|  |  |  |  |
| --- | --- | --- | --- |
| In today's work, I read a CSV file containing temperature data and applied the Isolation Forest algorithm on these data. The algorithm calculates the isolation depth for each data point and marks those with low values as potential anomalies. In the analysis conducted with approximately 300 trees, the sample size for each tree was set to 512, and the maximum depth value was tested in the range of 10–12. As a threshold value, 6 was used for the average depth. Data below this value were marked as anomalies. In tests, due to the random sampling nature of the algorithm, small differences in results were observed in each run. Additionally, some frequently occurring values in the dataset (for example, 49–50) were sometimes marked as anomalies, while extreme values like 100–110 were not always considered anomalies. These results directly show the effect of the sampling structure on the outputs. In today's work, both data processing in C++ and the practical behavior of anomaly detection algorithms were examined in detail. | | | |
| **Sayfa No** | **Çalışmanın** | | **KONTROL** |
|  | Konusu :.........................................  ......................................................... | Yapıldığı Tarih  ...../..../202.. | ......................................  ...................................... |