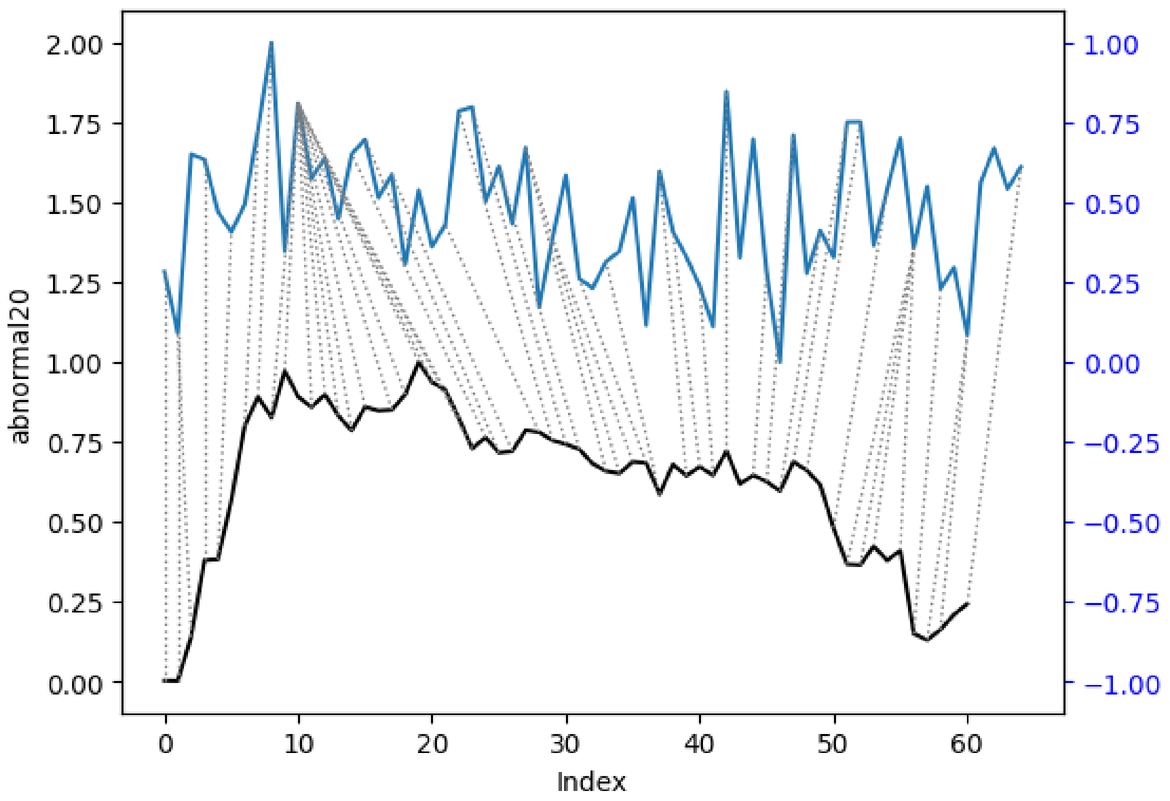
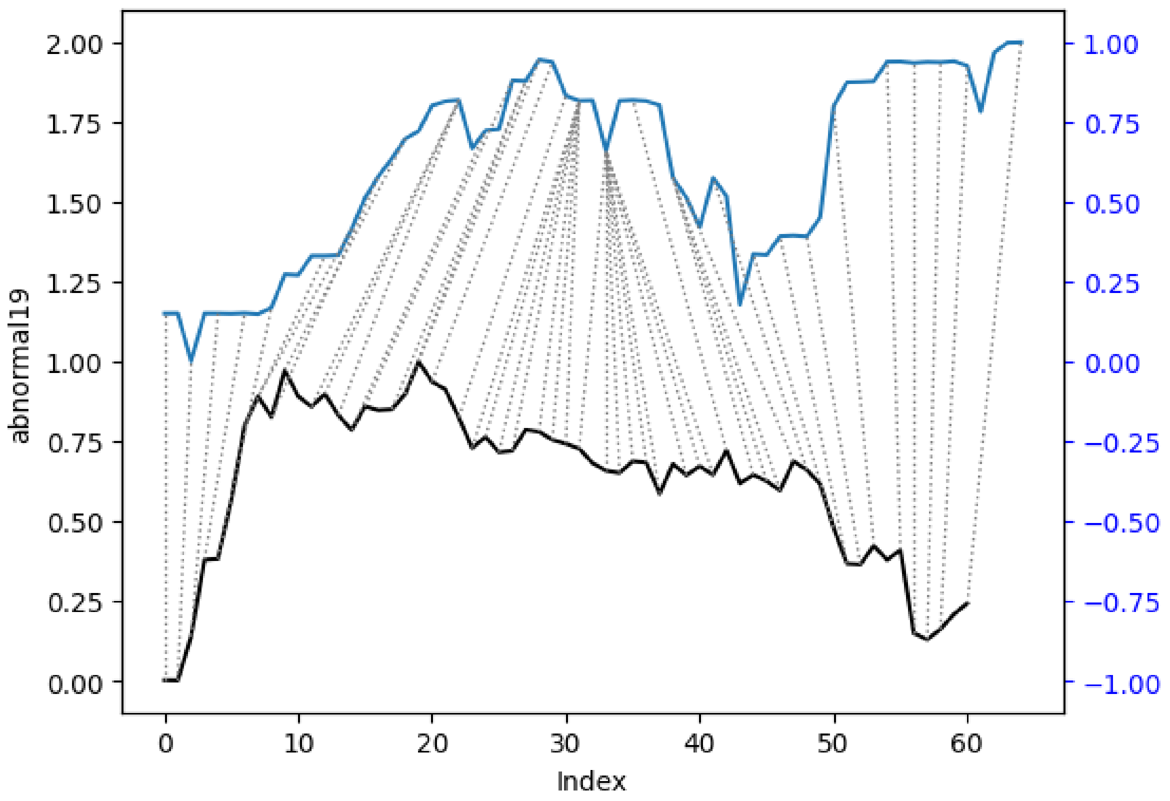
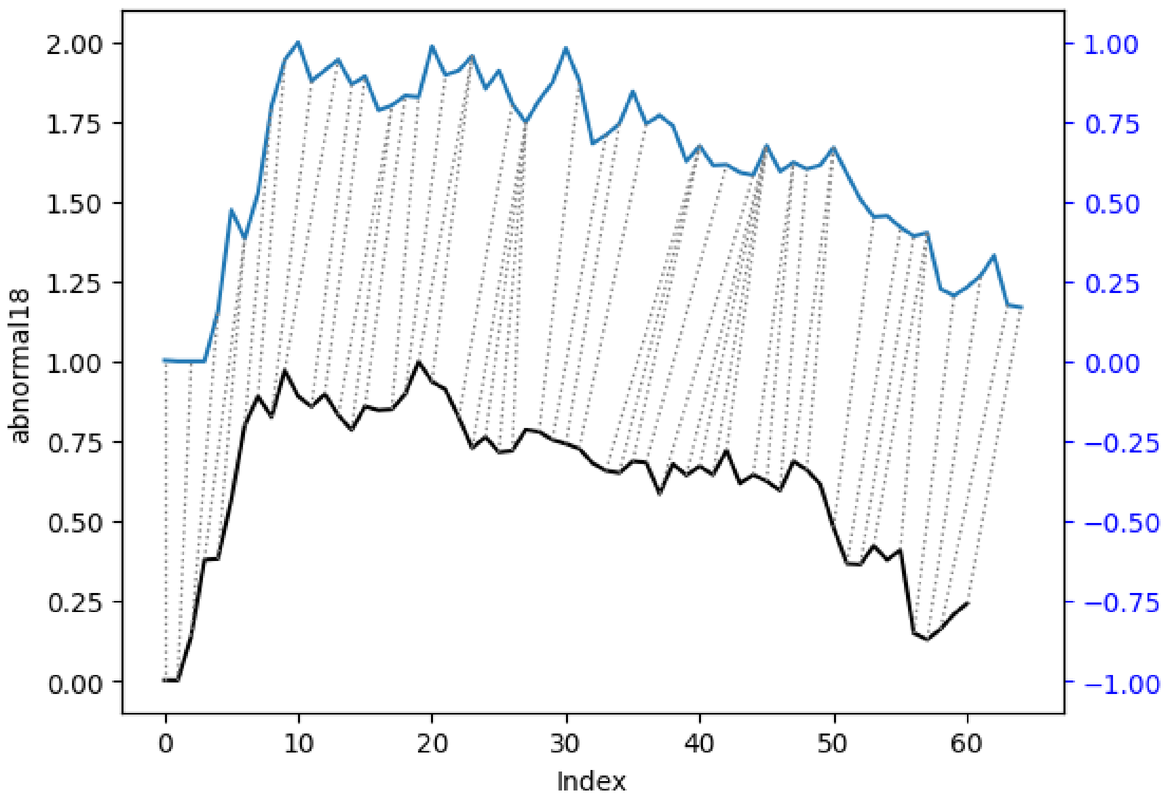
```python
# from .init import search_blue
from flask import request
import pymysql
from dtw import *

# 创建连接
db = pymysql.connect(host='localhost', user='tester', password='test', port=3306)
# 创建游标
cursor = db.cursor()


# 相似度比对
# @search_blue.route("/similarity_search", methods=['POST'])
def similarity_search():
    input_json = request.get_json(force=True)
    rank = int(input_json['rank'])  # top10=>rank=10  top5=>rank=5
    ab_node = str(input_json['ab_node'])    # 异常数据表名
    start_time = str(input_json['start_time'])  # 开始时间
    end_time = str(input_json['end_time'])  # 截止时间
    # 相似度分析
    res = process_similarity(rank, ab_node, start_time, end_time)
    #todo: 处理结果
    return res


# 相似度分析函数
def process_similarity(rank: int, ab_node:str, start_time: str, end_time: str):
    # 返回值：根据rank返回一个list，元素为相似度从高到低的<node_name:simi_value>

    # 得到表名
    nodes_name = get_similarity_node()
    # 得到异常序列和对比序列列表
    ret = get_similarity_data(nodes_name, ab_node, start_time, end_time)
    nodes_name.remove(ab_node)
    query = ret.pop(ab_node)
    # normalize
    query['y_data_list'] = list(map(lambda y: (y-query['y_min']) /
