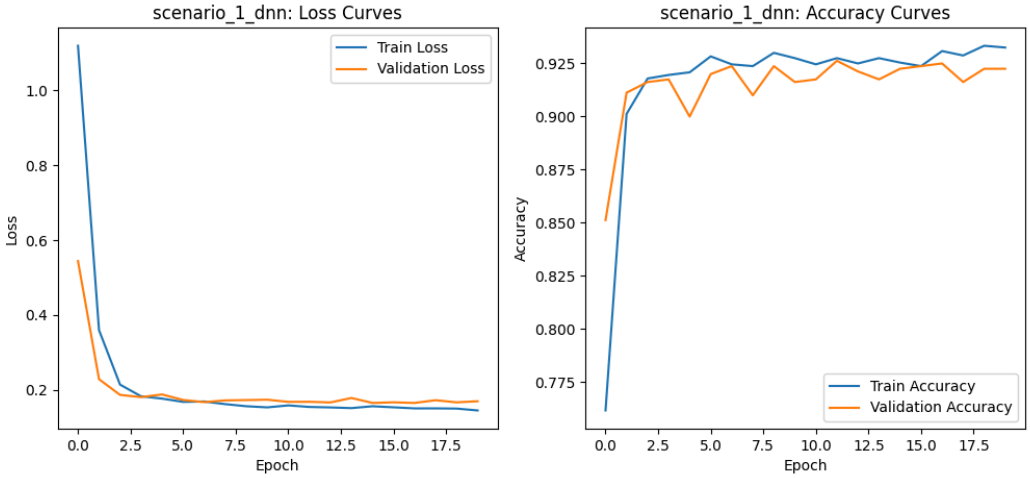
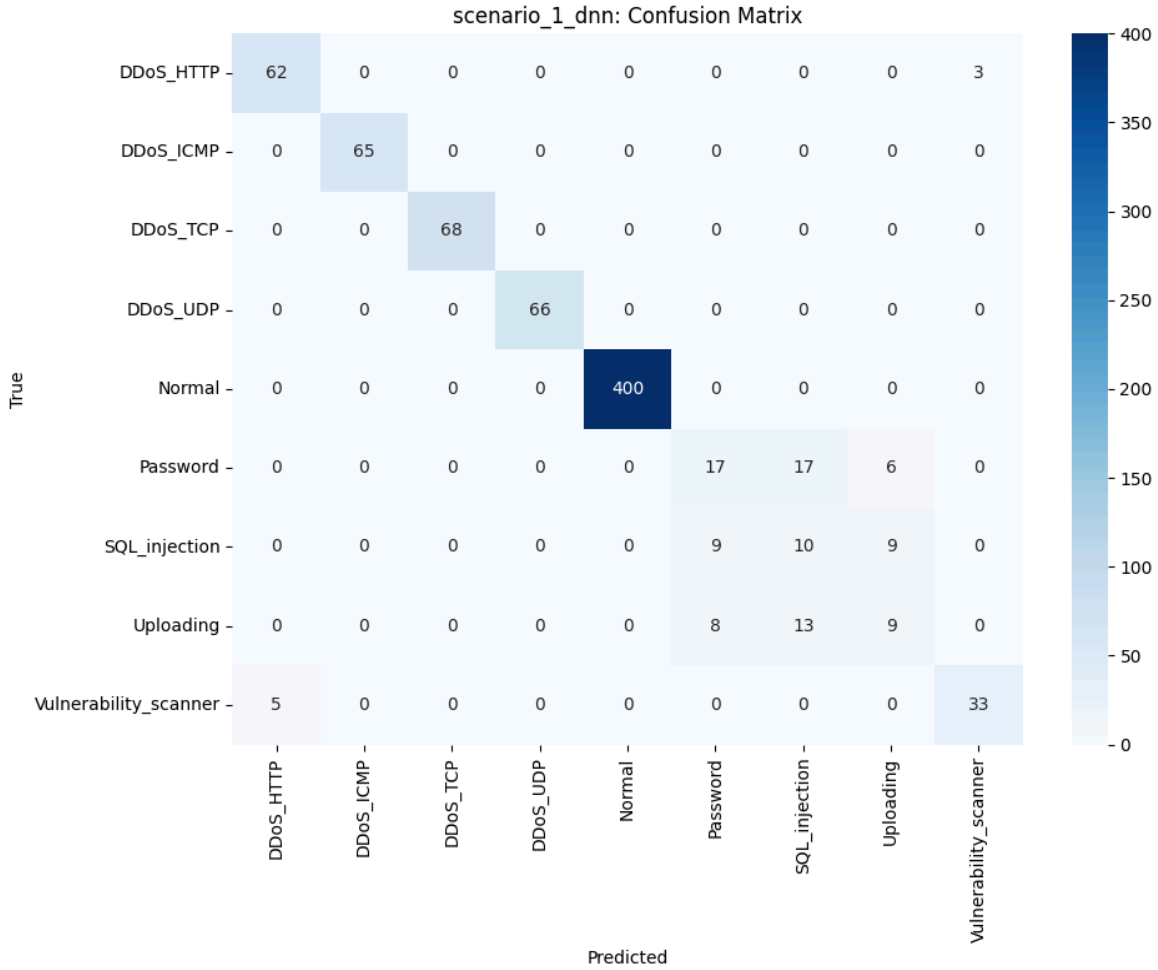
SYSC 4906D
Assignment 6

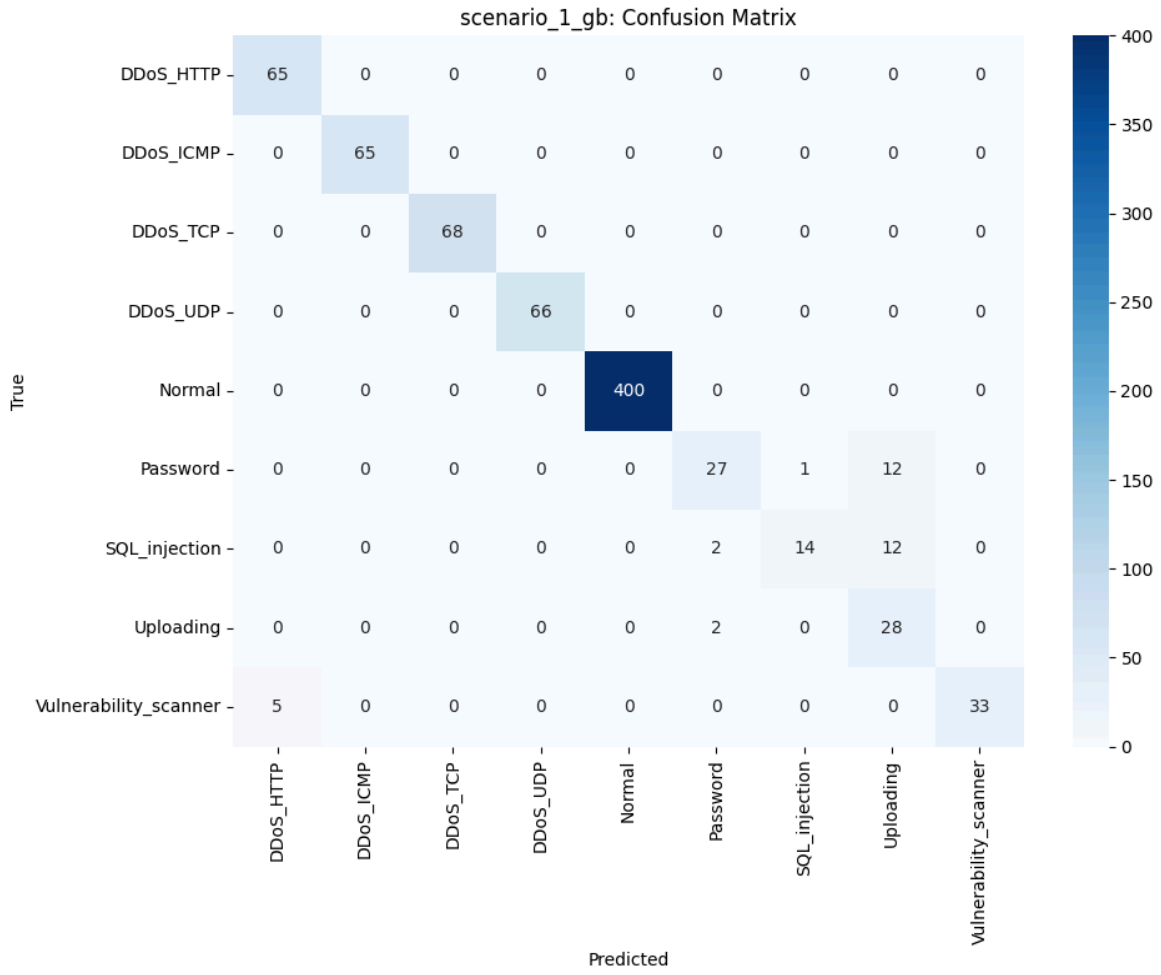
Ben Mostafa 101192825
Alec Tratnik 101220933

