Anomaly Detection of DDoS Attacks using Data Augmentation

Team Wesem
Wesley Unyoung Kim, Emily Bae

Abstract

In the field of anomaly detection for Distributed Denial of Service (DDoS) attack mitigation, datasets suffer a class imbalance issue. Specifically, the proportion of “normal” data points significantly outnumber that of “anomalous” cases, which represent the DDoS attacks. Such class imbalance can often deter the creation of performant anomaly detection datasets and decrease the accuracy of machine learning models, making it challenging to detect the less frequent yet crucial malicious attack cases. To address this imbalance issue, we will take the DDoS dataset and perform data augmentation on malicious DDoS attack cases to balance the dataset. Because these attacks are fundamentally networking challenges, detecting these cases involves monitoring network traffic and identifying patterns that deviate from the “norm”. This project directly ties to the course theme by employing ML techniques such as data augmentation to address systems problems and improve model performance. Other interesting research work related to this problem includes “Contrastive Attributed Network Anomaly Detection with Data Augmentation”, which also discusses implementing different data augmentation strategies and feeding the augmented datasets to a Siamese graph neural network encoder. Relatedly, the goal of this project is to research different methods for data augmentation and compare how each affects and improves the accuracy of anomaly detection classification results. We seek to maximize Precision and Recall value when labeling malicious cases positive. As normal cases are already prevalent in available datasets, accuracy is not as informative as precision, since predicting everything “normal” would still give us high accuracy.

Section 1: Data Pre-Processing

The dataset we used is called Hikari-2021 from Zenodo, developed by a team of researchers. It represents network intrusion data based on real and encrypted synthetic attack traffic. There are three types of attacks, namely XMRIGCC CryptoMiner, Bruteforce-XML, and Bruteforce. We group all of these cases together under label 1 to denote malicious attacks, labeling the remaining benign cases as 0. The proportion between benign cases, where no attack occurs, and malicious attacks is approximately 94.15% to 5.84%, which is an example of class imbalance typical of most anomaly detection problems. We first pre-processed the data to only include features relevant to network flow such as total number of packets, packets per second, IAT, etc. Then we created heatmap and distribution data visualizations to identify any correlation among features and compare the distribution of feature data between benign and attack cases.

