Clean-Label-Backdoor Attacks (on horizontal Federal Learning)

Alexander Kiel

10.10.24

Methodology + Results + Conclusion

Methodology

AdamW

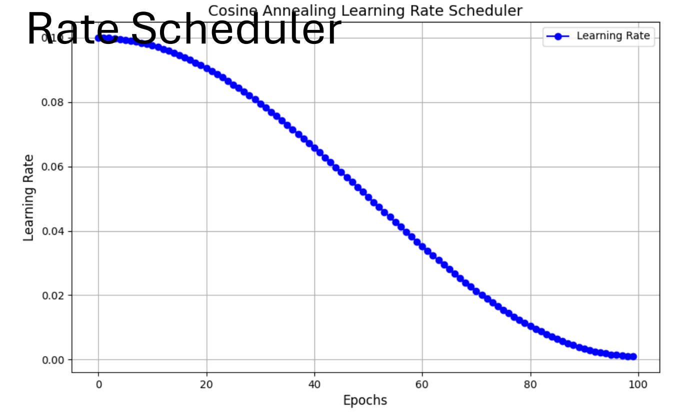
Adam

for $t = 1$ to $\dots$ do
  if *maximize* :
    $g_t \leftarrow -\nabla_\theta f_t(\theta_{t-1})$
  else
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
  $\boxed{\text{if } \lambda \neq 0 \quad g_t \leftarrow g_t + \lambda\theta_{t-1}}$
  $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
  $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
  $\widehat{m_t} \leftarrow m_t/(1-\beta_1^t)$
  $\widehat{v_t} \leftarrow v_t/(1-\beta_2^t)$
  if *amsgrad*
    $\widehat{v_t}^{max} \leftarrow \max(\widehat{v_t}^{max}, \widehat{v_t})$
    $\theta_t \leftarrow \theta_{t-1} - \gamma\widehat{m_t}/(\sqrt{\widehat{v_t}^{max}} + \epsilon)$
  else
    $\theta_t \leftarrow \theta_{t-1} - \gamma\widehat{m_t}/(\sqrt{\widehat{v_t}} + \epsilon)$

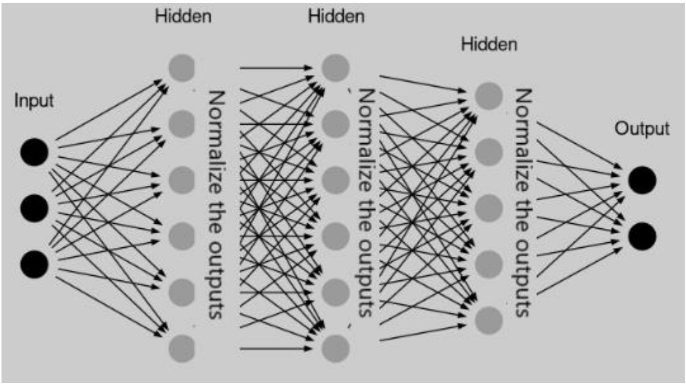
AdamW

for $t = 1$ to $\dots$ do
  if *maximize* :
    $g_t \leftarrow -\nabla_\theta f_t(\theta_{t-1})$
  else
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
  $\boxed{\theta_t \leftarrow \theta_{t-1} - \gamma\lambda\theta_{t-1}}$
  $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
  $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
  $\widehat{m_t} \leftarrow m_t/(1-\beta_1^t)$
  $\widehat{v_t} \leftarrow v_t/(1-\beta_2^t)$
  if *amsgrad*
    $\widehat{v_t}^{max} \leftarrow \max(\widehat{v_t}^{max}, \widehat{v_t})$
    $\theta_t \leftarrow \theta_t - \gamma\widehat{m_t}/(\sqrt{\widehat{v_t}^{max}} + \epsilon)$
  else
    $\theta_t \leftarrow \theta_t - \gamma\widehat{m_t}/(\sqrt{\widehat{v_t}} + \epsilon)$