Results and Observations

Scenario 1



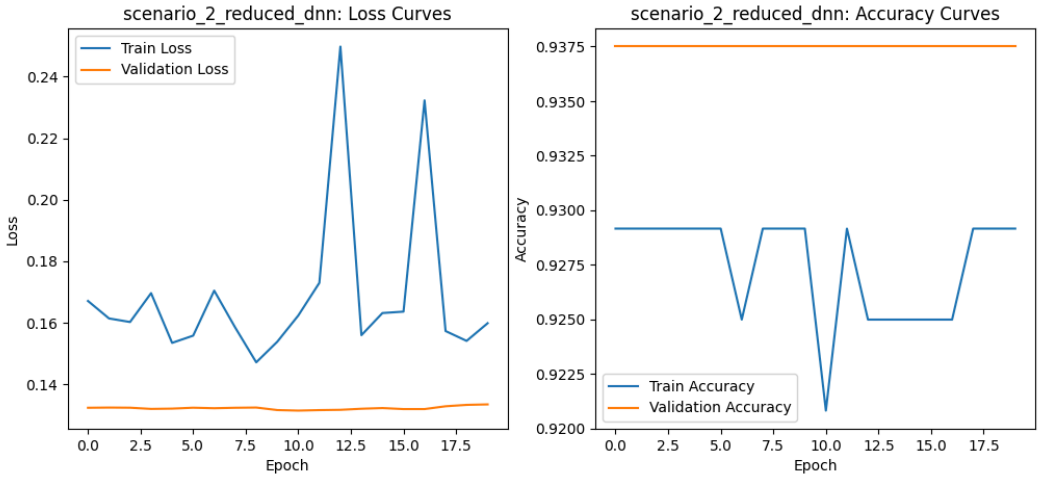


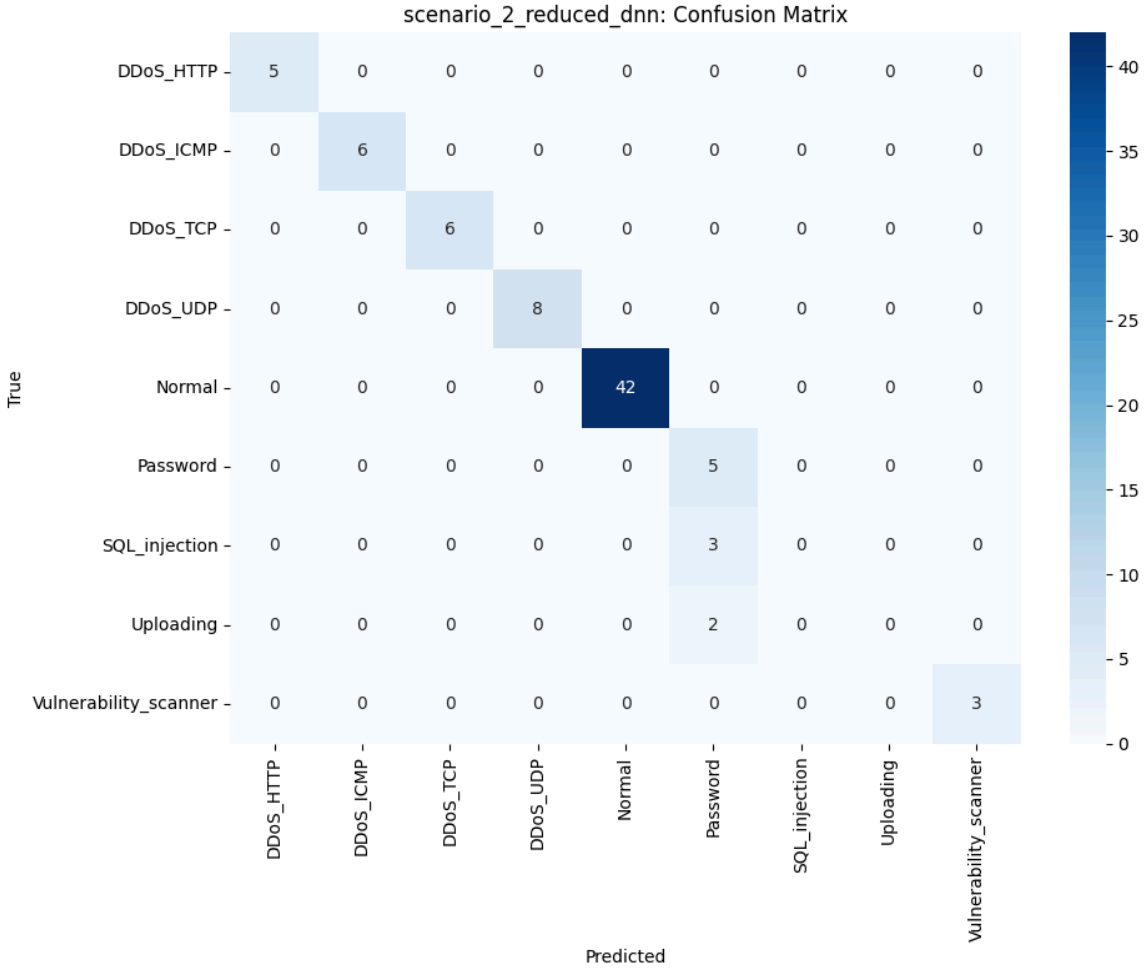


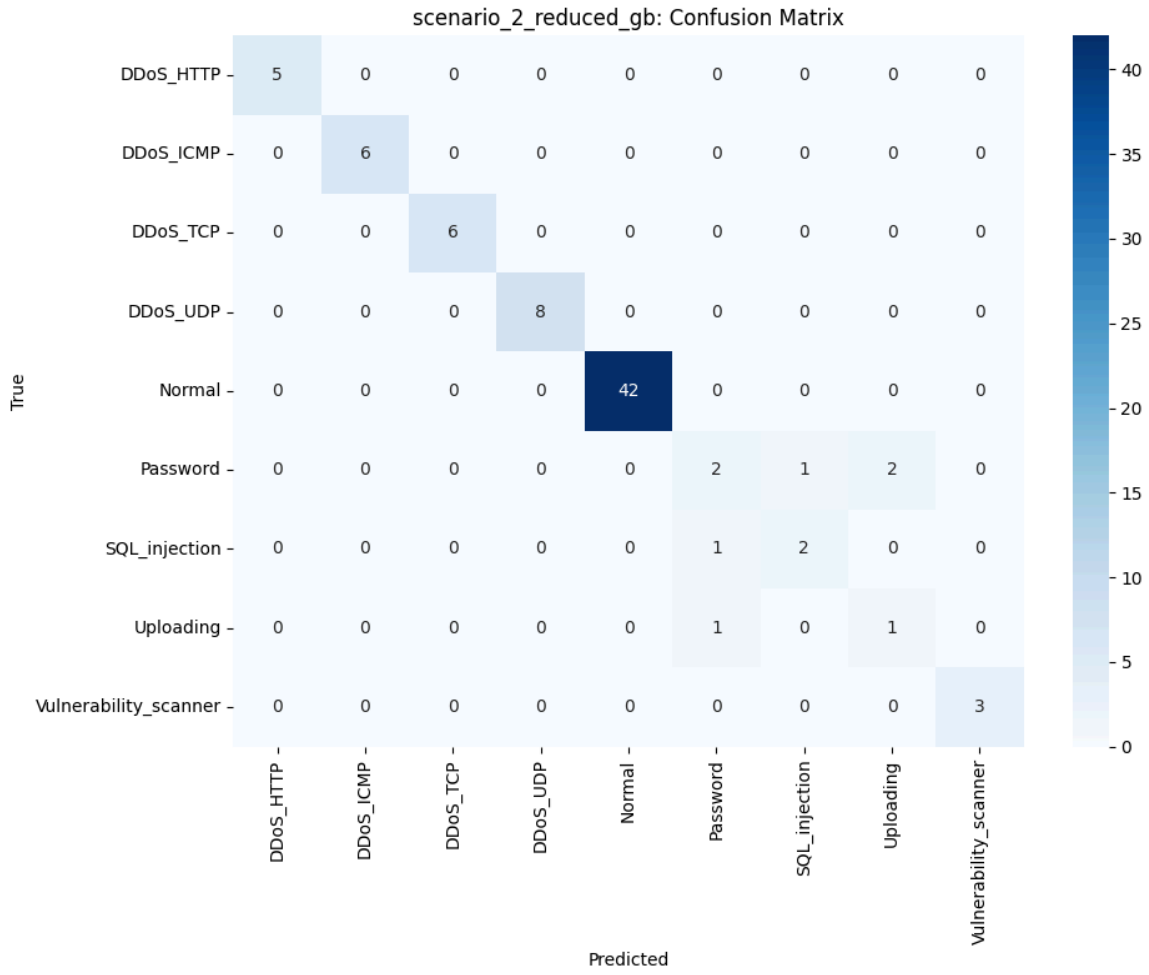
If the DNN loss accuracy curves, the loss stabilizes around 0.2 by epoch 5, with consistent training and validation trends. The accuracy also reaches ~92.5%, showing minimal overfitting and good generalization.
In the DNN confusion matrix normal traffic is perfectly classified, DDoS types perform well, but Password, SQL injection, and Uploading attacks show some confusion. Also, Vulnerability_scanner has minor misclassifications as DDoS_HTTP.
The GB confusion matrix shows similar behavior to DNN for normal and DDosS traffic. GB appears to better handle Password (27 vs. 17), SQL injection (14 vs. 9), and Uploading (28 vs. 13).
Overall, GB performs slightly better on minority classes.

