02/12/2024-09/12/2024

WEEK 4

Enhancing CNN Performance Through Outlier Detection Methods

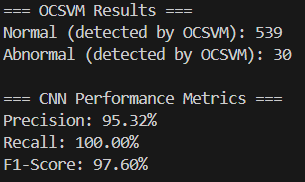
**In the first two weeks**, we focused on combining OCSVM with other outlier detection methods to determine the most efficient model. **In the third week**, we explored and studied how CNN models work. After these efforts, we decided to integrate OCSVM and CNN methodologies.

In this approach, the OCSVM model treats the **benign class** (normal) from the breast cancer dataset as the normal class and performs outlier detection on these data points. The detected outliers (-1) are categorized as the **abnormal class**, while the inliers (+1) are categorized as the **normal class**.

Following this, the CNN model processes the data based on these normal and anomalous classes. This combined approach is referred to as **Hybrid Anomaly Detection**.

By initially using OCSVM for outlier detection, followed by leveraging a powerful classifier like CNN, this method helps to clean and refine the dataset, making it more meaningful. As a result, the CNN model can better focus on distinguishing and understanding anomalous data points, improving its overall performance.

After creating our code using OCSVM and CNN, we printed the counts of normal and abnormal values as well as the F1, recall, and precision scores of the CNN model.



We encountered an issue with these outputs: the abnormal and normal data provided to the CNN model are highly imbalanced and insufficient. As a result, the F1, recall, and precision scores are unrealistic and overly optimistic.

Afterward, we began researching methods to address this issue. We decided to incorporate outlier detection methods that could work alongside the OCSVM approach into our code, similar to what we did in the earlier weeks.

Isolation Forest: Precision: 49.81%, Recall: 65.23%, F1-Score: 21.73%

Local Outlier Factor (LOF): Precision: 49.69%, Recall: 64.99%, F1-Score: 21.57%

One-Class SVM (OCSVM): Precision: 49.85%, Recall: 65.23%, F1-Score: 21.76%

Interquartile Range (IQR): Precision: 48.52%, Recall: 49.12%, F1-Score: 43.65%

Z-score: Precision: 45.92%, Recall: 48.91%, F1-Score: 37.83%

Combined Graph: Precision: 47.74%, Recall: 48.61%, F1-Score: 43.42%

Isolation Forest Outliers: 29

LOF Outliers: 29

OCSVM Outliers: 30

IQR Outliers: 103

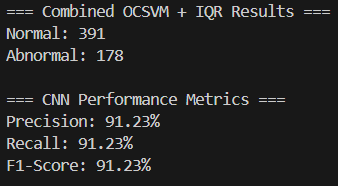
Z-score Outliers: 41

(LOF + Isolation Forest + OCSVM + IQR + Z-score) Outliers: 108

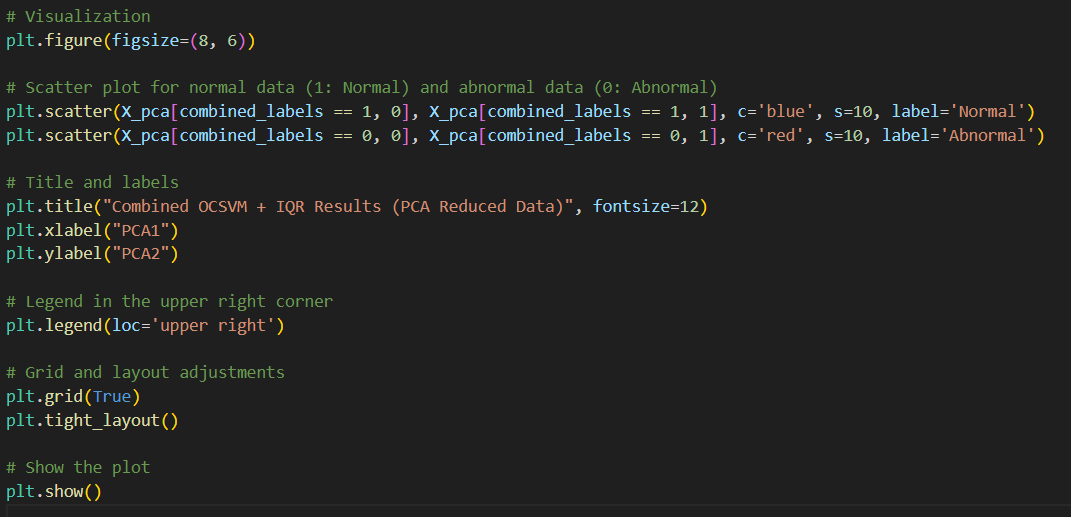
(LOF + Isolation Forest + OCSVM + IQR + Z-score) Common Outliers: 12

Building on the results we obtained in the first week, we started adding outlier detection methods. We decided to begin with the IQR method because it not only identifies the highest number of outliers but also does not negatively impact the F1, recall, and precision scores.