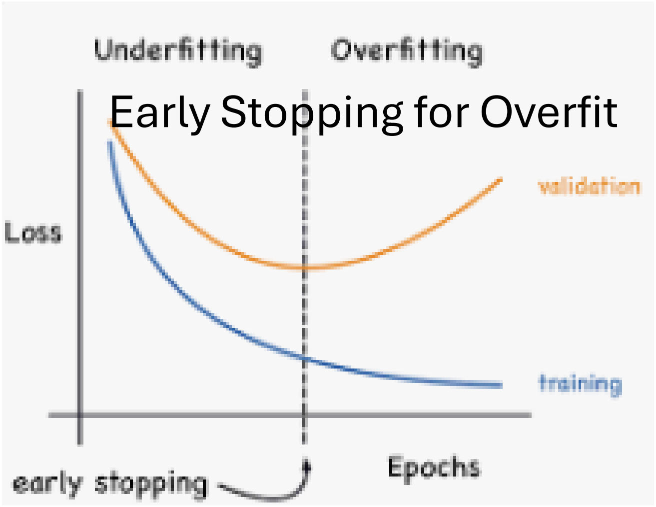
Cosine Annealing Learning Rate Scheduler



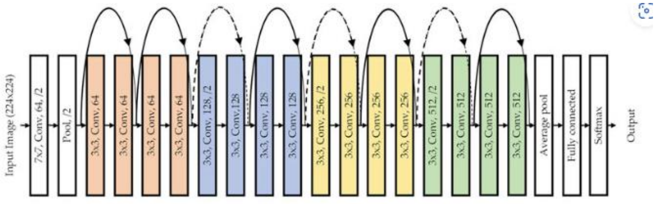
Normalization



Early Stopping for Overfit

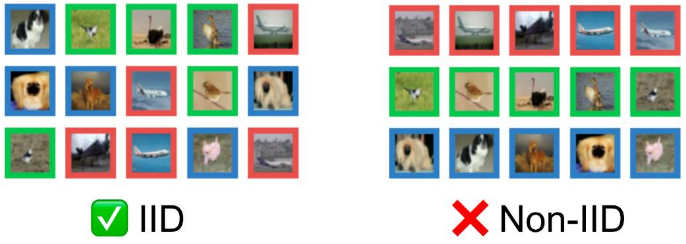


ResNet18



Structure of the Resnet-18 Model.

Non-IID



Non IID

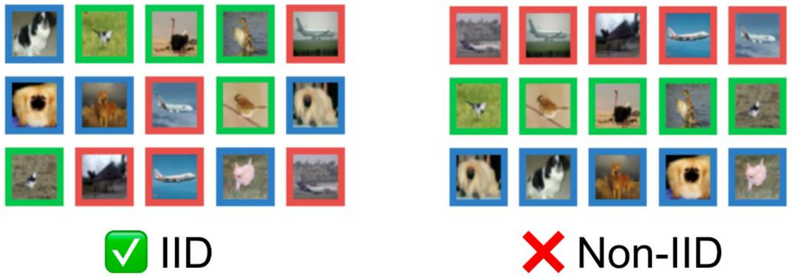


Table 1: Progression of Validation and Poison Metrics at Various Percentages of Epochs (Fashion-MNIST)

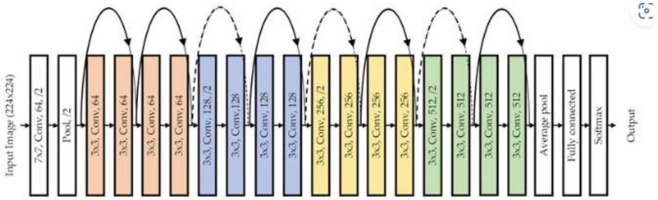
Percentage Epoch (1st)	**1% (3rd)**	**10% (6th)**	**20% (15th)**	**50% (23rd)**	**75% (27th)**	**90% (29th)**	**95% (30th)**	**100%**
Validation Loss	1.0699	0.4059	0.3201	0.2448	0.2217	0.2170	0.2148	0.2145
Validation Accuracy	0.7471	0.8507	0.8831	0.9093	0.9177	0.9213	0.9209	0.9240
Poison Loss	2.1165	3.0828	1.7900	0.0944	0.0117	0.0048	0.0038	0.0017
Poison Accuracy	0.1110	0.2160	0.4540	0.9740	0.9970	0.9980	0.9990	1.0000

