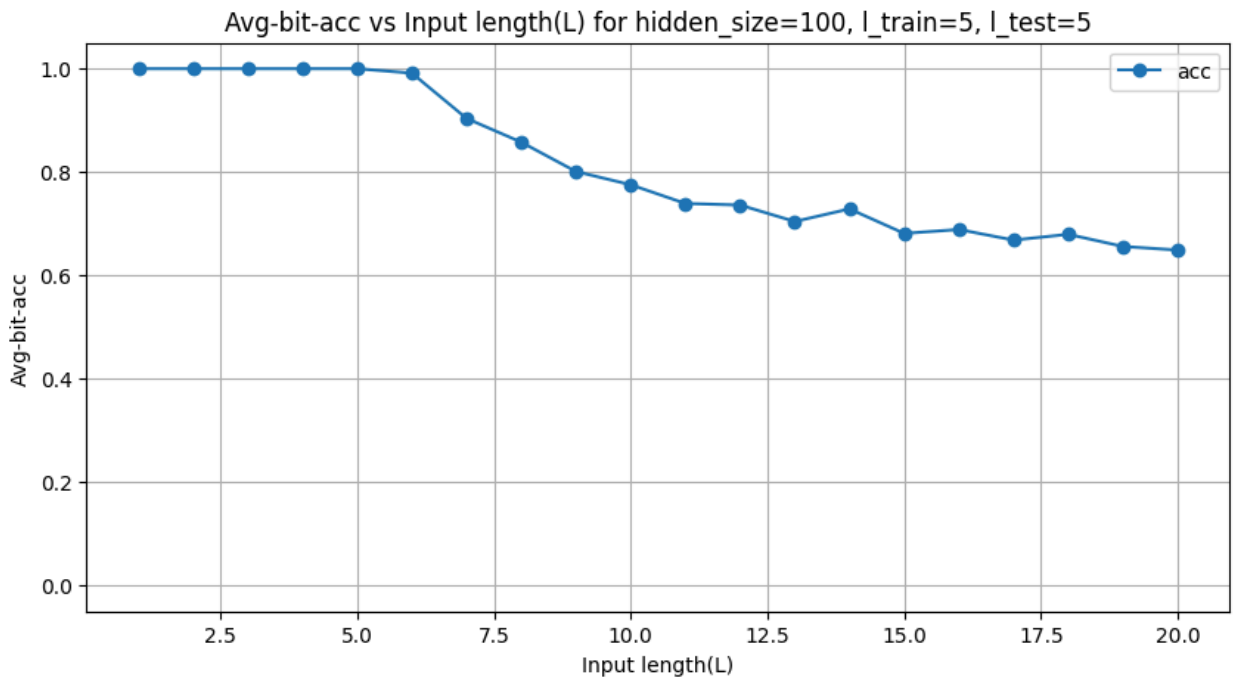
HARISH K N
EP22B026



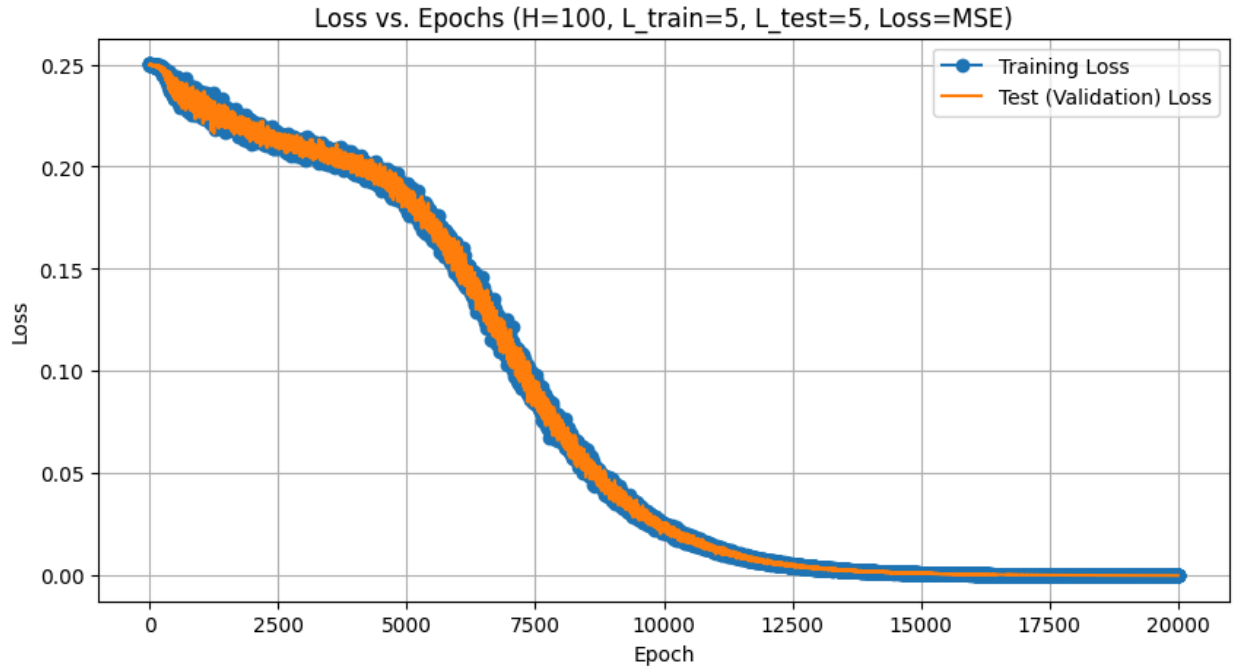
5.



