AI-Powered Intrusion Detection: Performance Analysis

Jonathan Muratalla Isabel Santiago Ahmet Ulusoy

April 24, 2025

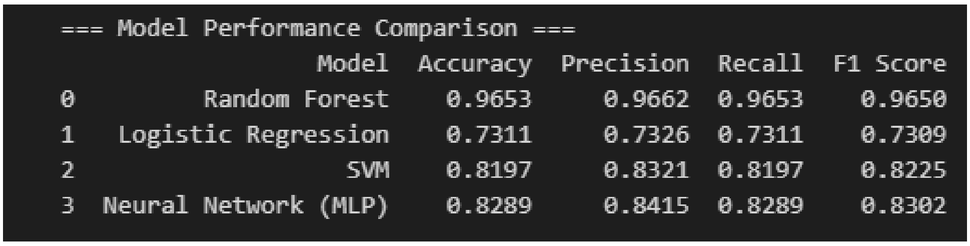
Introduction

In our cybersecurity course project, we aimed to evaluate how well machine learning models could generalize to unseen attack data. We specifically focused on examining whether feature selection (i.e., using only the top N features) had an impact on model performance when facing data not seen during training.

We tested four models: Random Forest, Logistic Regression, Support Vector Machine (SVM), and a Neural Network (MLP). Models were evaluated on both the training dataset and an unseen attack dataset. We experimented by selecting different top N features ($N = 2, 10, 15, 20, 40$, and 50) to analyze the impact.

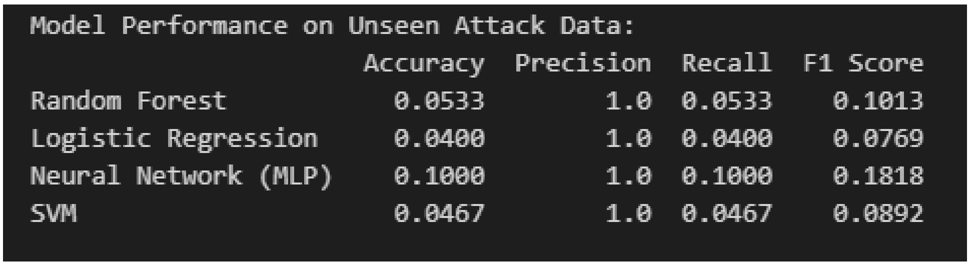
1 Data and Visual Results

Below are screenshots showing model performance comparisons:



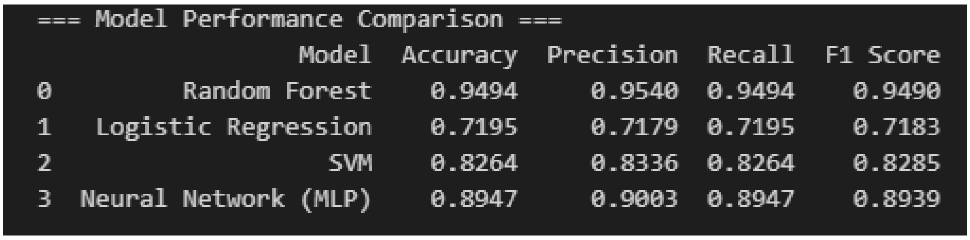
```
=== Model Performance Comparison ===
                 Model  Accuracy  Precision  Recall  F1 Score
0        Random Forest    0.9653     0.9662  0.9653    0.9650
1  Logistic Regression    0.7311     0.7326  0.7311    0.7309
2                  SVM    0.8197     0.8321  0.8197    0.8225
3 Neural Network (MLP)    0.8289     0.8415  0.8289    0.8302
```

Figure 1: Performance comparison of machine learning models across different feature selection sizes (N = 2) - Test Data