Table 2: Progression of Validation and Poison Metrics at Various Percentages of Epochs with non iid function (Fashion-MNIST)

Percentage Epoch	**1% 1st**	**10% 3rd**	**20% 6th**	**50% 15th**	**75% 23rd**	**90% 27th**	**95% 29th**	**100% 30th**
Validation Loss	1.48396	0.60744	0.48642	0.32297	0.25522	0.24123	0.24179	0.24180
Validation Accuracy	0.74670	0.78240	0.81630	0.87730	0.90510	0.91130	0.91330	0.91160
Poison Loss	1.98446	2.44363	2.76638	1.00774	0.08870	0.02909	0.03659	0.01755
Poison Accuracy	0.19300	0.10000	0.16000	0.61400	0.97300	0.98800	0.98700	0.99200

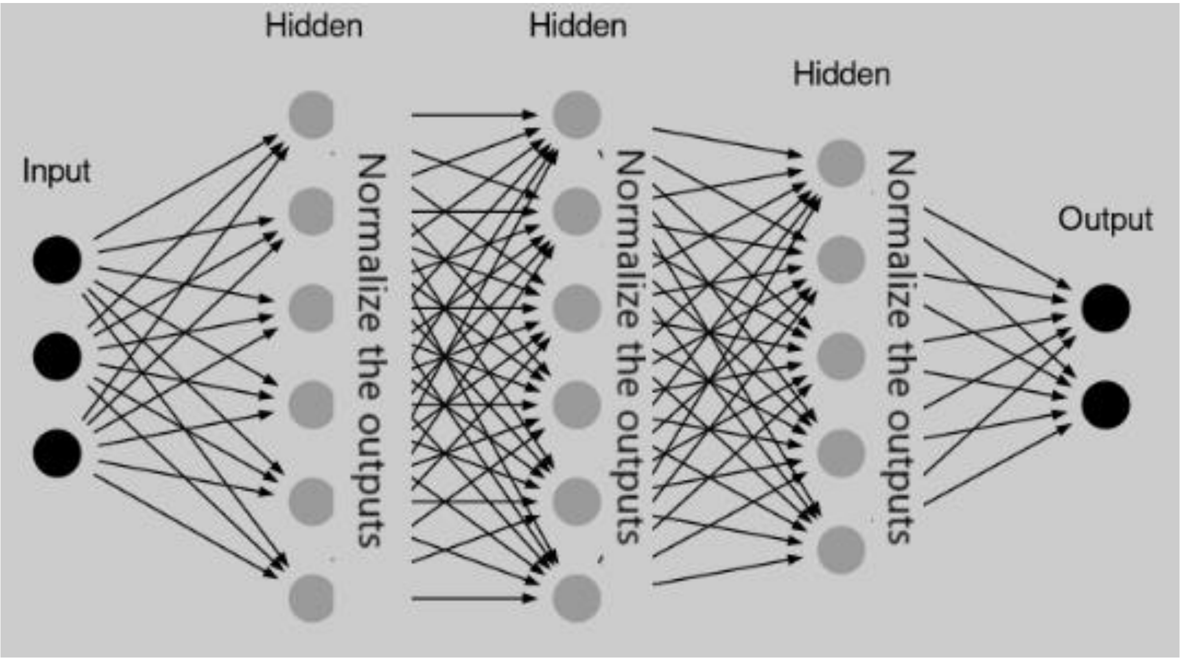
Resnet18

```python
class ResNet18(nn.Module):
    def __init__(self):
        super(ResNet18, self).__init__()
        self.model = models.resnet18(pretrained=False)  # Use ResNet18 model
        self.model.fc = nn.Linear(self.model.fc.in_features, 10)  # Adjust output layer to 10 classes for CIFAR-10
```



Structure of the Resnet-18 Model.

Batch Normalization




```
self.bn1 = nn.BatchNorm2d(64)
self.bn2 = nn.BatchNorm2d(128)
self.bn3 = nn.BatchNorm2d(256)
```

This leads to the normalization of the layer outputs:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

where μ_B and σ_B^2 are the mini-batch mean and variance, respectively.

AdamW

- AdamW optimizer with weight decay:

$$w_{t+1} = w_t - \eta_t \cdot \frac{\partial L}{\partial w_t} + \lambda w_t$$

where λ is the weight decay.