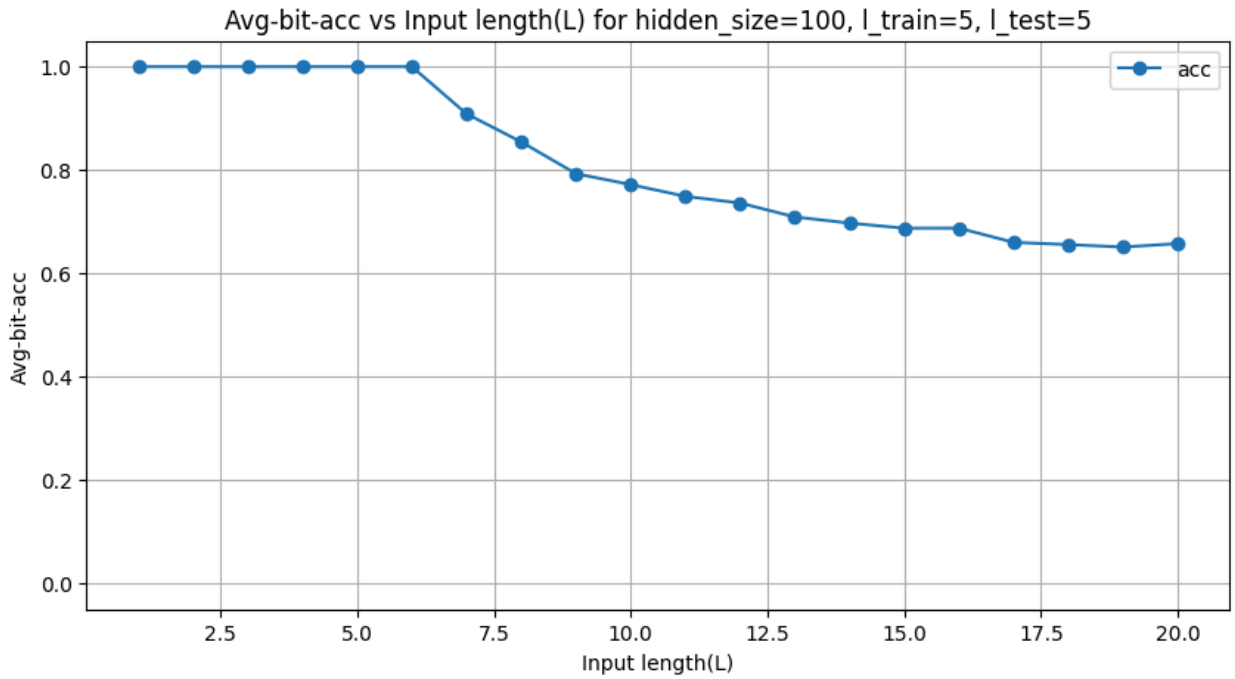
HARISH K N