Scenario 2

1.


scenario_2_reduced_dnn: Loss Curves
Train Loss
Validation Loss
Loss
Epoch
scenario_2_reduced_dnn: Accuracy Curves
Train Accuracy
Validation Accuracy
Accuracy
Epoch


scenario_2_reduced_dnn: Confusion Matrix
DDoS_HTTP
DDoS_ICMP
DDoS_TCP
DDoS_UDP
Normal
Password
SQL_injection
Uploading
Vulnerability_scanner
True
Predicted

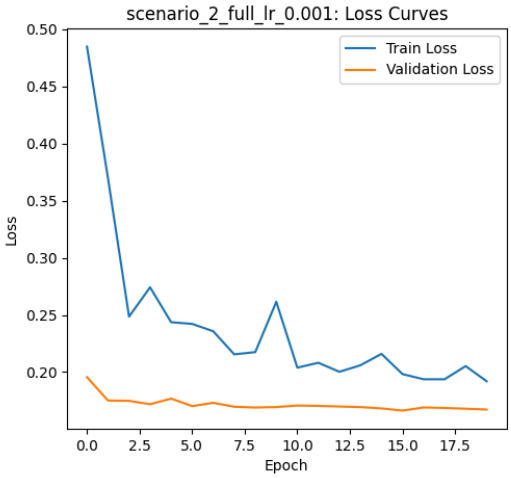


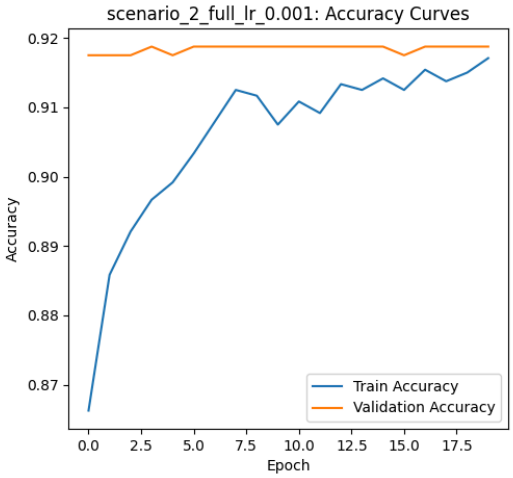
For the DNN performance, the loss curves show instability, with frequent spikes, indicating difficulty in convergence due to less data. The accuracy also fluctuates more compared to Scenario 1, stabilizing around ~92.5%, but with less consistency. The confusion matrix for DNN shows reduced performance on minority classes like Password and SQL injection.
As for the GB performance, the confusion matrix shows GB handles reduced data slightly better than DNN, with fewer misclassifications for Password and SQL injection. Minority class predictions still degrade compared to the full dataset in Scenario 1.
Overall reducing the dataset size increases instability in training especially for DNN and reduces performance on minority classes for both models. GB appears more robust to the reduced dataset size.

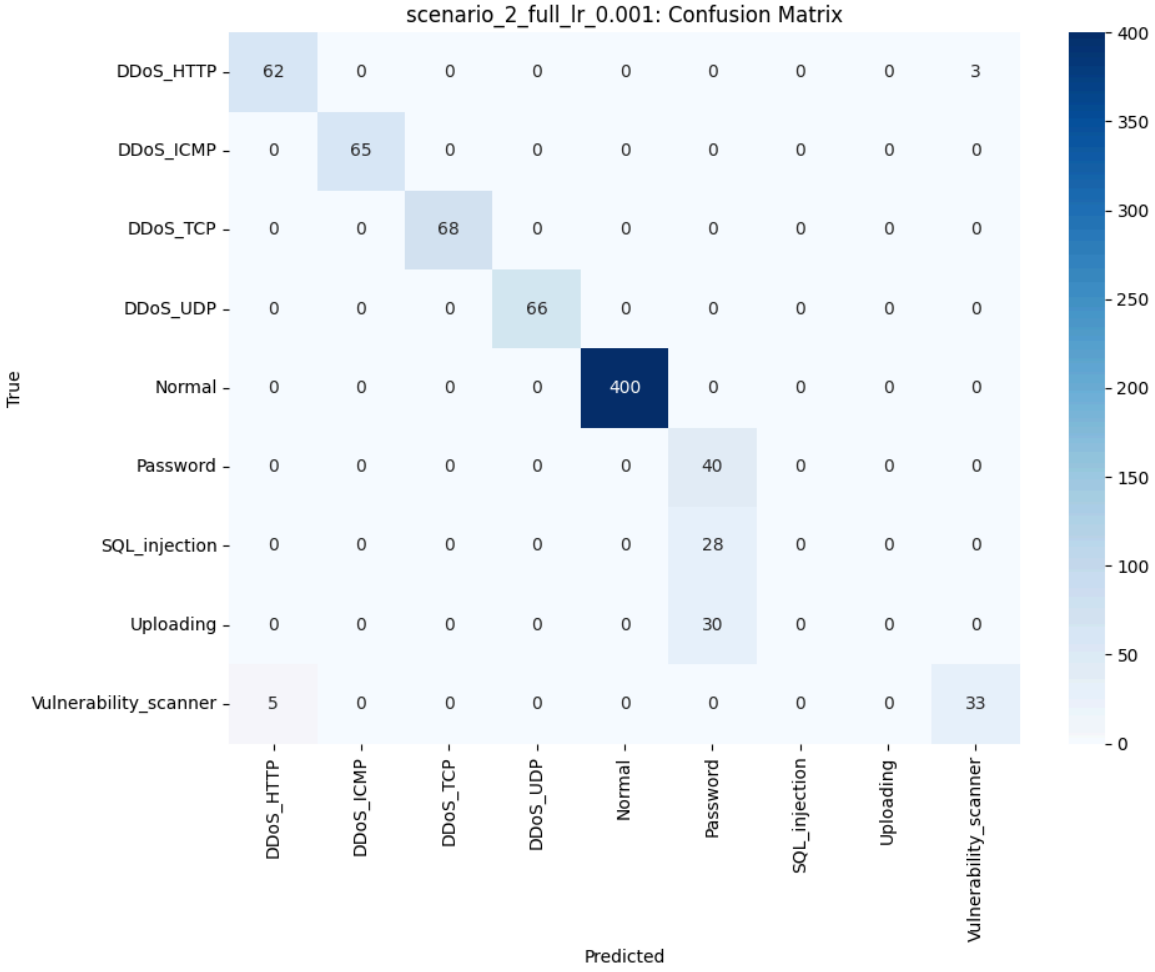
2.

a.