Adam

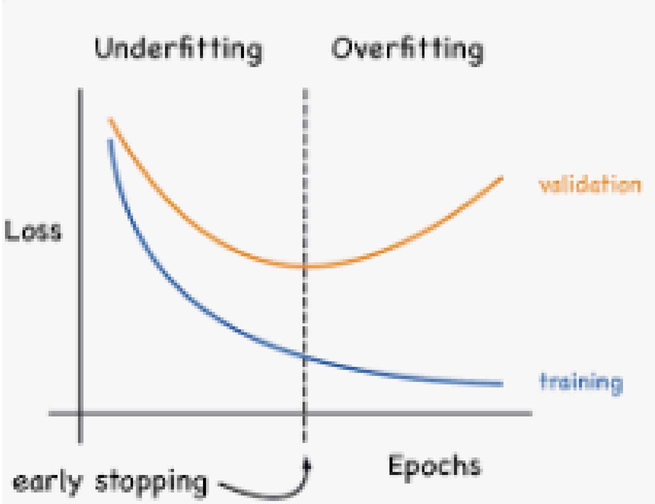
for $t = 1$ **to** $\ldots$ **do**
  if $maximize$:
    $g_t \leftarrow -\nabla_\theta f_t(\theta_{t-1})$
  else
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
  $\boxed{\textbf{if } \lambda \neq 0 \quad g_t \leftarrow g_t + \lambda\theta_{t-1}}$
  $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
  $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
  $\widehat{m_t} \leftarrow m_t / (1-\beta_1^t)$
  $\widehat{v_t} \leftarrow v_t / (1-\beta_2^t)$
  if $amsgrad$
    $\widehat{v_t}^{max} \leftarrow \max(\widehat{v_t}^{max}, \widehat{v_t})$
    $\theta_t \leftarrow \theta_{t-1} - \gamma\widehat{m_t} / (\sqrt{\widehat{v_t}^{max}} + \epsilon)$
  else
    $\theta_t \leftarrow \theta_{t-1} - \gamma\widehat{m_t} / (\sqrt{\widehat{v_t}} + \epsilon)$

AdamW

for $t = 1$ **to** $\ldots$ **do**
  if $maximize$:
    $g_t \leftarrow -\nabla_\theta f_t(\theta_{t-1})$
  else
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
  $\boxed{\theta_t \leftarrow \theta_{t-1} - \gamma\lambda\theta_{t-1}}$
  $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
  $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
  $\widehat{m_t} \leftarrow m_t / (1-\beta_1^t)$
  $\widehat{v_t} \leftarrow v_t / (1-\beta_2^t)$
  if $amsgrad$
    $\widehat{v_t}^{max} \leftarrow \max(\widehat{v_t}^{max}, \widehat{v_t})$
    $\theta_t \leftarrow \theta_t - \gamma\widehat{m_t} / (\sqrt{\widehat{v_t}^{max}} + \epsilon)$
  else
    $\theta_t \leftarrow \theta_t - \gamma\widehat{m_t} / (\sqrt{\widehat{v_t}} + \epsilon)$

Early Stopping

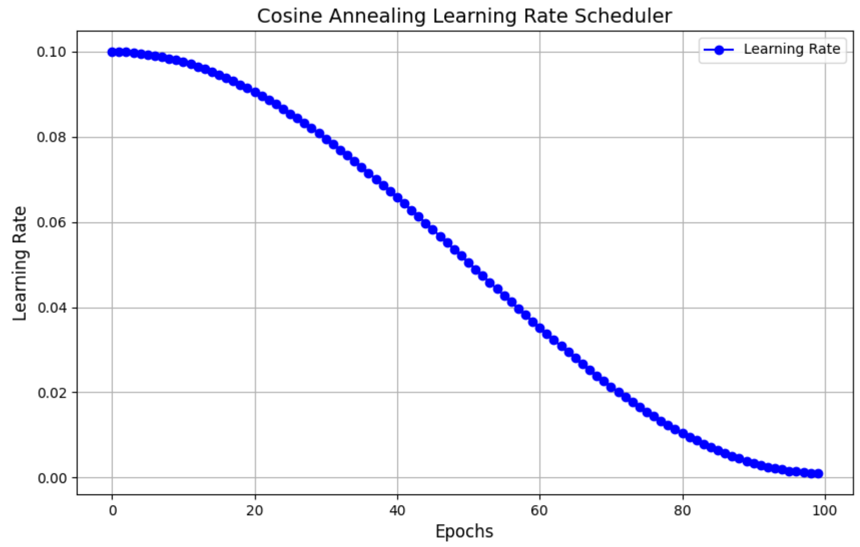
$$\text{Stop training if Accuracy}_t \leq \text{Best Accuracy}_{t-k}, \forall k \in [1, \text{Patience}]$$




```
Early stopping at round 15
 47%|                                                   | 14/30 [1:01:48<1:10:38, 264.90s/it]
Training finished.
```