EP22B026



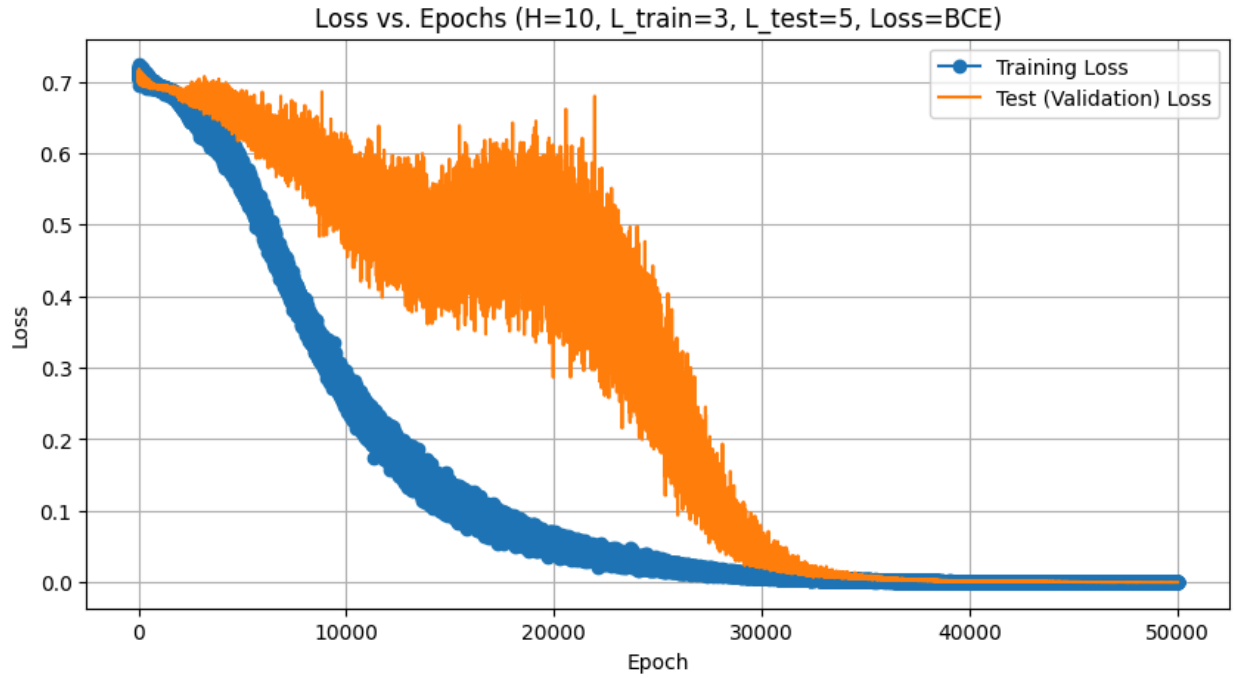
6.