(query['y_max']-query['y_min']), query['y_data_list']))
    refers = ret

    simi_list = []
    while(len(refers)):
        curr_node = nodes_name.pop()
        refer = refers.pop(curr_node)
        # normalize
        refer['y_data_list'] = list(map(lambda y: (y-refer['y_min']) /
(refer['y_max']-refer['y_min']), refer['y_data_list']))
        # dtw算法对比两个序列的相似度
        alignment = dtw(x=query['y_data_list'],
                        y=refer['y_data_list'],
                        dist_method="euclidean",
                        step_pattern=asymmetric,
                        keep_internals=True)
        alignment.plot(type="twoway", offset=1, ylab=curr_node)
        # 将相似度分析结果加入list
        node_simi = dict()
        node_simi['node_name'] = curr_node
        node_simi['simi_value'] = alignment.__getattribute__('distance')
        node_simi['x_data_list'] = refer['x_data_list']
        node_simi['y_data_list'] = refer['y_data_list']
        simi_list.append(node_simi)
    simi_list.sort(key=lambda e: e['simi_value'])
    simi_list = simi_list[:rank]

    # 0
    node_simi = dict()
    node_simi['node_name'] = "abnormal0"
    node_simi['simi_value'] = 0
    node_simi['x_data_list'] = simi_list[0]['x_data_list']

    y_data_list = [-1, -1]
    y_data_list.extend(query['y_data_list'])
    y_data_list.extend([-1, -1])
    node_simi['y_data_list'] = y_data_list
    simi_list.append(node_simi)


    for simi in simi_list:
        print(simi['node_name'], simi['simi_value'])
    return


# 获取相似度比对的数据表名
def get_similarity_node(filename="AIBMA-ES-CLUSTER-20230423"):
    database_name = filename.replace("-", "_")
    database_sql = "USE " + database_name
    cursor.execute(database_sql)
    db.commit()
    res = []
    # 查询表名
    sql = "select table_name from information_schema.tables where
table_schema='AIBMA_ES_CLUSTER_20230423' order by
cast(substr(table_name,9,length(table_name)) as signed ) asc;"
    cursor.execute(sql)
    ret = cursor.fetchall()
    for item in ret:
        res.append(item[0])
    return res


# 根据输入的表名获取相似度信息
def get_similarity_data(nodes_name:list, ab_node:str, start_time:str,
end_time:str):
    database_sql = "USE AIBMA_ES_CLUSTER_20230423"
    cursor.execute(database_sql)
    db.commit()
    res = dict()
    # 根据需要查询的节点名称去查询
    for node_name in nodes_name:
        res[node_name] = dict()
        datas = select_node_similarity_data(node_name, ab_node == node_name,