Cosine Annealing Learning Rate Scheduler

- Cosine Annealing helps avoid premature convergence by gradually reducing the learning rate, thus allowing the model to explore the parameter space more effectively



- Cosine Annealing LR schedule:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\left(1 + \cos\left(\frac{T_{cur}}{T_{max}}\pi\right)\right)$$

where T_{cur} is the current epoch, and T_{max} is the total number of epochs.

Results

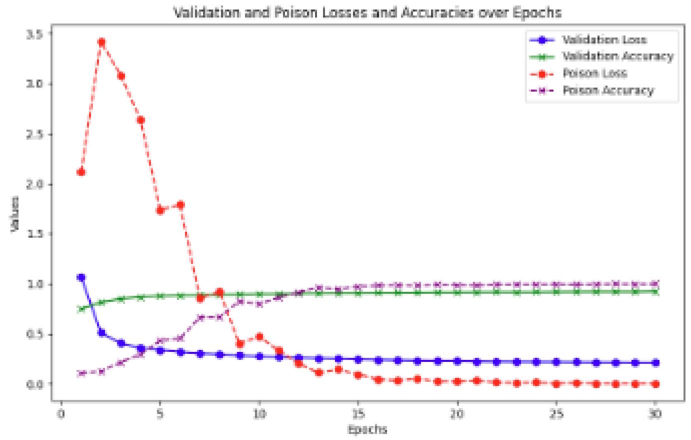


Figure 1: Experiment 1. Original Code and simple CNN with Fashion-MNIST in IID

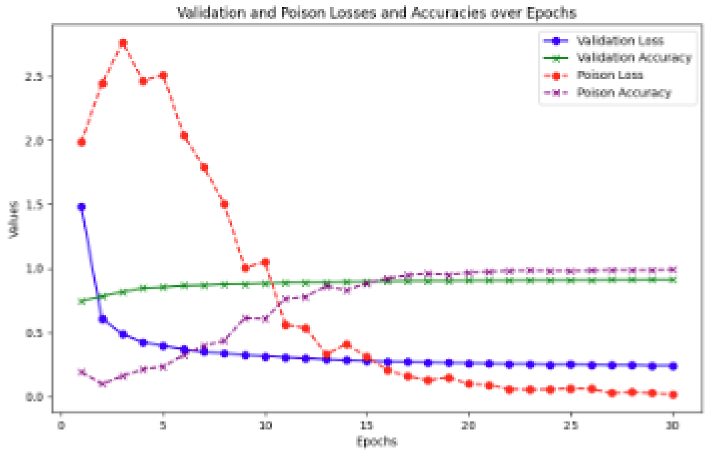


Figure 2: non IID function function with Fashion-MNIST and simple CNN

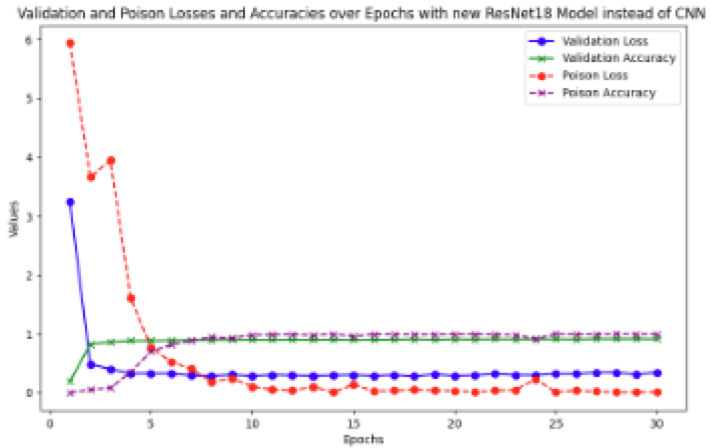


Figure 3: with Fashion-MNIST and ResNet18 in IID

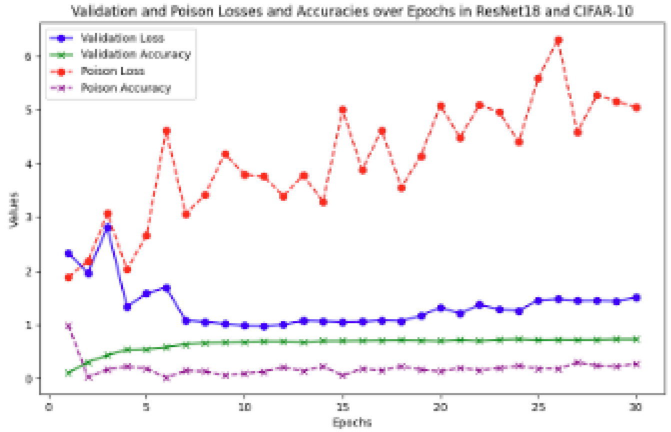
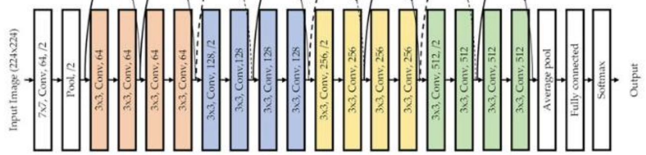


Figure 4: with CIFAR-10 and ResNet18



Structure of the Resnet-18 Model.

ResNet18

Table 1: Progression of Validation and Poison Metrics at Various Percentages of Epochs (Fashion-MNIST)

Percentage Epoch (1st)	1% (3rd)	10% (6th)	20% (15th)	50% (23rd)	75% (27th)	90% (29th)	95% (30th)	100%
Validation Loss	1.0699	0.4059	0.3201	0.2448	0.2217	0.2170	0.2148	0.2145
Validation Accuracy	0.7471	0.8507	0.8831	0.9093	0.9177	0.9213	0.9209	0.9240