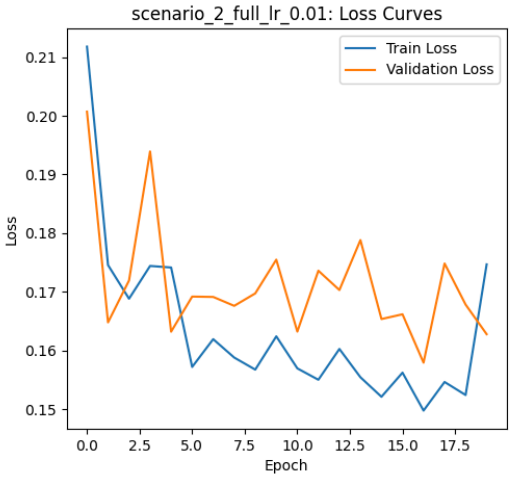
i. Learning Rate: 0.001

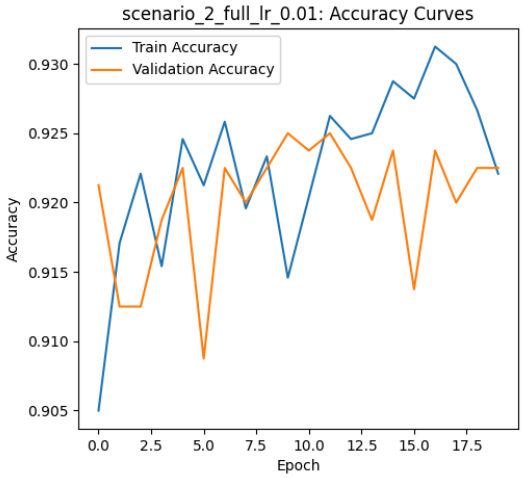


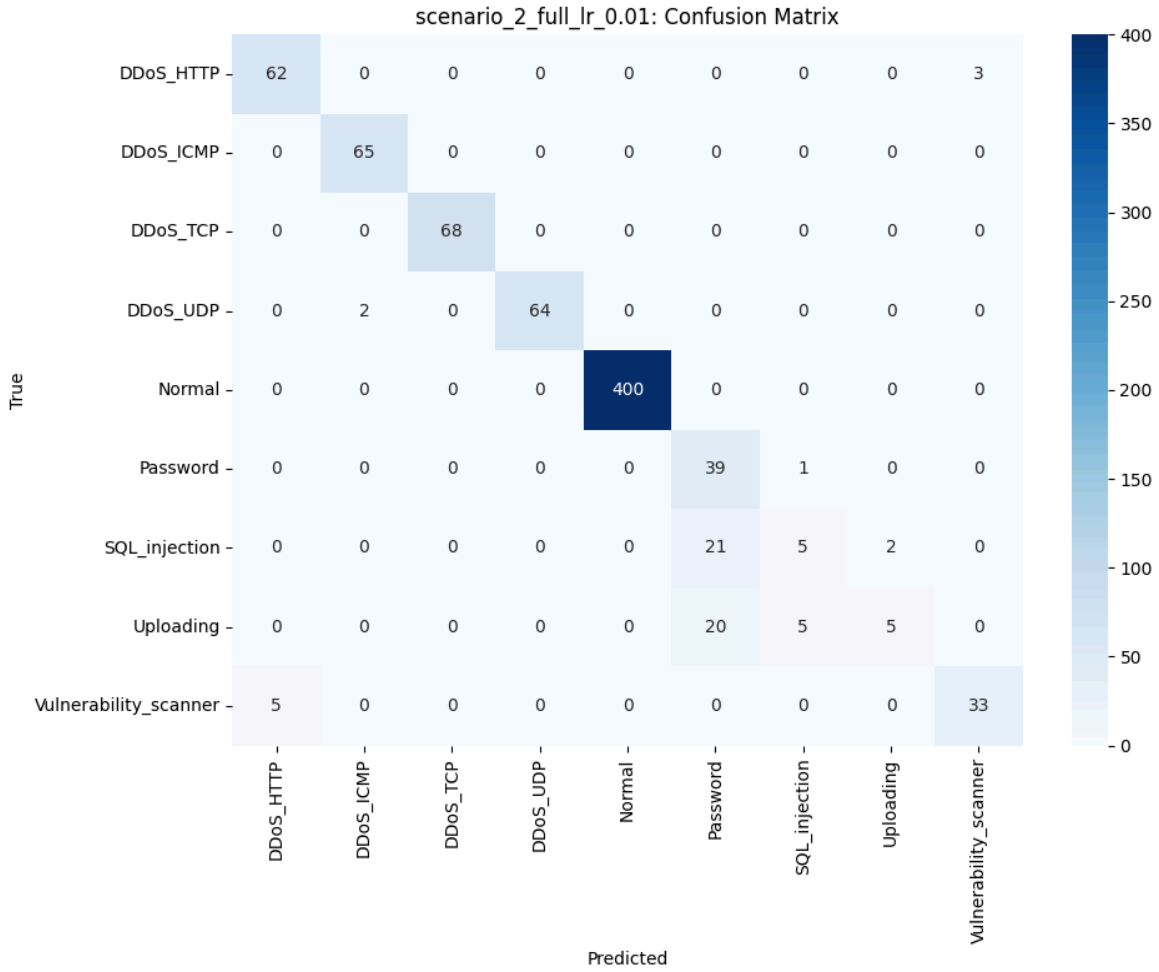




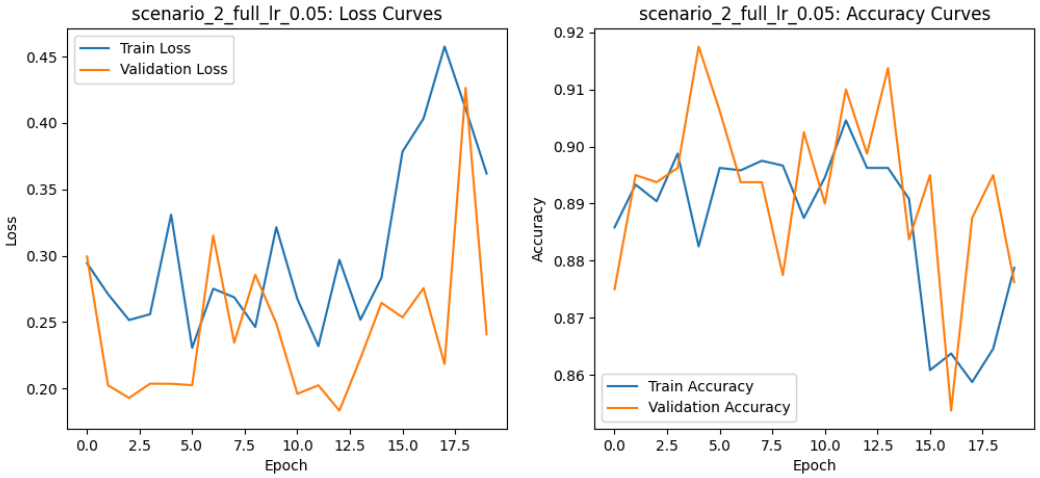
ii. Learning Rate: 0.01

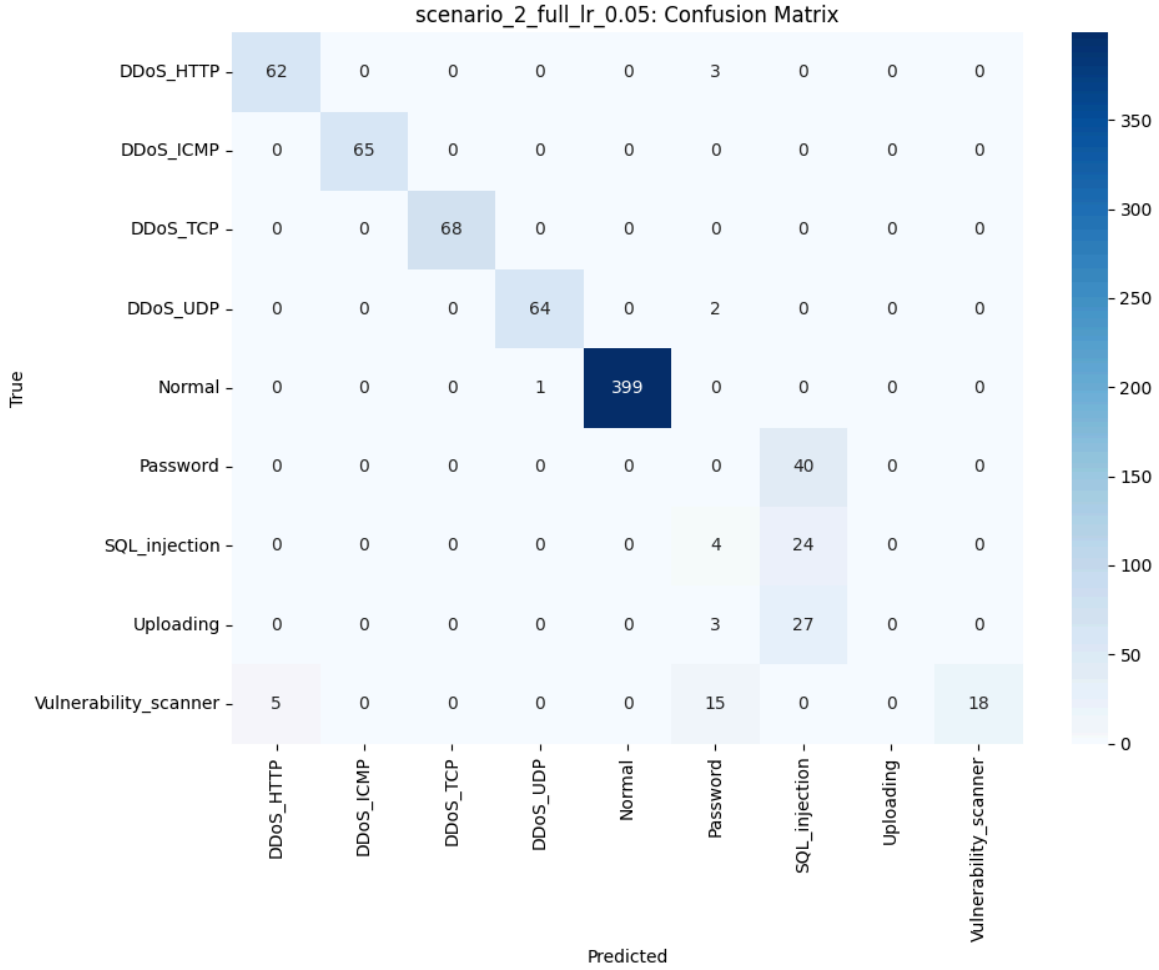






iii. Learning Rate: 0.05



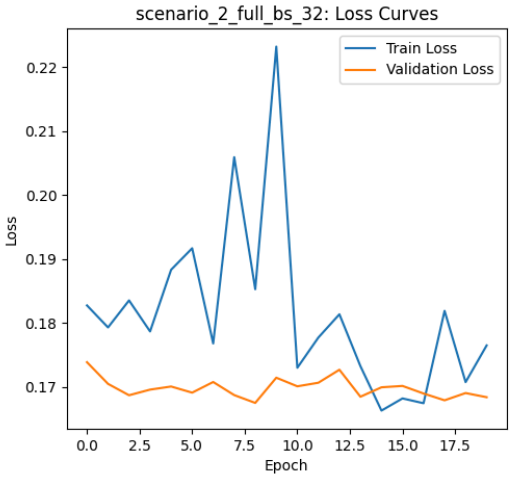


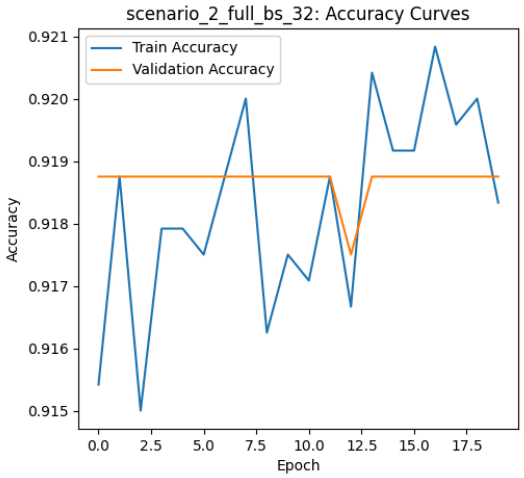
Lower learning rates result in smoother convergence, stable loss reduction, and better generalization, with strong performance across all classes, especially minority ones. Moderate learning rates (0.01) show slight fluctuations in loss and accuracy but maintain decent performance, though minority class predictions degrade slightly. Higher learning rates (0.05)

