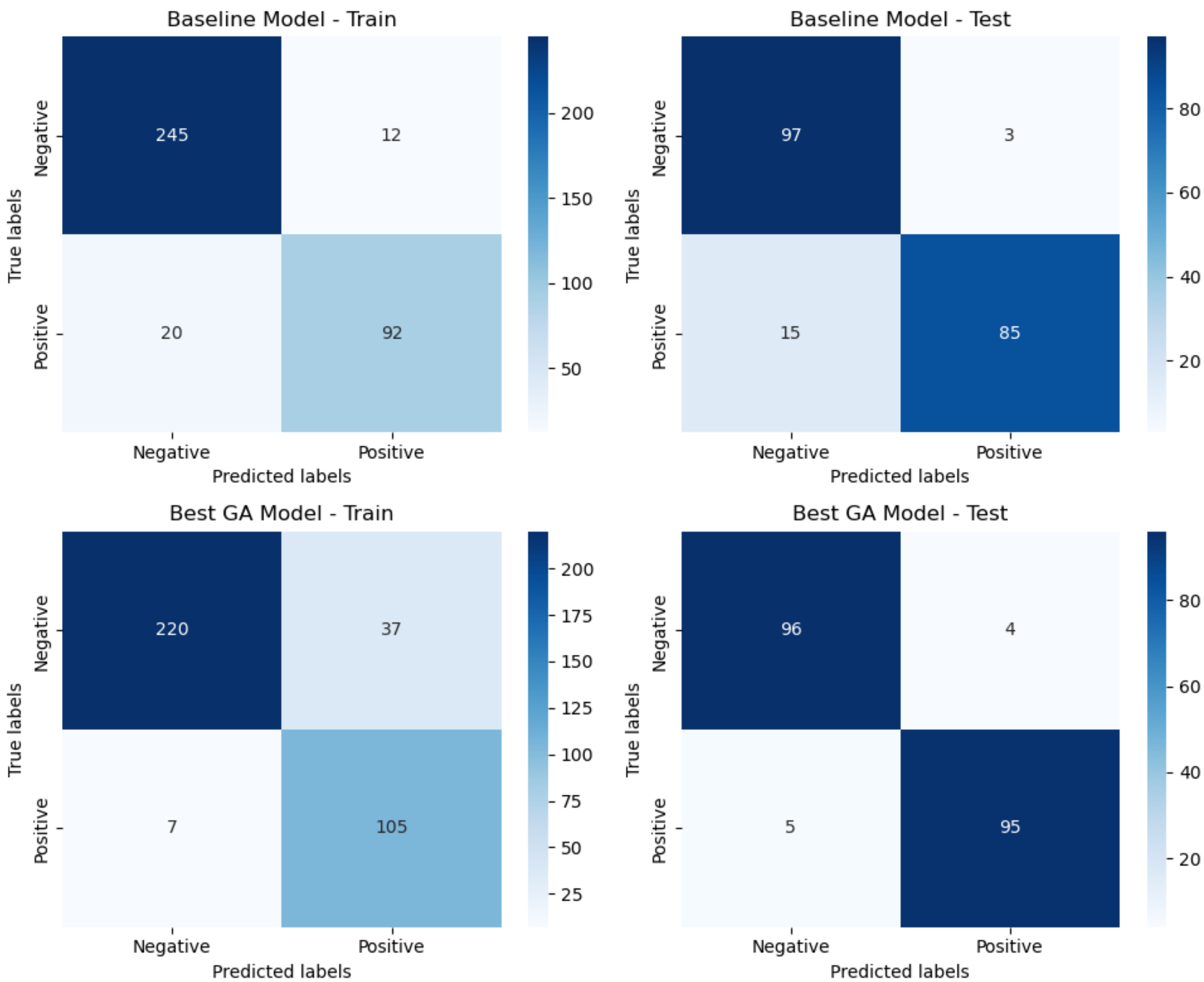
The objective was to optimize a neural network's first layer weights through a genetic algorithm (GA), specifically by masking half of these weights to zeroes while maximizing fitness. A generational GA was employed with **linear ranking selection**, a crossover probability of 0.9, **uniform crossover**, a mutation rate determined by the inverse of the bit string length, and a population size double that of the NN first layer's weights. Individuals were initialized with backpropagation-derived weights, randomly masking 50% as zeroes. The GA's effectiveness was compared against a baseline algorithm focusing on high-magnitude weights.



Confusion Matrices Analysis

- **Baseline Model:** Achieved train accuracy of 91.33% and test accuracy of 91%. The training data confusion matrix showed 245 true negatives, 12 false positives, 20 false negatives, and 92 true positives. For test data, there were 97 true negatives, 3 false positives, 15 false negatives, and 85 true positives.
- **Best GA-Optimized Model:** Showed train accuracy of 88.08% and test accuracy of 95.50%. The training confusion matrix displayed 220 true negatives, 37 false positives, 7 false negatives, and 105 true positives. Test data revealed 96 true negatives, 4 false positives, 5 false negatives, and 95 true positives.

Observations

- The GA-optimized model outperformed the baseline in test accuracy by approximately 4.5%, indicating superior generalization.
- Although the GA model had slightly lower training accuracy, its significantly higher test accuracy suggests a better balance between learning and generalization, reducing overfitting compared to the baseline.
- The increase in false positives for the GA model in training data did not adversely affect its overall test performance, highlighting the effective selection of weights contributing to predictive accuracy.

Conclusion

The GA successfully identified a set of weights that, despite being half-masked, provided improved generalization on unseen data compared to a baseline model that retained all high-magnitude weights. This approach demonstrates the potential of GAs in optimizing neural network architectures for better performance and efficiency.