Poison Loss	2.1165	3.0828	1.7900	0.0944	0.0117	0.0048	0.0038	0.0017
Poison Accuracy	0.1110	0.2160	0.4540	0.9740	0.9970	0.9980	0.9990	1.0000

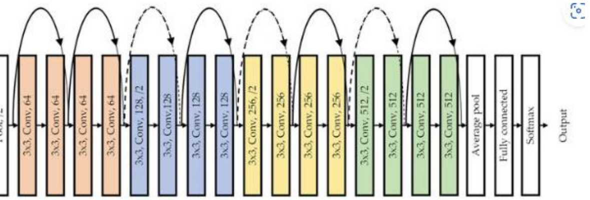
Table 2: Progression of Validation and Poison Metrics at Various Percentages of Epochs with non iid function (Fashion-MNIST)

Percentage Epoch	1% 1st	10% 3rd	20% 6th	50% 15th	75% 23rd	90% 27th	95% 29th	100% 30th
Validation Loss	1.48396	0.60744	0.48642	0.32297	0.25522	0.24123	0.24179	0.24180
Validation Accuracy	0.74670	0.78240	0.81630	0.87730	0.90510	0.91130	0.91330	0.91160
Poison Loss	1.98446	2.44363	2.76638	1.00774	0.08870	0.02909	0.03659	0.01755
Poison Accuracy	0.19300	0.10000	0.16000	0.61400	0.97300	0.98800	0.98700	0.99200

Table 3: Progression of Validation and Poison Metrics at Various Percentages of Epochs with ResNet18 Model and Fashion-MNIST

Percentage Epoch	1% 1st	10% 3rd	20% 6th	50% 15th	75% 23rd	90% 27th	95% 29th	100% 30th
Validation Loss	3.2416	0.4877	0.4017	0.3304	0.3292	0.3261	0.2940	0.2866
Validation Accuracy	0.1930	0.8269	0.8617	0.8853	0.8849	0.8915	0.8971	0.8987
Poison Loss	5.9446	3.6600	3.9430	1.6162	0.7666	0.5250	0.4046	0.1898
Poison Accuracy	0.0000	0.0540	0.0840	0.3510	0.7030	0.8140	0.8870	0.9470

ResNet18



e Resnet-18 Model.