start_time, end_time)
        res[node_name]['x_data_list'] = datas[0]
        res[node_name]['y_data_list'] = datas[1]
        res[node_name]['y_min'] = datas[2]
        res[node_name]['y_max'] = datas[3]
    return res


# 从表中截取序列
def select_node_similarity_data(table_name:str, isQuery:bool, start_time:str,
end_time:str):
    # 找到开始序列的首尾id
    sql = "select id from " + table_name + " where date=\'" + start_time +"\' or
date=\'" + end_time +"\'"
    cursor.execute(sql)
    datas = cursor.fetchall()
    start_id, end_id = datas[0][0], datas[1][0]

    sql = "select id,date,value from " + table_name + " where date!=''"
    cursor.execute(sql)
    datas = cursor.fetchall()
    x_data_list = []
    y_data_list = []
    # normalize
    y_min = float('inf')
    y_max = float(0)
    # 对于对比序列，考虑波动提前/延后的情况，将对比时间段边界放宽
    for data in datas:
        if (not isQuery and data[0] >= start_id - 2 and data[0] < start_id):
            x_data_list.append(data[1])
            y_data_list.append(float(data[2]))
        elif (data[0] >= start_id and data[0] <= end_id):
            x_data_list.append(data[1])
            y_data_list.append(float(data[2]))
        elif (not isQuery and data[0] > end_id and data[0] <= end_id + 2):