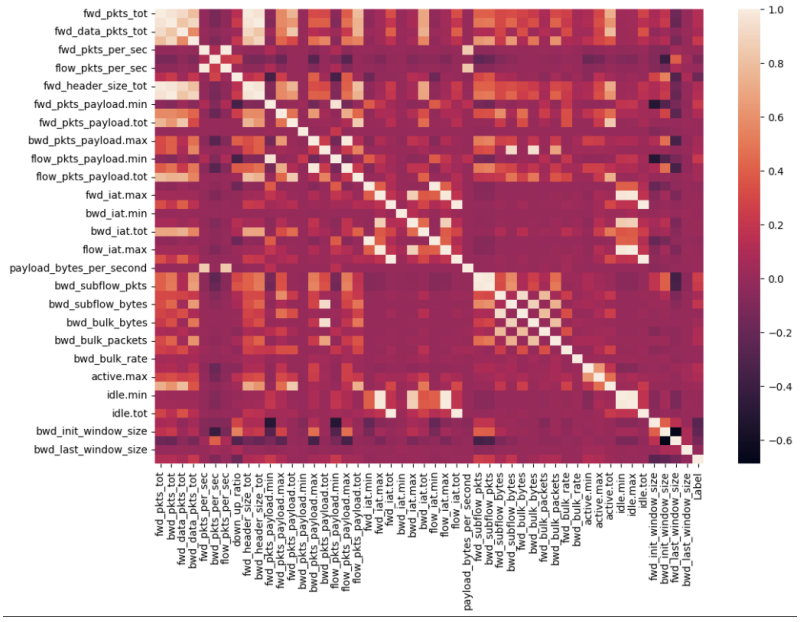
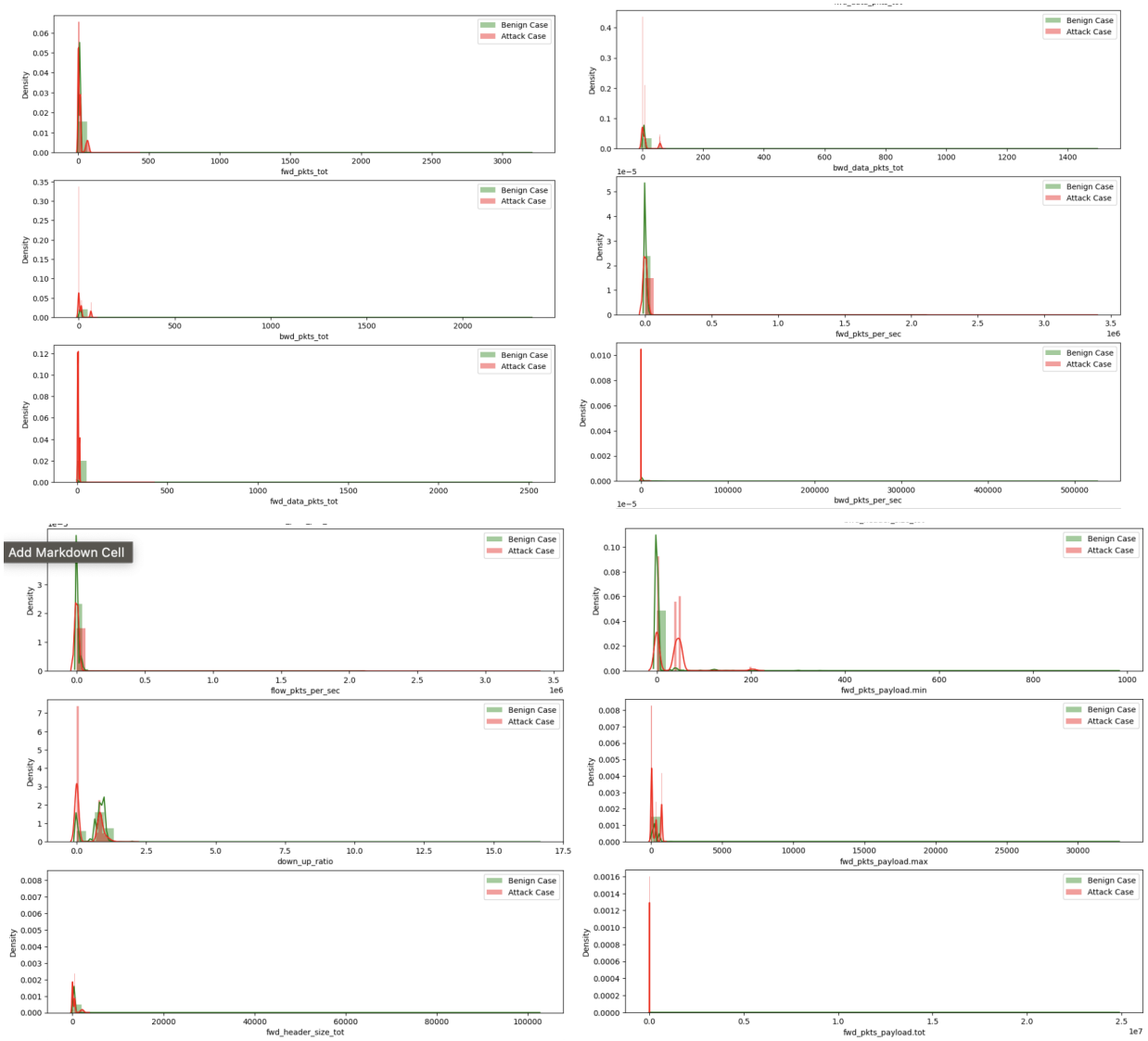


Figure 1
Heat map of correlation across features

Based on the color scale, the lighter colors indicate positive correlation, while darker colors indicate negative correlation. For example, we see that the features 'fwd_pkts_tot' and 'fwd_data_pkts_tot' have a positive correlation, which makes sense because the total number of packets sent should be equal to the total number of data packets sent plus the total number of control packets sent. On the contrary, 'fwd_pkts_per_sec' has a negative correlation with 'fwd_pkts_payload' as indicated by the dark purple cell. This makes sense given that the bigger the packet payload, the larger the packet size, which reduces latency since the packet takes a longer time to be transferred across the network. The diagonal axis of light cream colored cells typically represent the correlation of each feature with itself. It provides a reference point for interpreting off-diagonal cells. If the color of those cells are the same as the diagonal color, then those features are as strongly correlated with each other as they are with themselves,

suggesting a very strong positive correlation. For instance, 'fwd_iat_max' has a strong correlation with idle times. As such, this correlation diagram suggests which features are highly correlated with each other. It's typical in anomaly detection to remove features that are highly correlated with each other to reduce the dimensionality of the dataset. However, in our case, the features have a weak correlation as the cells are mostly magenta-colored. So we leave the dataset as is.

