*ML22/23-12: Implement Anomaly Detection Sample*

|  |  |
| --- | --- |
| Md Sohel Rana  sohel.rana@stud.fra-uas.de | Md Ashiqur Rahman  ashiqur.rahman@stud.fra-uas.de |

*Abstract*—HTM (Hierarchical Temporal Memory) is an impactful machine learning algorithm approach that is biologically inspired in both aspects, structurally and functionally, by the neocortex of a human brain that processes time series data in a distributed manner using a hierarchical network of nodes. HTM works in a decentralized manner with the help of a tiered arrangement that allows each node and column to learn and recognize patterns in input data. This feature enables actions such as processing information, recognizing and identifying patterns, and making future predictions based on previous learning. This is a potential approach which can be used for anomaly detection and prediction in numerous sectors such as healthcare, finance, geological disasters, cyber-intrusion detection, military surveillance, system fault detection. This paper presents an anomaly detection sample using an HTM model trained on multiple simple numeric integer sequences. This model learns patterns from the input data and identifies anomalies by comparing real data with predicted data from learning within a set tolerance threshold. The paper also provides a detailed explanation of anomaly detection techniques with algorithm implementation.

Keywords—HTM, anomaly detection, machine learning, multi-sequence learning, NeoCortex API.

# Introduction

Hierarchical Temporal Memory (HTM), a biologically constrained machine intelligence technique, was created by Numenta. It was first published in 2004 by Sandra Blakeslee and Jeff Hawkins, a brain scientist and the founder of the Redwood Neuroscience Research Institute [1].This machine learning algorithm works based on the theory of how the biological neocortex works, and this approach basically depends on principles of the Thousand Brains Theory.

# MethodS

This section should describe your work in detail. Here you can use references for your work and external sources.

# Results

This Part of the text describes results of your works. There can only be mentioned references, MUST point back to Methods and Intro chapter. No more external references.

Code examples must be provided to demonstrate how to use the algorithm/module. Provide a reference to more unit tests, which show the same in more detail. Also provide all diagrams with comments and reference to unit tests, which generate diagrams.

# Discussion

The conclusion of your work should be precise and concise. How was the project, what was done, what was the result. There can be discussion on further work and direction.

##### References

|  |  |
| --- | --- |
| [1] | J. H. a. S. Blakeslee, "On Intelligence," in *Henry Holt*, New York, 2004. |