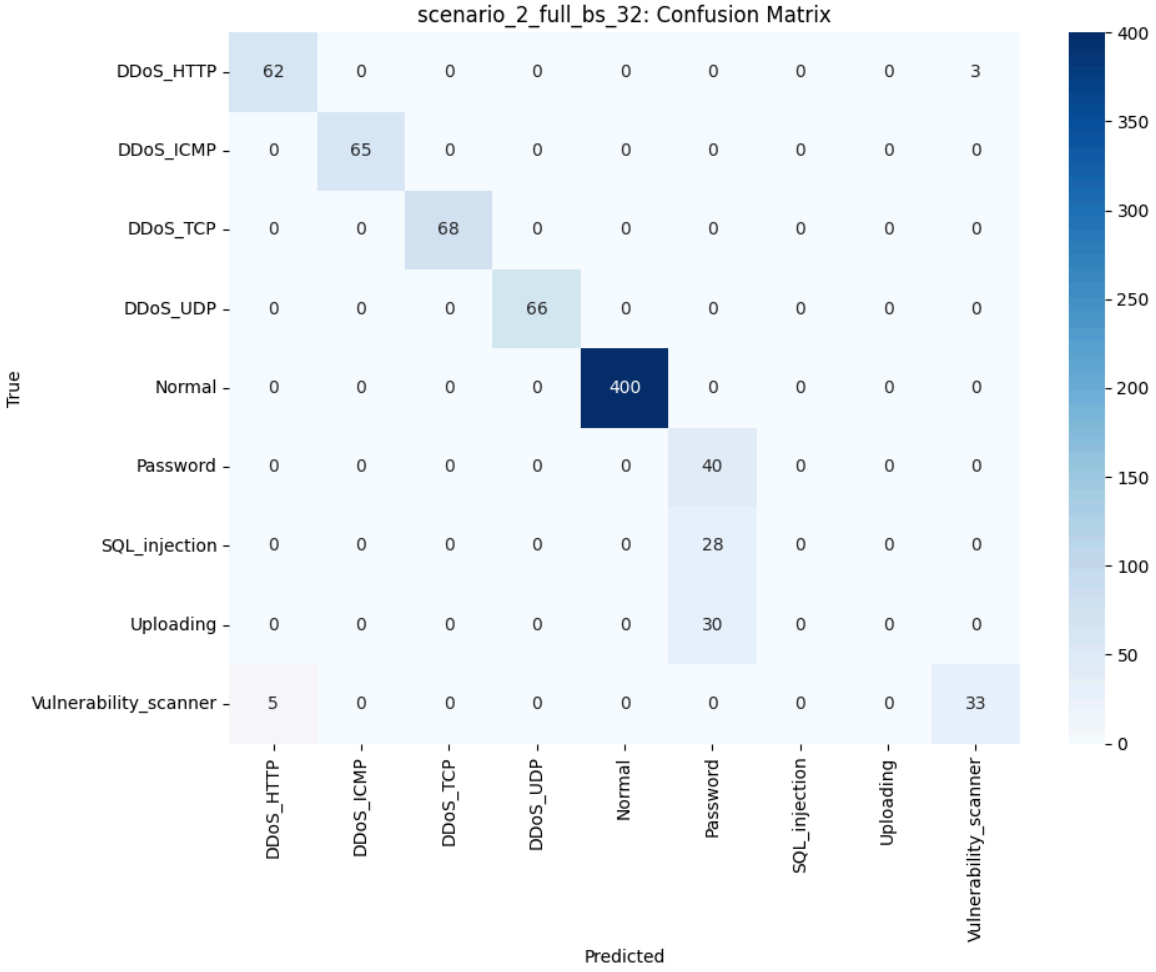
cause significant instability in training, with spikes in loss and declining accuracy after early epochs, leading to poor classification of minority classes like SQL injection and Vulnerability_scanner. Overall, lower learning rates are more reliable for consistent model performance.

b.