Now we compare the distribution of feature data for benign and attack cases.



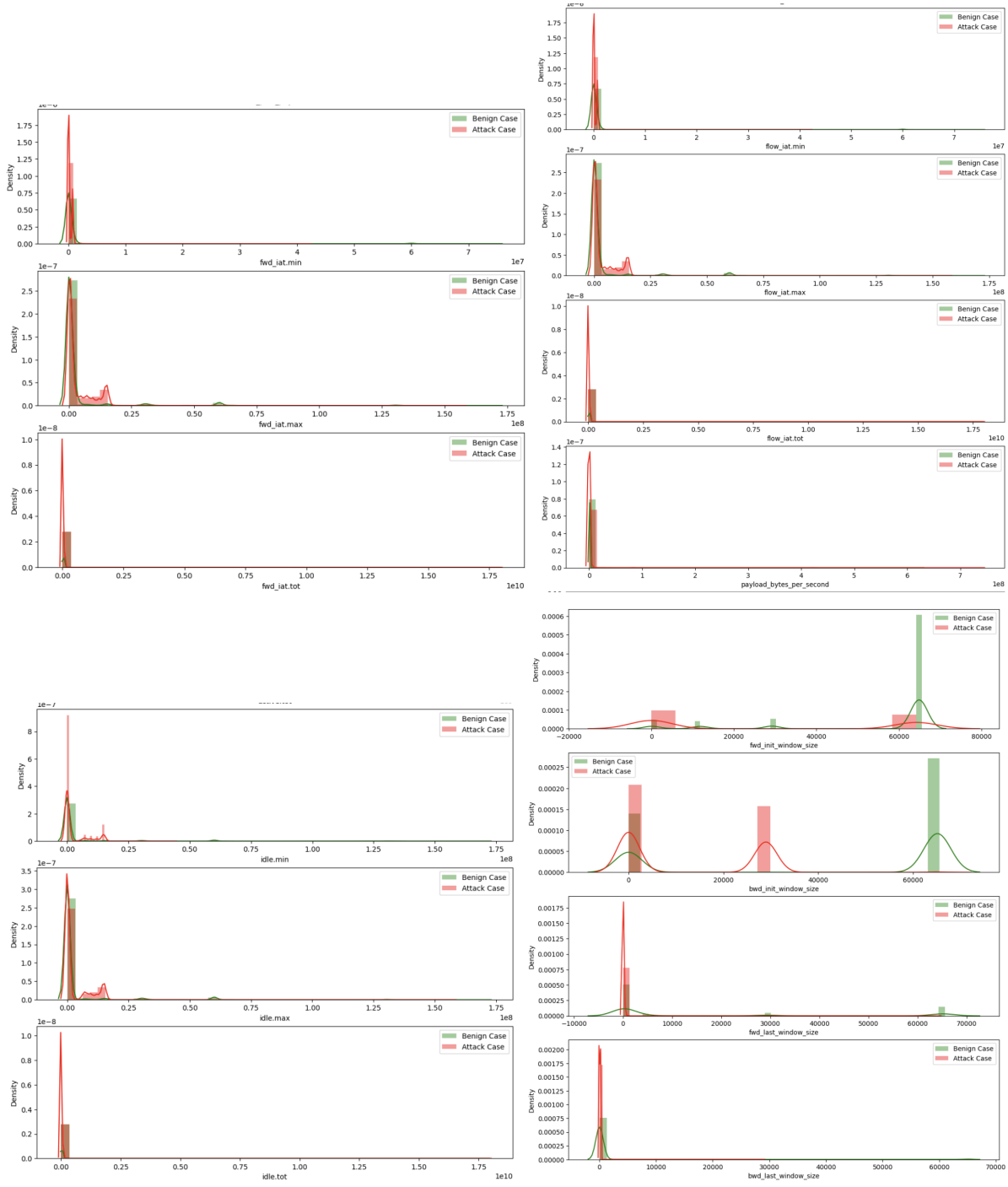


Figure 2

Visualization of distribution of feature data for benign and attack cases

Based on the above distribution plots, we can see that the distributions of the features between benign and attack cases have significant overlap for the majority. However, there are some features that have a clear distinction between benign and attack cases such as 'window_size'. In most cases, the distributions are not distinct enough to be able to clearly distinguish between

benign and attack cases, though attack cases seem to have a higher density (higher peak) in general.