Table 4: Progression of Validation and Poison Metrics at Various Percentages of Epochs with ResNet18 Model and CIFAR-10

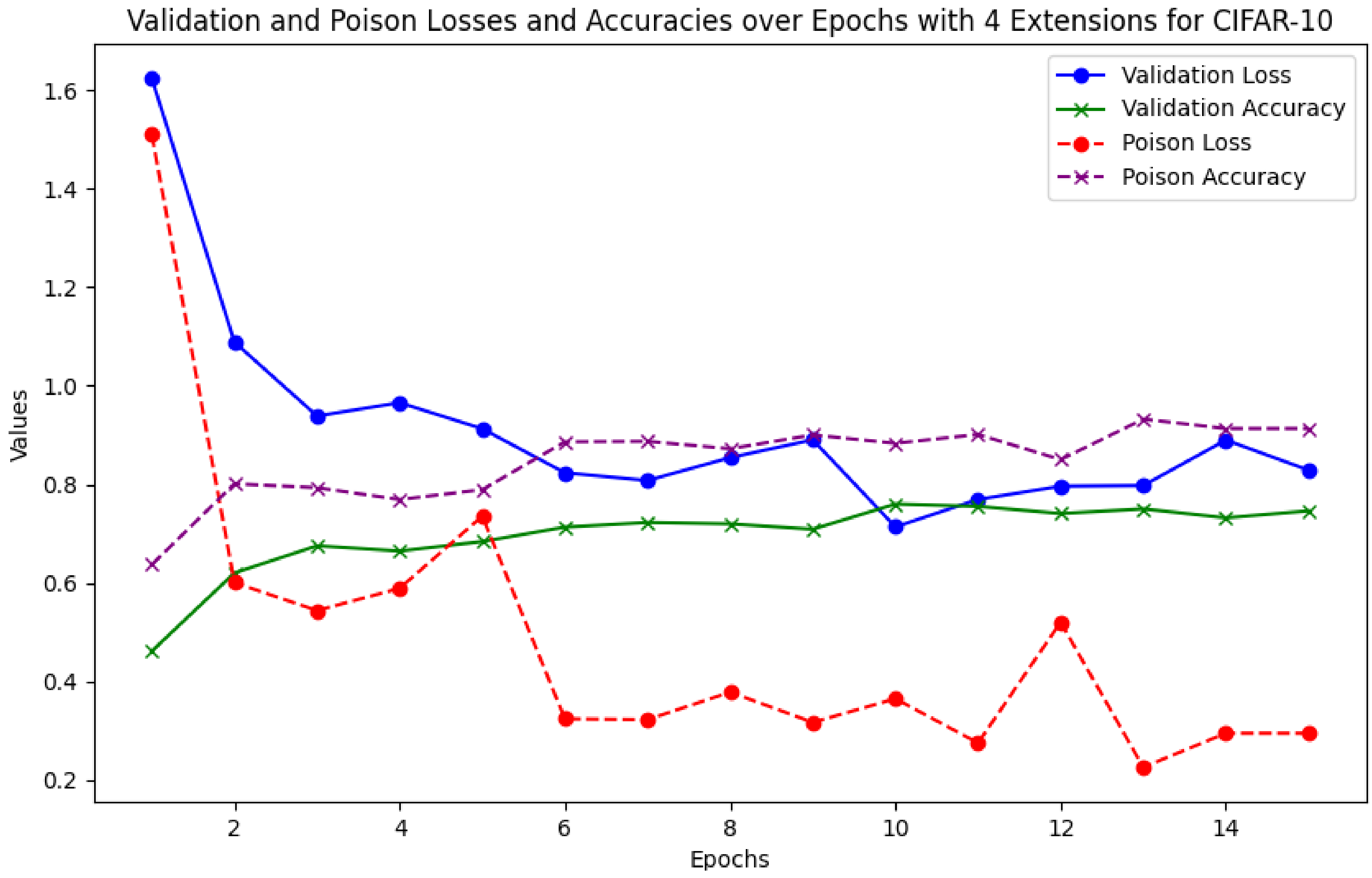
Percentage **Epoch**	**1%** **1st**	**10%** **3rd**	**20%** **6th**	**50%** **15th**	**75%** **23rd**	**90%** **27th**	**95%** **29th**	**100%** **30th**
Validation Loss	2.3418	1.9684	2.8175	1.3384	1.5843	1.6966	1.0783	1.5107
Validation Accuracy	0.1006	0.3038	0.4306	0.5335	0.5457	0.5817	0.6359	0.7311
Poison Loss	1.8851	2.1871	3.0659	2.0318	2.6675	4.6099	3.0589	5.0441
Poison Accuracy	0.9960	0.0280	0.1640	0.2210	0.1850	0.0150	0.1430	0.2720