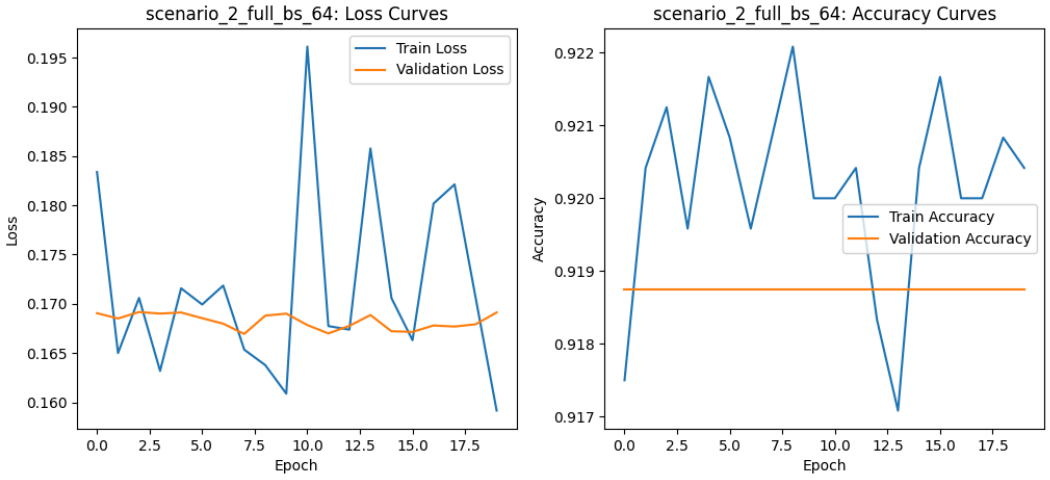
i. Batch Size: 32

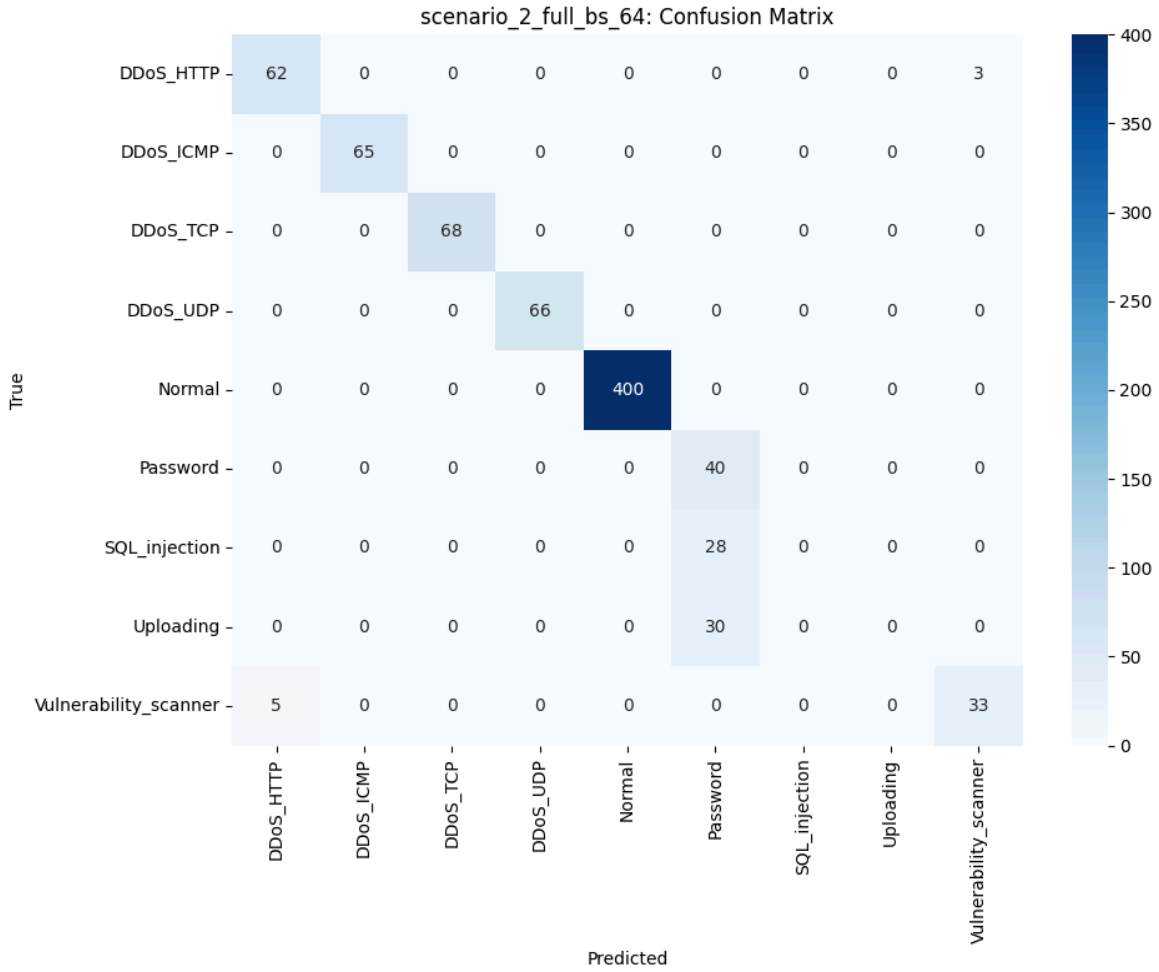






ii. Batch Size: 64



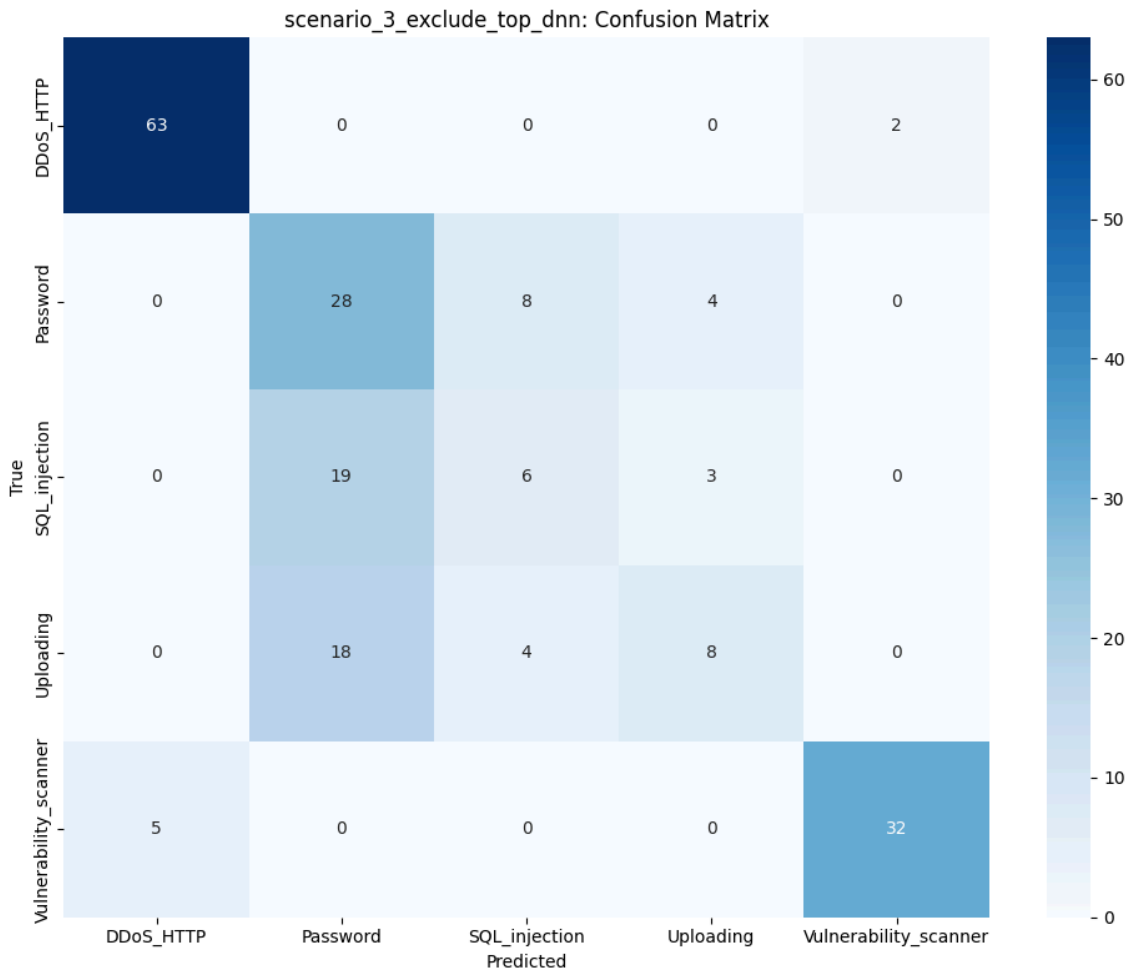


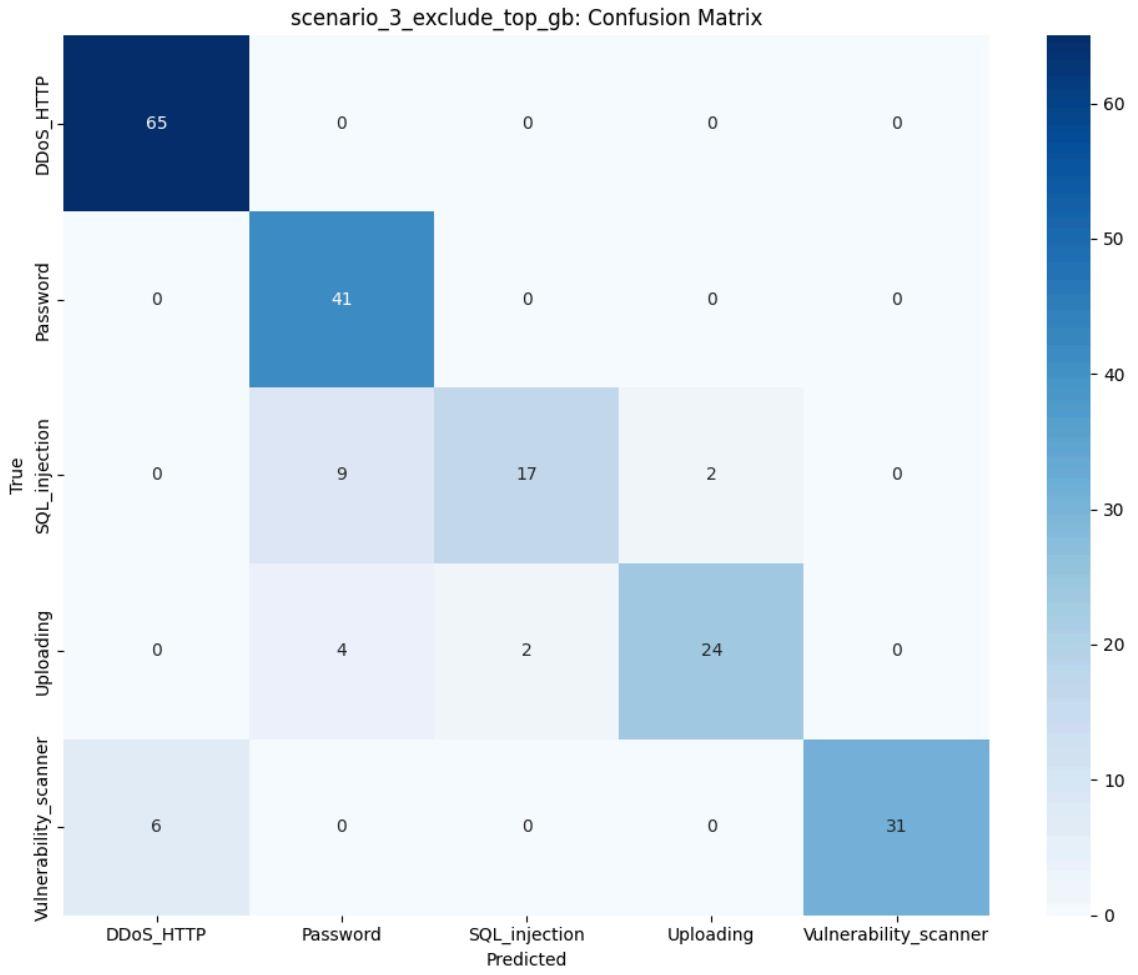
Comparing batch sizes of 32 and 64 reveals interesting learning dynamics. The larger batch size (64) produces slightly smoother training curves overall, though both still exhibit fluctuations. Batch size 64 maintains slightly lower loss values (around 0.16-0.19) compared to batch size 32 (0.17-0.22), suggesting more efficient parameter updates with larger batches. Accuracy curves

show similar patterns of fluctuation between both configurations, with batch size 64 offering slightly more stability in training accuracy. Notably, the confusion matrices are nearly identical between both batch sizes, indicating that final classification performance remains consistent regardless of batch size choice. Both configurations maintain stable validation metrics, suggesting good generalization, with the primary difference being in training dynamics rather than end performance.