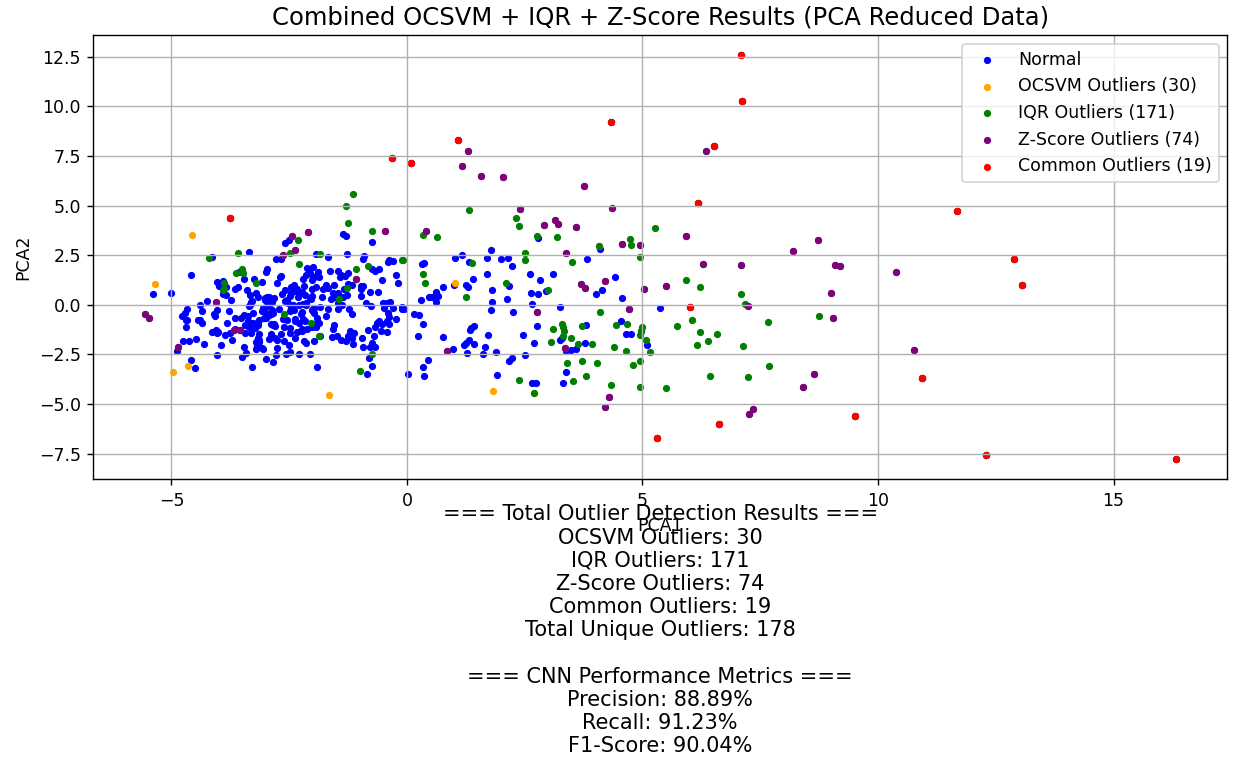
After adding the IQR method to the OCSVM approach, the CNN model continued processing more outliers, and the F1, precision, and recall scores became more realistic.



Afterward, to better observe the outliers, we made color adjustments and modifications to the plot.



To enable better observation and analysis on the plot, we added the number of outliers detected by each method, the F1, recall, and precision scores of the CNN model, and the total number of outliers detected by all methods combined below the plot. To enhance the performance and efficiency of the CNN, we incorporated the Z-Score method alongside the OCSVM and IQR methods.



Finally, we adjusted the size and font of the graph. To enable more meaningful observations, we also added the count of data points not classified as outliers to our code.

metin, diyagram, çizgi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Referencess:

<https://www.geeksforgeeks.org/z-score-for-outlier-detection-python/>

<https://www.mathworks.com/discovery/convolutional-neural-network.html#:~:text=A%20convolutional%20neural%20network%20(CNN,-series%2C%20and%20signal%20data>.

<https://www.kaggle.com/code/redpen12/anomaly-detection-with-cnn-autoencoders>

<https://www.kaggle.com/code/edumisvieramartin/butterfly-multiclass-image-classification-cnn>

<https://www.kaggle.com/datasets/phucthaiv02/butterfly-image-classification/data>