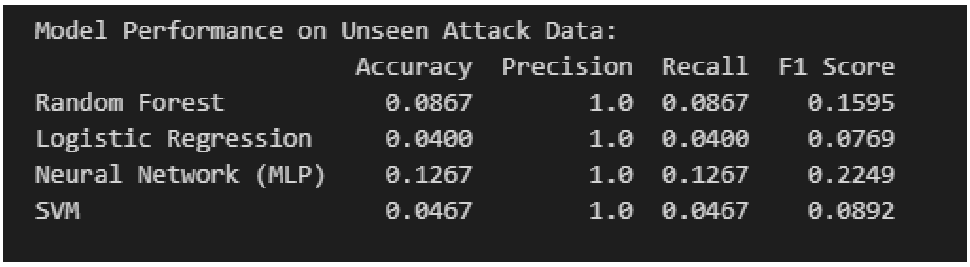
```
Model Performance on Unseen Attack Data:
                      Accuracy  Precision  Recall  F1 Score
Random Forest           0.0533        1.0  0.0533    0.1013
Logistic Regression     0.0400        1.0  0.0400    0.0769
Neural Network (MLP)    0.1000        1.0  0.1000    0.1818
SVM                     0.0467        1.0  0.0467    0.0892
```

Figure 2: Performance comparison of machine learning models across different feature selection sizes (N = 2) - Unseen Data



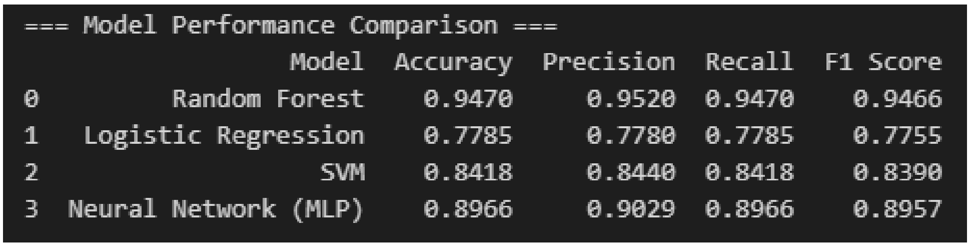
```
=== Model Performance Comparison ===
                 Model  Accuracy  Precision  Recall  F1 Score
0        Random Forest    0.9494     0.9540  0.9494    0.9490
1  Logistic Regression    0.7195     0.7179  0.7195    0.7183
2                  SVM    0.8264     0.8336  0.8264    0.8285
3 Neural Network (MLP)    0.8947     0.9003  0.8947    0.8939
```

Figure 3: Performance comparison of machine learning models across different feature selection sizes (N = 10) - Test Data



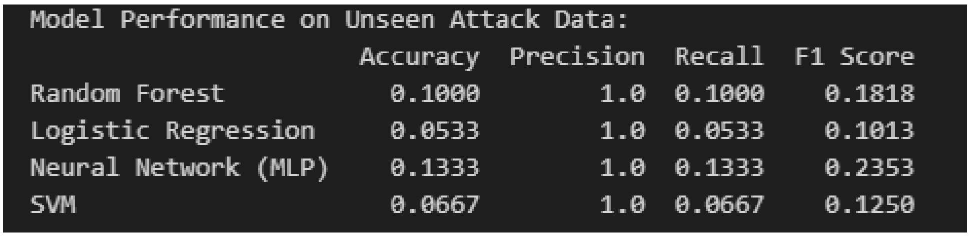
```
Model Performance on Unseen Attack Data:
                      Accuracy  Precision  Recall  F1 Score
Random Forest           0.0867        1.0  0.0867    0.1595
Logistic Regression     0.0400        1.0  0.0400    0.0769
Neural Network (MLP)    0.1267        1.0  0.1267    0.2249
SVM                     0.0467        1.0  0.0467    0.0892
```

Figure 4: Performance comparison of machine learning models across different feature selection sizes (N = 10) - Unseen Data



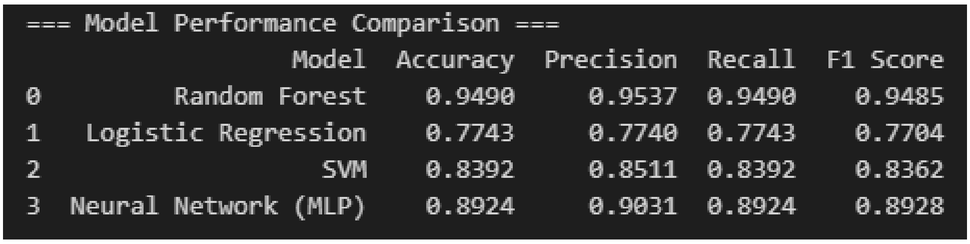
```
=== Model Performance Comparison ===
                  Model  Accuracy  Precision  Recall  F1 Score
0         Random Forest    0.9470     0.9520  0.9470    0.9466
1   Logistic Regression    0.7785     0.7780  0.7785    0.7755
2                   SVM    0.8418     0.8440  0.8418    0.8390
3  Neural Network (MLP)    0.8966     0.9029  0.8966    0.8957
```