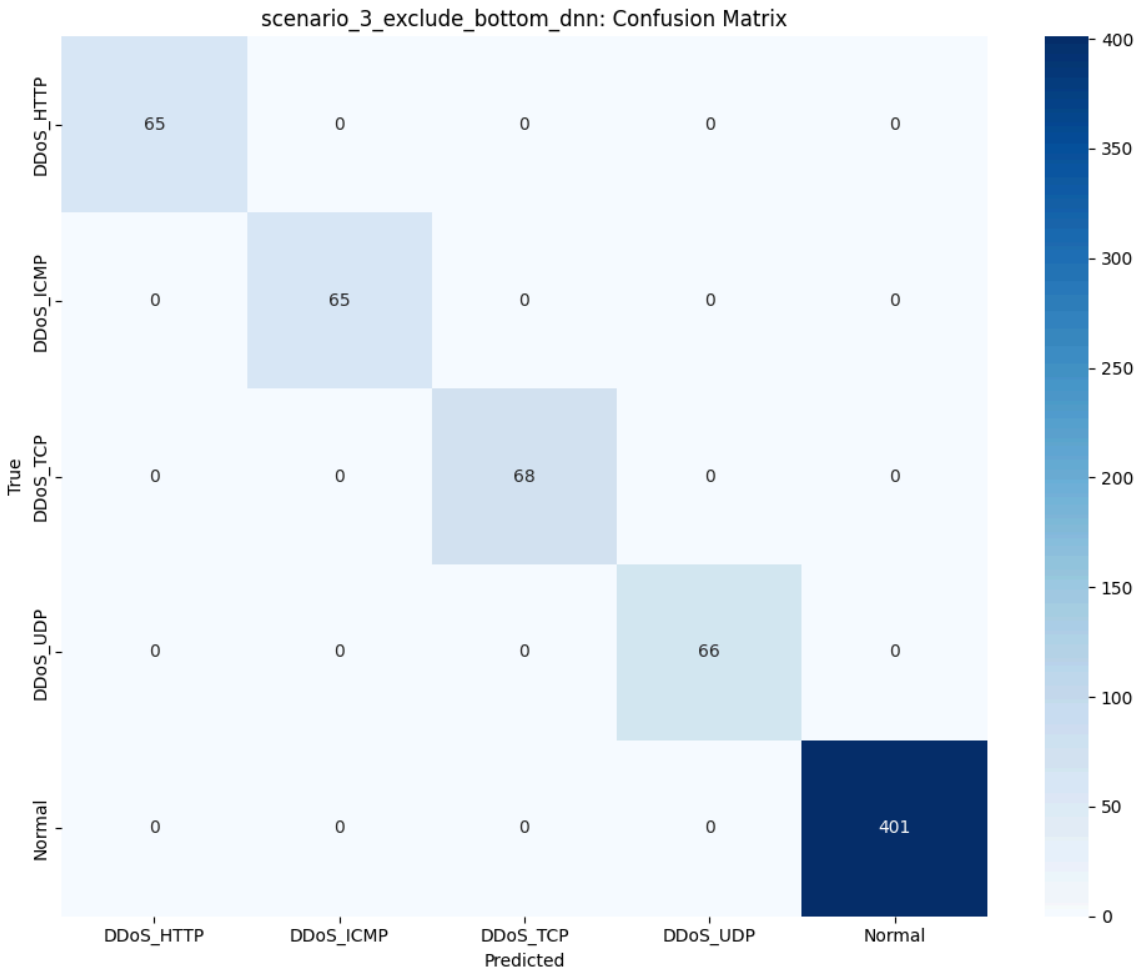
Scenario 3

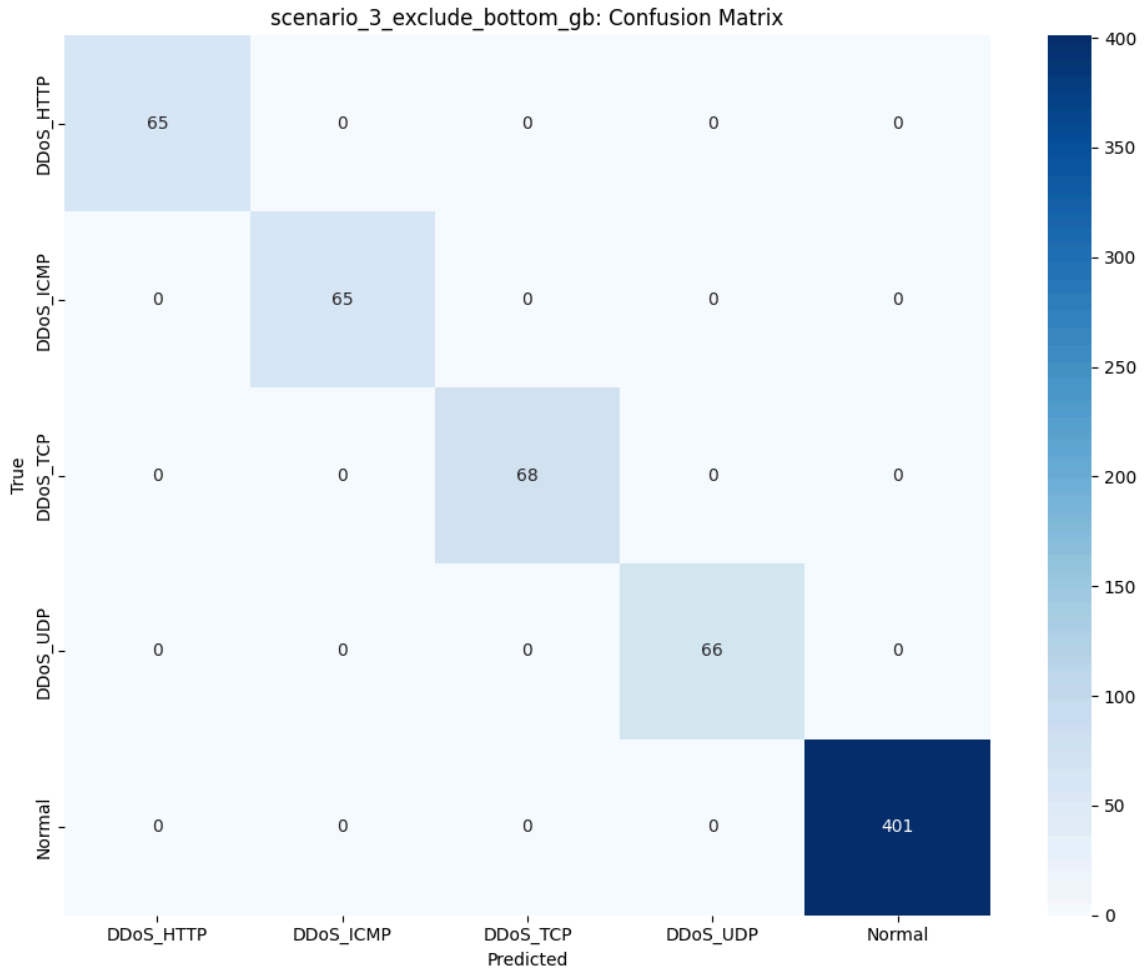
1.




scenario_3_exclude_top_gb: Confusion Matrix
True
DDoS_HTTP
Password
SQL_injection
Uploading
Vulnerability_scanner
65 0 0 0 0
0 41 0 0 0
0 9 17 2 0
0 4 2 24 0
6 0 0 0 31
DDoS_HTTP
Password
SQL_injection
Uploading
Vulnerability_scanner
Predicted
60
50
40
30
20
10
0

2.

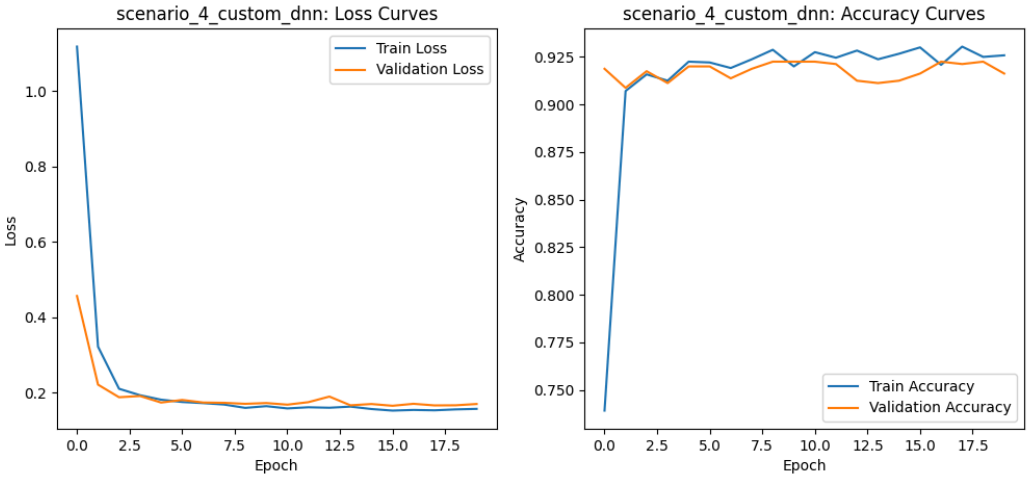

scenario_3_exclude_bottom_dnn: Confusion Matrix
True
DDoS_HTTP
DDoS_ICMP
DDoS_TCP
DDoS_UDP
Normal
65 0 0 0 0
0 65 0 0 0
0 0 68 0 0
0 0 0 66 0
0 0 0 0 401
DDoS_HTTP
DDoS_ICMP
DDoS_TCP
DDoS_UDP
Normal
Predicted
400
350
300
250
200
150
100
50
0

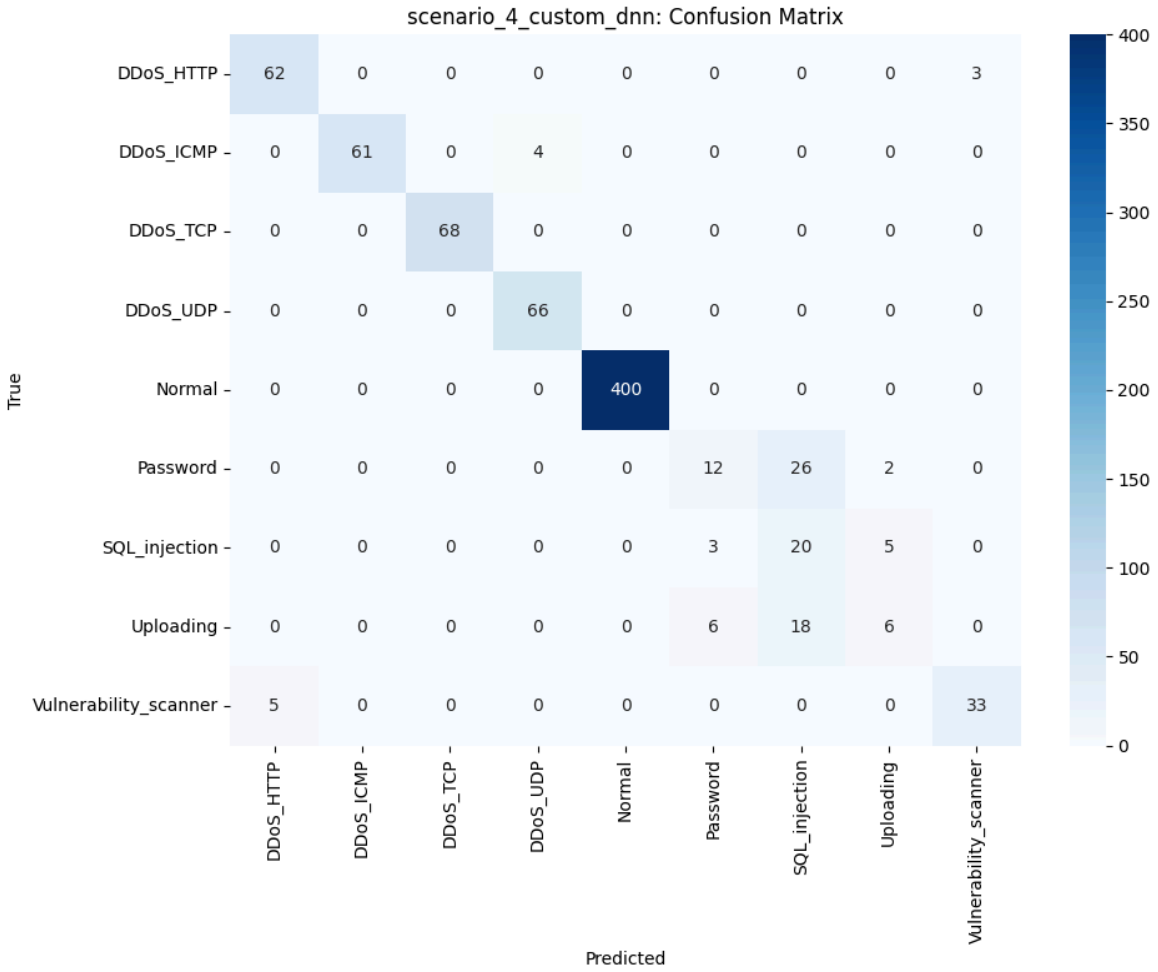


Excluding the top attack types significantly improves classification for Password and Uploading classes in both DNN and Gradient Boosting models compared to Scenario 1, with fewer misclassifications. However, SQL injection performance drops noticeably, showing increased confusion with other classes. Gradient Boosting continues to outperform DNN on minority classes, maintaining better overall balance and accuracy.