Table 5: Progression of Validation and Poison Metrics at Various Percentages of Epochs with simple CNN and CIFAR-10

Percentage **Epoch**	**1%** **1st**	**10%** **3rd**	**20%** **6th**	**50%** **15th**	**75%** **23rd**	**90%** **27th**	**95%** **29th**	**100%** **30th**
Validation Loss	2.0965	1.8894	1.6880	1.1264	0.8475	0.7536	0.7111	0.6853
Validation Accuracy	0.1874	0.3095	0.3912	0.6026	0.7117	0.7429	0.7629	0.7698
Poison Loss	2.2606	2.1211	2.0135	2.7687	2.2143	1.8587	1.6772	1.6733
Poison Accuracy	0.0000	0.0590	0.1080	0.0410	0.2290	0.4010	0.5020	0.5170

Table 6: Final Experiment with 4 Extensions: AdamW + Cosine Annealing Learning Rate Scheduler + Early Stopping for Overfit ting Prevention + Batch Normalization (CIFAR-10)

Percentage **Epoch**	**1%** **1st**	**10%** **3rd**	**20%** **6th**	**50%** **15th**	**75%** **23rd**	**90%** **27th**	**95%** **29th**	**100%** **30th**
Validation Loss	1.6236	1.0884	0.9385	0.7135	0.7693	0.7960	0.7974	0.8287
Validation Accuracy	0.4614	0.6202	0.6753	0.7599	0.7552	0.7406	0.7500	0.7459
Poison Loss	1.5096	0.6006	0.5437	0.3649	0.2753	0.5202	0.2260	0.2950
Poison Accuracy	0.6370	0.8010	0.7930	0.8830	0.9010	0.8500	0.9320	0.9130


Validation and Poison Losses and Accuracies over Epochs with 4 Extensions for CIFAR-10
Validation Loss
Validation Accuracy
Poison Loss
Poison Accuracy
Values
Epochs
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
2
4
6
8
10
12
14

Conclusion

- Both CNN and ResNet18 models perform well on Fashion-MNIST, achieving high validation accuracy. On CIFAR-10, ResNet18 struggles more, with slower and less stable progress, possibly due to the complexity of the dataset and the non-IID distribution.

- Non-IID distribution also introduces variability in poison metrics, causing the models to become vulnerable to poison attacks at a slower pace compared to IID data.

- —>AdamW optimizer, Cosine Annealing Learning Rate Scheduler, Early Stopping, and Batch Normalization—demonstrates significant improvements in both validation and poison metrics.

References

- Loshchilov, Ilya, and Frank Hutter. ”Decoupled weight decay regular-ization.” International Conference on Learning Representations (ICLR),2017.
- Loshchilov, Ilya, and Frank Hutter. ”SGDR: Stochastic Gradient Descentwith Warm Restarts.” International Conference on Learning Representa-tions (ICLR), 2016.
- Prechelt, Lutz. ”Early stopping—but when?” In Neural Networks: Tricksof the trade, pp. 55-69. Springer, Berlin, Heidelberg, 1998.
- Ioffe, Sergey, and Christian Szegedy. ”Batch normalization: Acceleratingdeep network training by reducing internal covariate shift.” In Proceedingsof the International Conference on Machine Learning (ICML), pp. 448-456. 2015.