Figure 5: Performance comparison of machine learning models across different feature selection sizes (N = 15) - Test Data



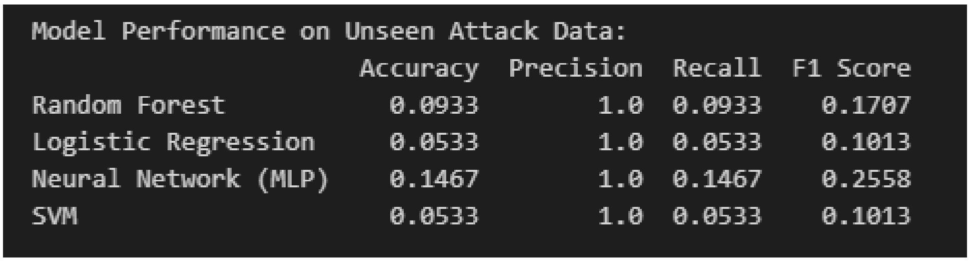
```
Model Performance on Unseen Attack Data:
                      Accuracy  Precision  Recall  F1 Score
Random Forest           0.1000        1.0  0.1000    0.1818
Logistic Regression     0.0533        1.0  0.0533    0.1013
Neural Network (MLP)    0.1333        1.0  0.1333    0.2353
SVM                     0.0667        1.0  0.0667    0.1250
```

Figure 6: Performance comparison of machine learning models across different feature selection sizes (N = 15) - Unseen Data



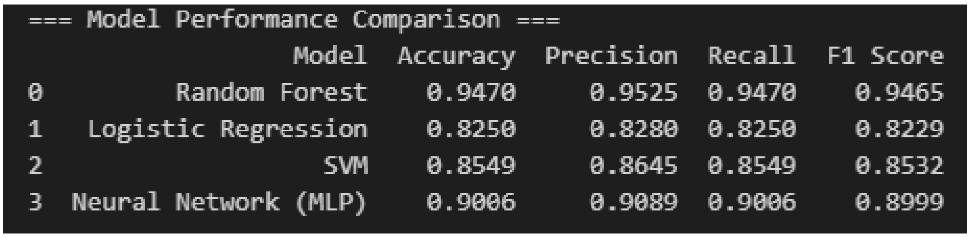
```
=== Model Performance Comparison ===
                 Model  Accuracy  Precision  Recall  F1 Score
0        Random Forest    0.9490     0.9537  0.9490    0.9485
1  Logistic Regression    0.7743     0.7740  0.7743    0.7704
2                  SVM    0.8392     0.8511  0.8392    0.8362
3 Neural Network (MLP)    0.8924     0.9031  0.8924    0.8928
```

Figure 7: Performance comparison of machine learning models across different feature selection sizes (N = 20) - Test Data



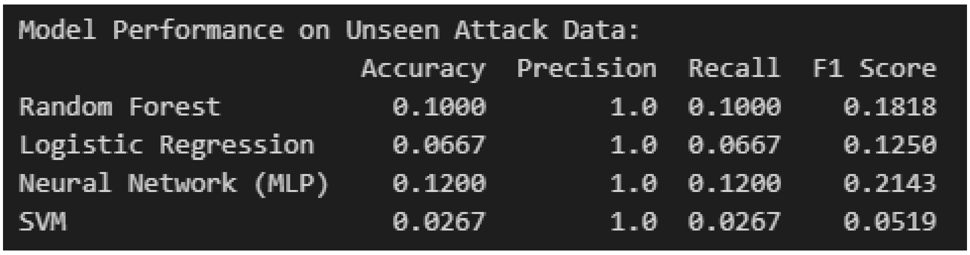
```
Model Performance on Unseen Attack Data:
                      Accuracy  Precision  Recall  F1 Score
Random Forest           0.0933        1.0  0.0933    0.1707
Logistic Regression     0.0533        1.0  0.0533    0.1013
Neural Network (MLP)    0.1467        1.0  0.1467    0.2558
SVM                     0.0533        1.0  0.0533    0.1013
```