Section 2: Model Training Part 1

After pre-processing the data, we performed anomaly detection using two models.
1. Logistic Regression and 2. Isolation Forest.

As shown, we have the following classification report and accuracy score for each model, which indicates that Logistic Regression performs better in general, in terms of accuracy score, precision, recall, and f1-score.

```
Logistic Regression
Accuracy: 0.922
              precision    recall  f1-score   support

           0       0.96      0.96      0.96     30099
           1       0.34      0.35      0.34      1857

    accuracy                           0.92     31956
   macro avg       0.65      0.66      0.65     31956
weighted avg       0.92      0.92      0.92     31956

Isolation Forest
Accuracy: 0.9084678933533609
              precision    recall  f1-score   support

           0       0.95      0.95      0.95     30099
           1       0.22      0.23      0.23      1857

    accuracy                           0.91     31956
   macro avg       0.59      0.59      0.59     31956
weighted avg       0.91      0.91      0.91     31956
```

Figure 3
Classification report and accuracy score of Logistic Regression and Isolation Forest models

We then computed confusion matrices for each model and plotted it, as shown below.

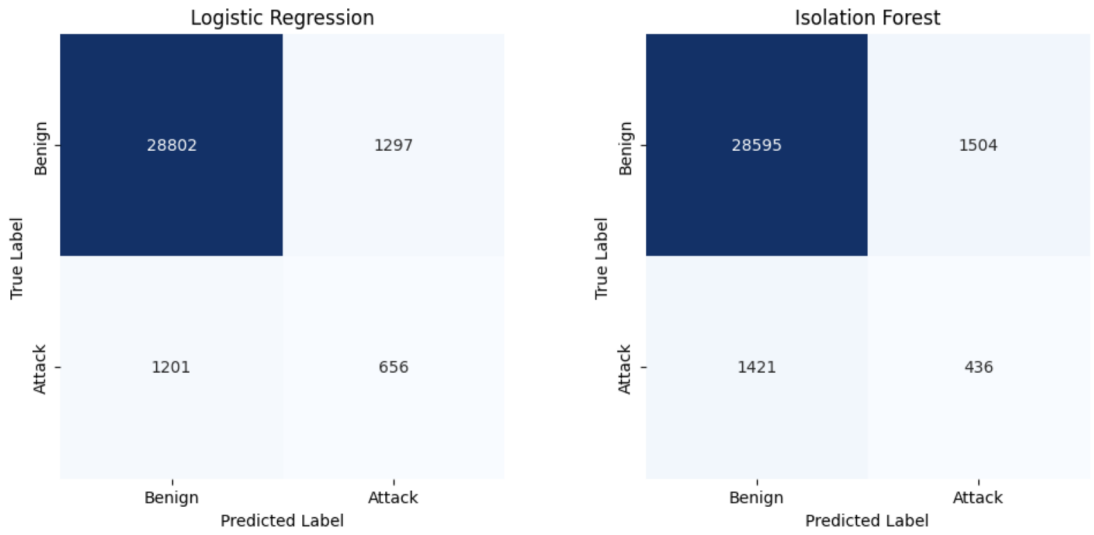


Figure 4
Confusion Matrices for Logistic Regression and Isolation Forest models

The first confusion matrix plot on the left shows that Logistic Regression has more true positives and fewer false negatives than Isolation Forest. This means that Logistic Regression is more likely to correctly classify attack cases. This is ideal for our case, as we want to minimize the number of false negatives and capture as many attack cases as possible. Although Logistic Regression has a greater number of false positives, mistaking benign cases for attacks, this is less of a concern in our case, as our objective is to detect anomalies. The second plot shows that Isolation Forest has more false negatives and fewer false positives, indicating that it is less sensitive in detecting attacks, but more conservative in classifying malicious attacks. This presents a problem, as in anomaly detection, we would rather misclassify benign cases as attacks than the opposite direction.

Therefore, these metrics, coupled with the fact that Logistic Regression has a higher accuracy score, precision, recall, and f1-score, led us to choose Logistic Regression as our anomaly detection model.

However, as we noted before, there is class imbalance in the dataset between benign cases and attack cases, with 95.14% constituting normal, benign samples. Therefore, we decided to conduct data augmentation to oversample attack cases such that we can better identify underlying patterns in the data points corresponding to malicious intrusions.