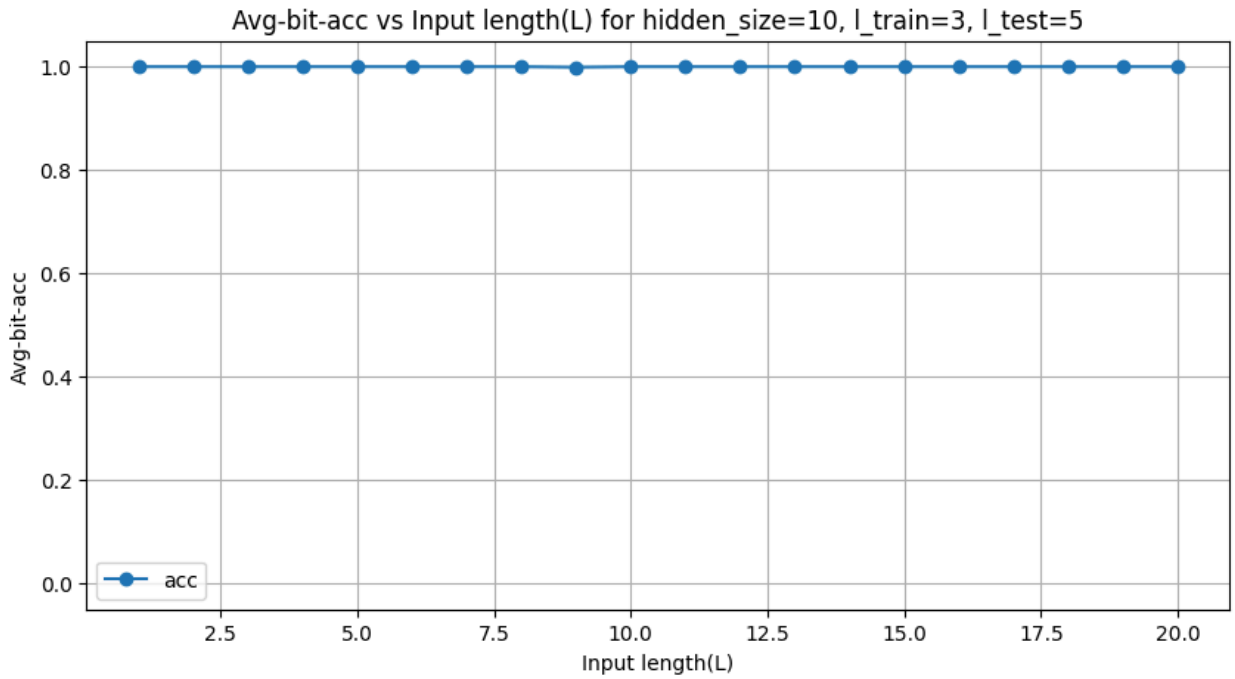
HARISH K N
EP22B026



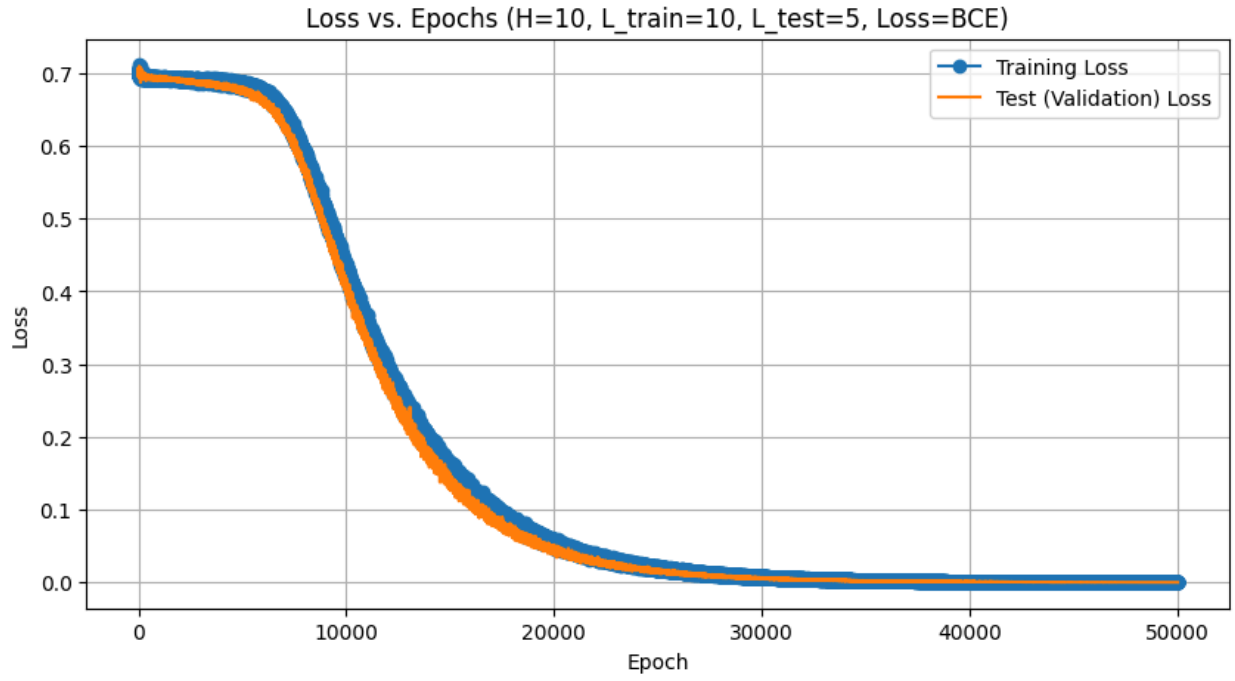
7.