Figure 8: Performance comparison of machine learning models across different feature selection sizes (N = 20) - Unseen Data



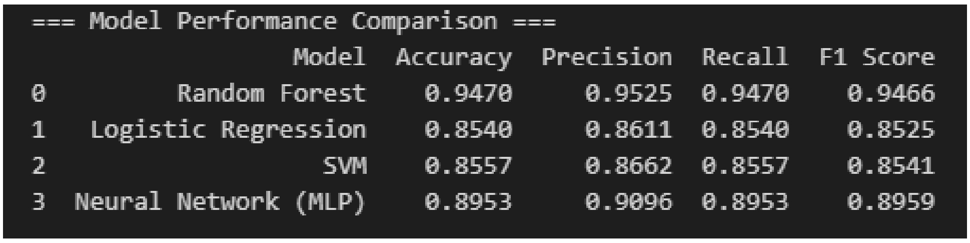
```
=== Model Performance Comparison ===
                 Model  Accuracy  Precision  Recall  F1 Score
0        Random Forest    0.9470     0.9525  0.9470    0.9465
1  Logistic Regression    0.8250     0.8280  0.8250    0.8229
2                  SVM    0.8549     0.8645  0.8549    0.8532
3 Neural Network (MLP)    0.9006     0.9089  0.9006    0.8999
```

Figure 9: Performance comparison of machine learning models across different feature selection sizes (N = 40) - Test Data



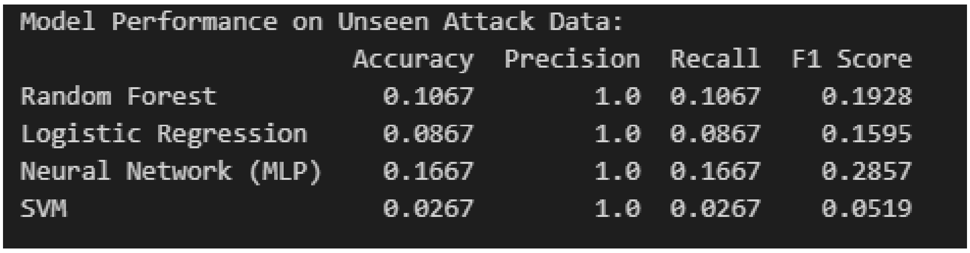
```
Model Performance on Unseen Attack Data:
                      Accuracy  Precision  Recall  F1 Score
Random Forest           0.1000        1.0  0.1000    0.1818
Logistic Regression     0.0667        1.0  0.0667    0.1250
Neural Network (MLP)    0.1200        1.0  0.1200    0.2143
SVM                     0.0267        1.0  0.0267    0.0519
```

Figure 10: Performance comparison of machine learning models across different feature selection sizes (N = 40) - Unseen Data



```
=== Model Performance Comparison ===
                  Model  Accuracy  Precision  Recall  F1 Score
0         Random Forest    0.9470     0.9525  0.9470    0.9466
1   Logistic Regression    0.8540     0.8611  0.8540    0.8525
2                   SVM    0.8557     0.8662  0.8557    0.8541
3  Neural Network (MLP)    0.8953     0.9096  0.8953    0.8959
```

Figure 11: Performance comparison of machine learning models across different feature selection sizes (N = 50) - Test Data



```
Model Performance on Unseen Attack Data:
                      Accuracy  Precision  Recall  F1 Score
Random Forest           0.1067        1.0  0.1067    0.1928
Logistic Regression     0.0867        1.0  0.0867    0.1595
Neural Network (MLP)    0.1667        1.0  0.1667    0.2857
SVM                     0.0267        1.0  0.0267    0.0519
```

Figure 12: Performance comparison of machine learning models across different feature selection sizes (N = 50) - Unseen Data

Analysis

From the results, we observed the following patterns:

- Training performance was consistently high across all models and feature counts (N values), especially for Random Forest and Neural Network (MLP).
- Performance on unseen data dropped drastically for all models, regardless of the number of features selected.

Specifically:

- Random Forest achieved the highest F1 scores on the training data but exhibited the largest performance drop on unseen data.
- The Neural Network (MLP) maintained relatively higher F1 scores compared to other models when evaluated on unseen data, although performance was still significantly lower than on training data.