Section 3: Data Augmentation

We selected two prominent data augmentation techniques, namely:
1. SMOTE Oversampling (Synthetic Minority Over-sampling Technique)
2. Random Oversampling

Section 3.1 : SMOTE Oversampling

SMOTE is a technique used for generating synthetic data of minority class (attack cases) to prevent overfitting. The process goes like the following:

- For each sample in the minority class, the algorithm computes the k-nearest neighbors.
- Depending on the amount of over-sampling needed, one or more of the k-nearest neighbors are chosen to create the synthetic samples.
- A synthetic sample is created by choosing one of the k-nearest neighbors and computing a linear combination of the feature vectors of the minority class sample and its chosen neighbor.

SMOTE helps overcome overfitting problems posed by random over-sampling, as it generates new, synthetic training samples that are plausible rather than simply copying existing ones. It is especially useful in anomaly detection cases where the class of interest (i.e. malicious network intrusions) is underrepresented in the available training data. Using SMOTE, we now have a more balanced dataset, which can help improve the performance of the classification algorithm.

We compared the results of logistic regression modeling on SMOTE oversampled data vs. our regular data from before.

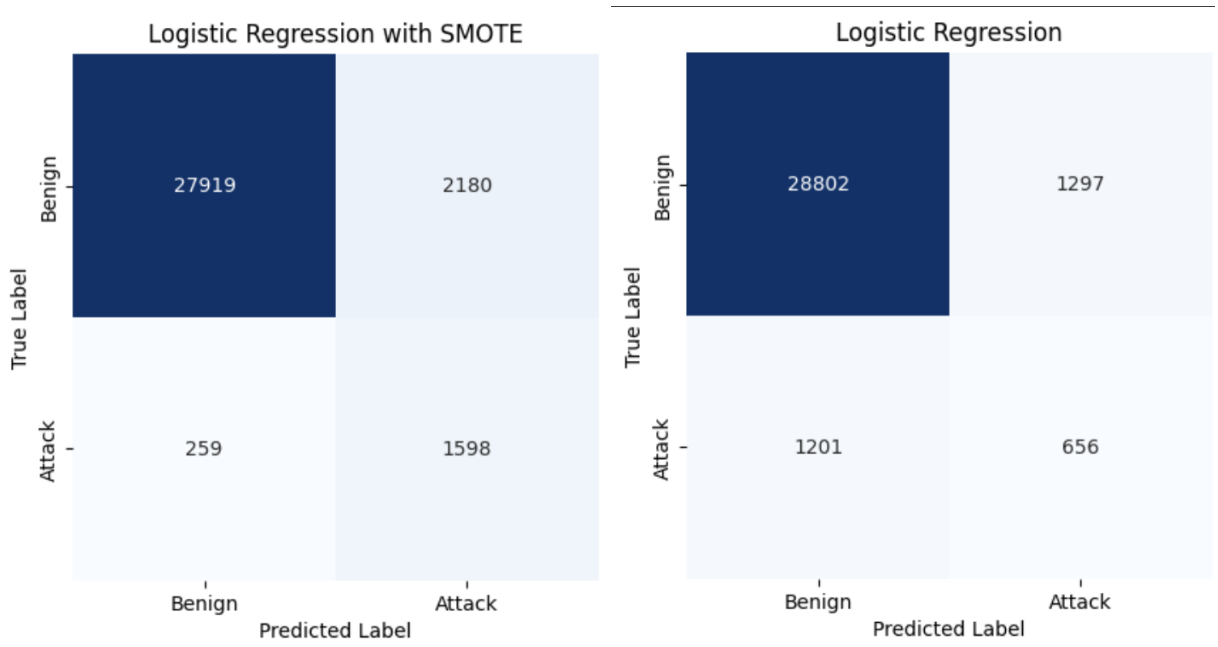


Figure 5
Comparison of Confusion Matrices for SMOTE vs. non-SMOTE Oversampled Data

From these two confusion matrices, we observed that after using SMOTE to augment the data, the logistic regression model has become more sensitive to the attack class, correctly identifying a higher number of attacks (TP), but at the cost of incorrectly labeling more benign cases as attacks (FP). Since we are more concerned with correctly identifying attack cases, we determined that our model performance has improved.

Next we performed data augmentation using the second technique, Random Oversampling.