Scenario 4

1.

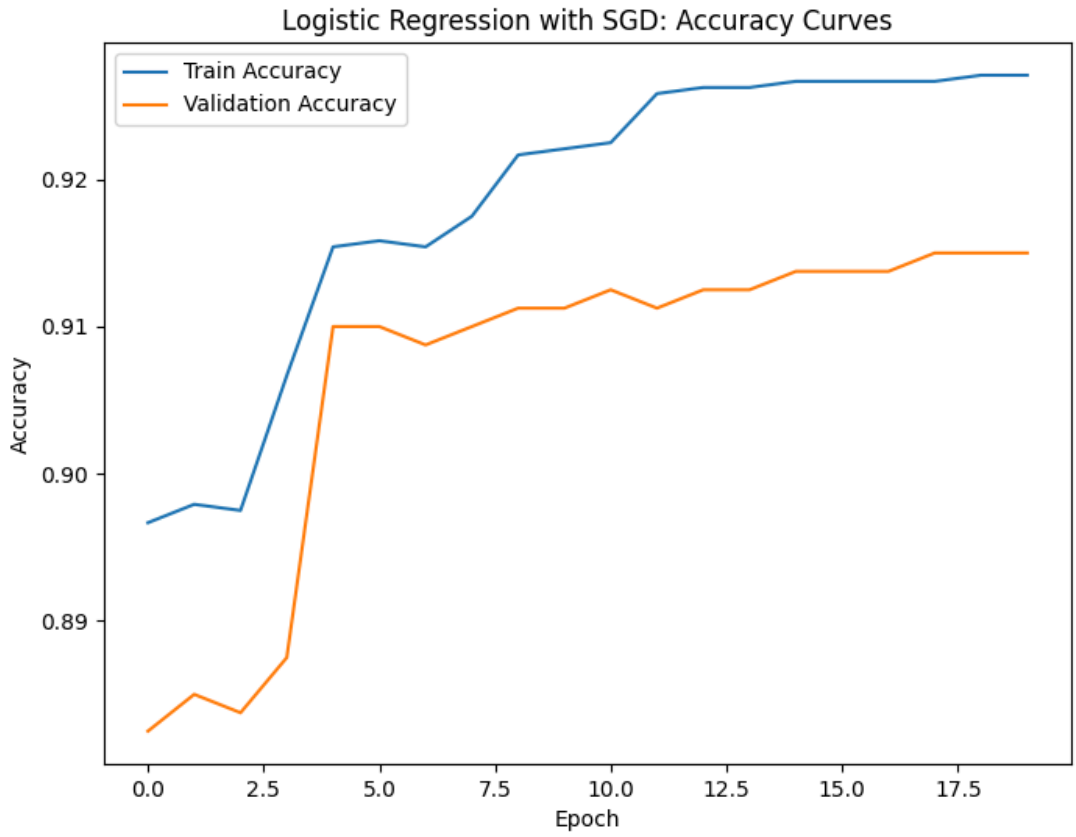




The custom DNN in Scenario 4 shows slightly improved performance compared to the original DNN in Scenario 1. Loss and accuracy curves indicate faster convergence and smoother training dynamics, with validation accuracy stabilizing around 92.5%. The confusion matrix

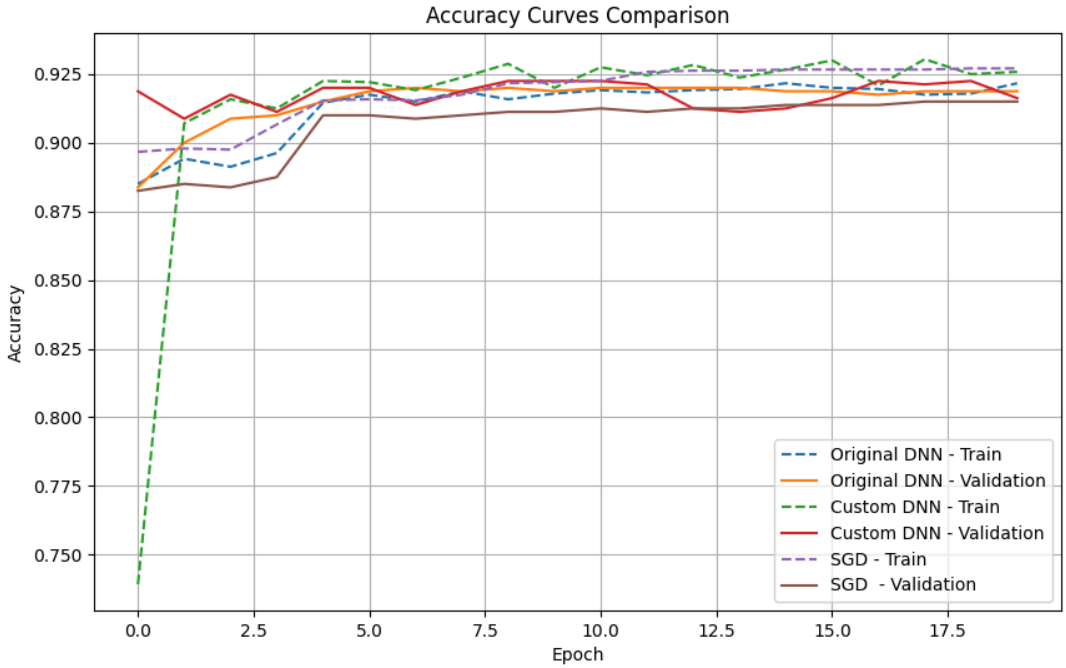
highlights better classification for minority classes like Password (12 vs. 9) and SQL injection (20 vs. 14), though some misclassifications persist for Vulnerability_scanner. Overall, the custom architecture enhances learning stability and improves minority class performance.

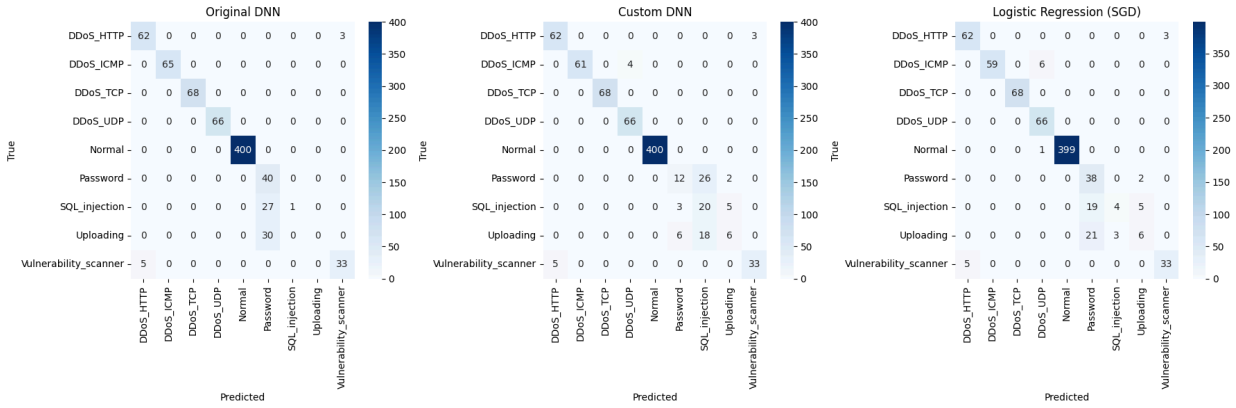
2.




scenario_4_sgd: Confusion Matrix
DDoS_HTTP
DDoS_ICMP
DDoS_TCP
DDoS_UDP
Normal
Password
SQL_injection
Uploading
Vulnerability_scanner
True
Predicted
62 0 0 0 0 0 0 0 3
0 59 0 6 0 0 0 0 0
0 0 68 0 0 0 0 0 0
0 0 0 66 0 0 0 0 0
0 0 0 1 399 0 0 0 0
0 0 0 0 0 38 0 2 0
0 0 0 0 0 19 4 5 0
0 0 0 0 0 21 3 6 0
5 0 0 0 0 0 0 0 33
350
300
250
200
150
100
50
0

3.





For the accuracy curves, the custom DNN achieves the highest peak training accuracy (~92.7%) compared to original DNN (~92.3%) and SGD (~92.5%). However, it shows the most dramatic initial fluctuation, starting at ~73% before rapidly improving. The original DNN demonstrates the most consistent learning pattern with steadily increasing curves. SGD shows moderate performance with slight fluctuations in validation accuracy.

For the confusion matrices, the original DNN performs perfect for normal traffic (400/400), strong on DDoS variants, and reasonable on minority classes (Password: 40/40, SQL_injection: 27/28, Uploading: 30/30). The custom DNN also perfectly identifies normal traffic, but shows notable degradation in password classification (12 vs 40 correct) and increased confusion with SQL_injection. However, it maintains comparable performance on DDoS variants and

Vulnerability_scanner. SGD has competitive performance on password classification (38 correct) and very slightly weaker normal traffic detection (399/400) and moderate SQL_injection performance (19 correct).

Overall the best performing model is the original DNN model. It demonstrates the best overall performance with optimal balance between accuracy and generalization. While custom DNN achieves marginally higher peak accuracy, it shows inconsistent performance across classes and potential overfitting (larger gap between training/validation curves). The SGD model performs surprisingly well but lacks the classification consistency of the original DNN.
The original DNN's superior generalization ability, balanced class performance, and stable learning curve make it the most suitable for this multi-class network traffic classification task, especially considering the importance of correctly identifying all attack types in a cybersecurity context.