- Increasing the number of features (from 2 to 50) did not significantly improve model performance on unseen data.

Conclusion

Feature selection (choosing top-N features) has some effect on model performance, but it does not resolve the problem of poor generalization to unseen data. The drop in F1 Score persisted across all tested N values.

This leads to the conclusion that the primary reason for poor performance on unseen data is not feature selection, but likely the distributional shift between the training and unseen datasets. In other words, the attack data has characteristics that differ substantially from what the models were trained on, causing the models to fail in generalizing to these new patterns.

Key Takeaways

- Feature selection (Top-N) contributes to model robustness but cannot fully bridge the generalization gap.
- All models experience a large drop in F1 Score on unseen data, regardless of how many features are used.
- Random Forest has the highest F1 score during training but generalizes poorly to new attack patterns.
- Neural Network (MLP) demonstrates relatively better robustness across varying N values.
- Increasing N from 2 to 50 does not significantly reduce the performance drop, confirming that the issue lies more in the nature of the unseen data than in the quantity of selected features.

2 Analysis of Poor Model Performance on Live Slowloris Attack Data

Objective

The goal was to understand why a trained machine learning intrusion detection model performed poorly on live demo traffic, despite showing strong performance during validation on known attack data. The live traffic was generated using a Slowloris attack, executed for approximately 15 seconds and processed via `CICFlowMeter` to generate flow-based features.

Background

The models (Random Forest, Logistic Regression, SVM, MLP) were trained on a balanced dataset composed of:

- Normal traffic
- Slowloris attacks
- UDP/ICMP flood attacks

Feature selection was performed using Random Forest feature importance, with the top 30 features selected for final model training.

3 Step-by-Step Methodology

1. **Feature Selection Context**
   The trained models were using the top 30 features selected via Random Forest importance from the balanced dataset. These features included:

   - Timing metrics (`Flow IAT Min`, `Flow IAT Max`, `Bwd Pkts/s`)
   - Port information
   - Byte/packet rates

   All of which were considered discriminative for the known attacks in the dataset.

2. **Live Demo Traffic Capture**
   The Slowloris traffic was captured in a `.pcap` file and converted to `.csv` using `CICFlowMeter`, yielding ∼1200 flows. This file (`demotraffic.csv`) represented the real-world unseen traffic used to test model generalization.

3. **Model Predictions**
   When tested on this data, all models showed very poor performance:

   - Random Forest
   - Logistic Regression
   - Support Vector Machine (SVM)
   - Neural Network

Table 1: Model Performance on Unseen Attack Data

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.0898	1.0	0.0898	0.1648
Logistic Regression	0.0719	1.0	0.0719	0.1341
Neural Network (MLP)	0.1497	1.0	0.1497	0.2604
SVM	0.0180	1.0	0.0180	0.0353

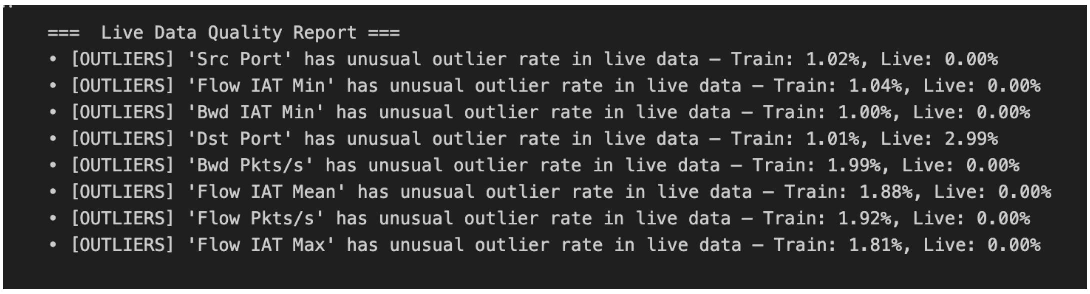
4 Diagnostic Methodology

To investigate the model's poor performance, we ran a feature-level comparison between the live attack traffic (`demotraffic.csv`) and the training dataset (`balanced_data.csv`).