Section 3.2 : Random Oversampling

Random oversampling is a technique used to balance class distribution in a dataset with imbalanced classes. Specifically, it involves randomly replicating instances from the minority class to increase its representation. This method can help improve the performance of classification algorithms by providing a more balanced class distribution.

We compared the results of logistic regression modeling on randomly oversampled data vs. our regular data from before.

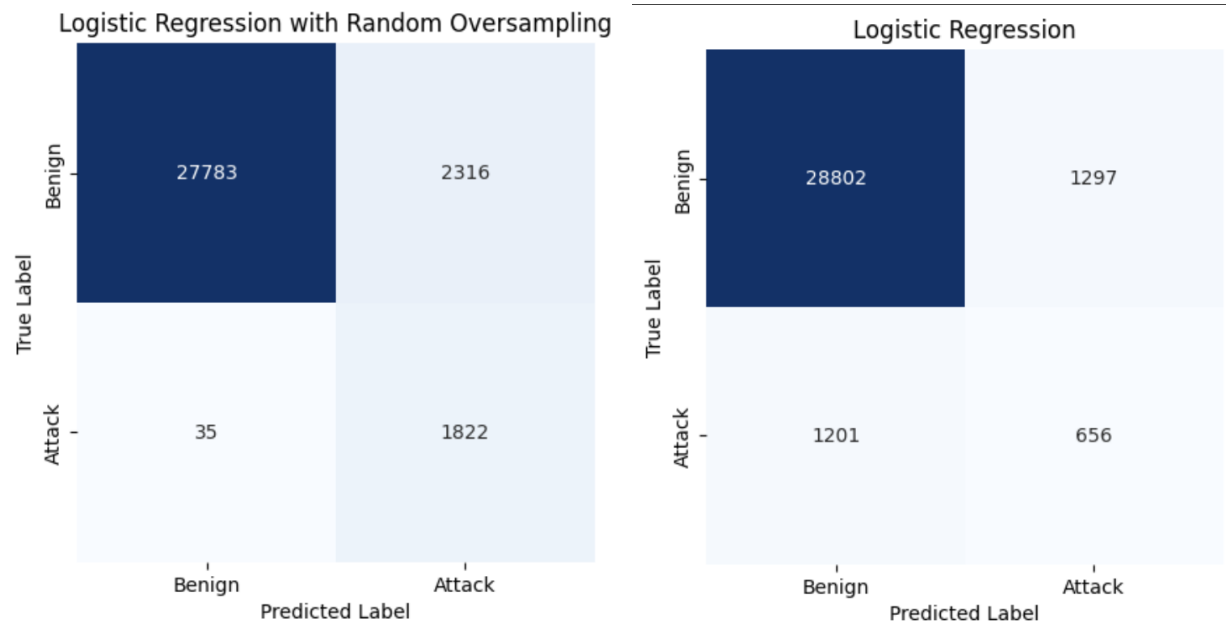


Figure 6
Comparison of Confusion Matrices for Random Oversampled vs. non-oversampled Data

As shown, Random Oversampling results in a much higher number of True Positives, which means it is better at detecting attacks than the model without oversampling. It also shows a lower number of False Negatives, indicating fewer missed attacks. However, this augmented model also has more False Positives, which means it incorrectly labels more benign cases as

attacks. The non-augmented model is more conservative in predicting attacks, resulting in fewer False Positives but also missing more actual attacks (higher False Negatives). As such, given our objective of identifying attack cases correctly, we again observed that our model performance has improved.

Section 4: Conclusion

Based on the comparisons we made above, we can conclude that in the field of anomaly detection for identifying malicious network intrusions, both Random Oversampling and SMOTE are valuable techniques for addressing class imbalance between benign cases and attack cases. This is a common challenge in anomaly detection problems where the number of normal instances far exceeds the anomalous ones. Through Random Oversampling, we can increase the representation of the minority class by duplicating existing samples, which can lead to a more balanced dataset and thus enhance the sensitivity of anomaly detection models. However, this may also cause overfitting due to the mere replication of instances. On the other hand, if we employ the SMOTE data augmentation technique, we can generate new synthetic minority class samples by interpolating between existing ones, which introduces more diversity and helps in creating a more generalized model that is robust against overfitting to the minority class. While both methods improve the detection rates of malicious activities by providing more examples for the model to learn from, SMOTE's ability to generate new, yet reasonable samples offers a more effective and nuanced approach to improving classification performance in the context of network security.