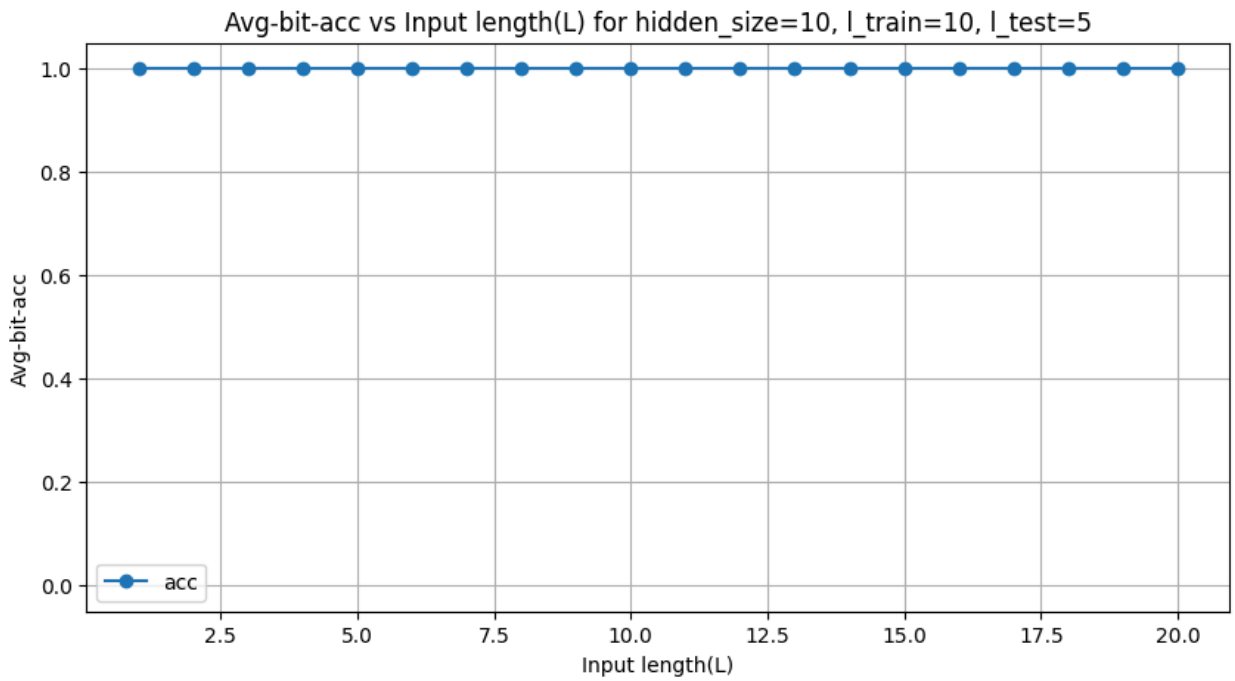


HARISH K N
EP22B026



8.





1. Experiment 1: Varying State Vector Size

Increasing the hidden size from 5 to 10 resulted in near-perfect (99.99%) average bit accuracy. However, a further increase to `hidden_size=100` caused a significant accuracy drop to 81.3%. This suggests the largest model (`H=100`) **overfit** the short `L_train=5` sequences and fails to generalize to the longer sequences in the L=1 to 20 test. The smaller models (`H=5`, `H=10`) were more robust and learned a generalizable solution.

2. Experiment 2: MSE vs. Cross-Entropy Loss

Binary Cross-Entropy (BCE) was the superior loss function. While the data shows that **MSE also converged to zero error** on a long run for larger hidden sizes (`H=10`, `H=100`), BCE was more consistent. It performed significantly better for `H=5` (99.9% vs 91.7%) and is the more appropriate choice for this binary classification task.

3. Experiment 3: Varying Training Length

As the table shows, all models, regardless of training length (`L_train=3, 5, 10`), **eventually converged to zero error** when tested on `L_test=5`. The graphs, however, reveal a key insight into *how* they converged:

- **Graph 1 (`L_train=3`, `L_test=5`):** The extremely volatile test loss shows the model struggling to **generalize to a sequence longer than it was trained on**.

- **Graph 3 (`L_train=10`, `L_test=5`):** The test loss perfectly tracks the training loss because the test task (`L=5`) was a simpler subset of the training task (`L=10`).

This confirms that a model only learns to robustly generalize to sequences up to the length it has seen during training.