The diagnostic script performed the following:

1. **Feature Selection**: Selected the top 10 features from training (those with highest predictive importance)
2. **Data Cleaning**:
   - Dropped unused columns (Flow ID, IP addresses, etc.)
   - Removed infinite or missing values
3. **Feature Analysis**: For each feature:
   - Checked for `NaN` or `Inf` values (data corruption)
   - Compared mean shift (to identify significant behavioral differences)
   - Calculated outlier rate in training vs live data, based on 1st and 99th percentiles

5 Script Output



```
===  Live Data Quality Report ===
• [OUTLIERS] 'Src Port' has unusual outlier rate in live data – Train: 1.02%, Live: 0.00%
• [OUTLIERS] 'Flow IAT Min' has unusual outlier rate in live data – Train: 1.04%, Live: 0.00%
• [OUTLIERS] 'Bwd IAT Min' has unusual outlier rate in live data – Train: 1.00%, Live: 0.00%
• [OUTLIERS] 'Dst Port' has unusual outlier rate in live data – Train: 1.01%, Live: 2.99%
• [OUTLIERS] 'Bwd Pkts/s' has unusual outlier rate in live data – Train: 1.99%, Live: 0.00%
• [OUTLIERS] 'Flow IAT Mean' has unusual outlier rate in live data – Train: 1.88%, Live: 0.00%
• [OUTLIERS] 'Flow Pkts/s' has unusual outlier rate in live data – Train: 1.92%, Live: 0.00%
• [OUTLIERS] 'Flow IAT Max' has unusual outlier rate in live data – Train: 1.81%, Live: 0.00%
```

Figure 13: Live Data Quality Report showing outlier analysis between training and live attack data

6 Interpretation

The live Slowloris traffic exhibited excessive uniformity and low variance, with minimal deviation across packets. This behavior aligns with expected Slowloris attack characteristics:

- Maintaining connections open indefinitely
- Slowly transmitting partial HTTP headers to evade detection
- Mimicking normal traffic patterns in both volume and transmission rate

Consequently, the live traffic failed to activate the high-risk thresholds learned during model training. The classifiers—particularly Random Forest and SVM—demonstrated limited capability to associate this stealthy traffic pattern with either:

- The aggressive signatures of UDP/ICMP floods, or
- Even the synthetic Slowloris examples present in the training data

Table 2: Model Sensitivity to Attack Characteristics

Model	**Limitation**
Random Forest	Overfitted to high-variance attack patterns
SVM	Linear kernel ineffective for low-variance detection
MLP	Marginally better at subtle pattern recognition
Logistic Regression	Oversimplified decision boundaries

Conclusion and Recommendations

The live Slowloris attack evaluation revealed a critical gap in the current model's ability to generalize beyond the aggressive patterns seen during training. Despite achieving high accuracy on the balanced dataset, all models—particularly Random Forest and SVM—performed poorly on live traffic due to its unusually low variance and uniform feature distribution. This behavior is consistent with the stealthy nature of Slowloris attacks, which aim to evade detection by mimicking benign traffic patterns while maintaining open connections with minimal activity.

This discrepancy highlights a **distributional shift** between the training and live datasets. The models were effective at detecting high-intensity attacks (e.g., UDP/ICMP floods) but lacked sensitivity to low-volume, persistent threats like Slowloris, which failed to trigger learned risk thresholds across key features such as `Flow IAT Min`, `Bwd IAT Min`, and `Flow Pkts/s`.

Recommendations for Improving Model Robustness

1. **Augment Training Data with More Slowloris Variants**
   Include multiple Slowloris samples with varying intensity and timing profiles to better represent real-world variability. This prevents overfitting to any single pattern of Slowloris behavior.

2. **Feature Engineering for Temporal Behavior**
   Introduce features that track *session-level duration*, *incomplete connections*, or *delayed header transmissions*—more indicative of application-layer DoS attacks like Slowloris.

3. **Apply Data Augmentation or Adversarial Simulation**
   Use synthetic traffic generators to simulate harder-to-detect variants of existing attacks. This improves resilience to evasive tactics.

4. **Integrate Anomaly Detection Models**
   Combine the supervised classifiers with unsupervised or semi-supervised models (e.g., Isolation Forest, Autoencoders) to flag deviations in session consistency, even when absolute feature values remain within expected ranges.

5. **Use Feature Drift Monitoring in Production**
   Implement ongoing statistical comparison between live traffic and training distributions to detect when models are operating out of scope—allowing proactive retraining or adaptation.