            x_data_list.append(data[1])
            y_data_list.append(float(data[2]))
        elif (data[0] > end_id + 2):
            break
        # normalize
        if(float(data[2]) > y_max): y_max = float(data[2])
        if(float(data[2]) < y_min): y_min = float(data[2])
    return x_data_list, y_data_list, y_min, y_max



if __name__ == '__main__':
    process_similarity(20, 'abnormal0', '2023/4/23 15:20', '2023/4/23 17:20')
```

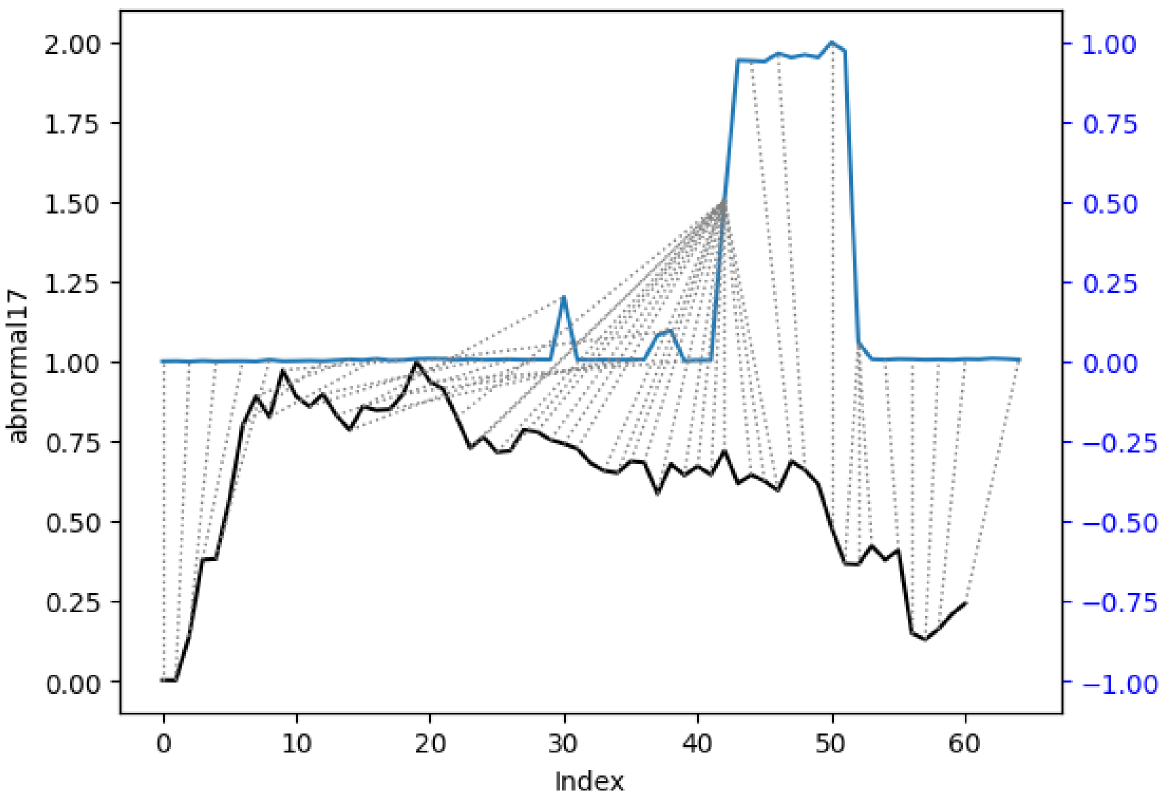
```
abnormal7 0.2841149520877342
abnormal14 1.6176622488894306
abnormal18 1.6402277584129519
```

```
abnormal9 1.7770795800035557
abnormal1 2.169332245890944
abnormal2 3.8874433503456367
abnormal16 6.009844201961209
abnormal10 7.250950154629579
abnormal13 7.419987630307434
abnormal20 7.428285102610168
abnormal11 7.543856745867301
abnormal6 7.936415273792573
abnormal4 8.346021935155434
abnormal3 9.519339739699864
abnormal19 9.917487191786924
abnormal15 15.273207908410988
abnormal5 15.772887093415138
abnormal12 21.123665972750704
abnormal8 22.103723271394397
abnormal17 23.32439814523837
abnormal0 0
```

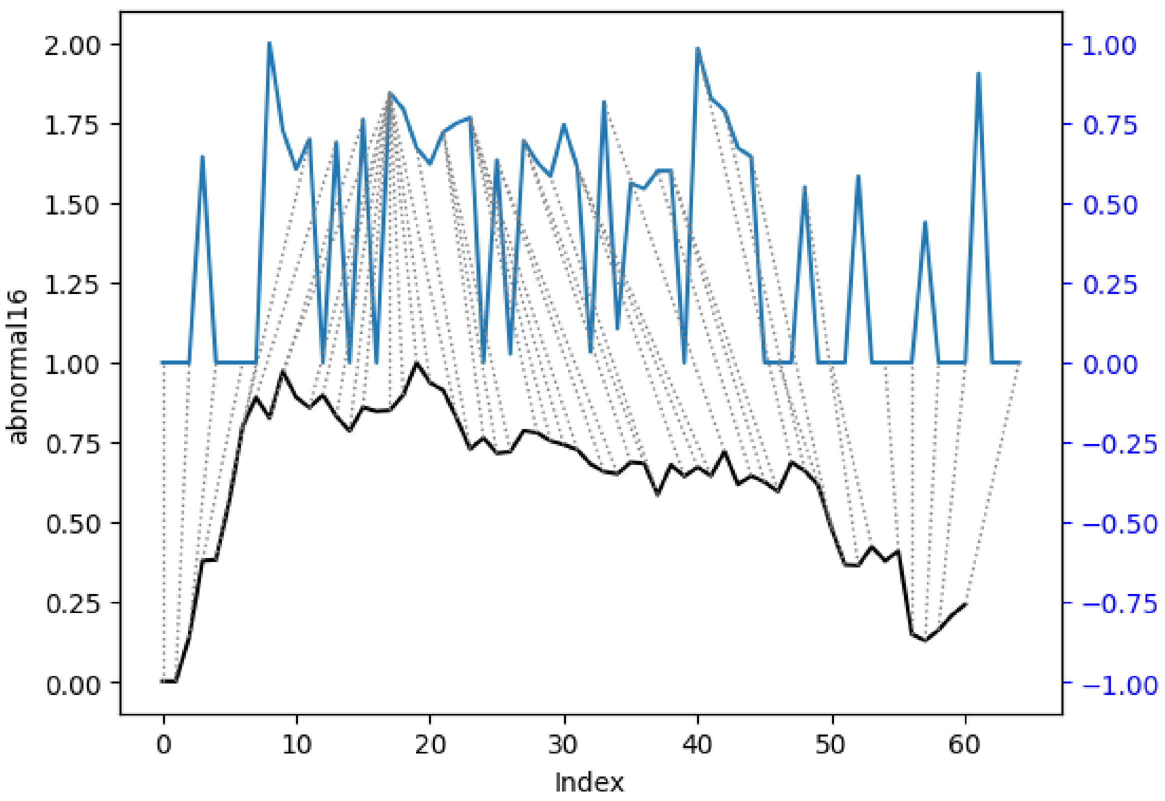



abnormal19
Index
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
60

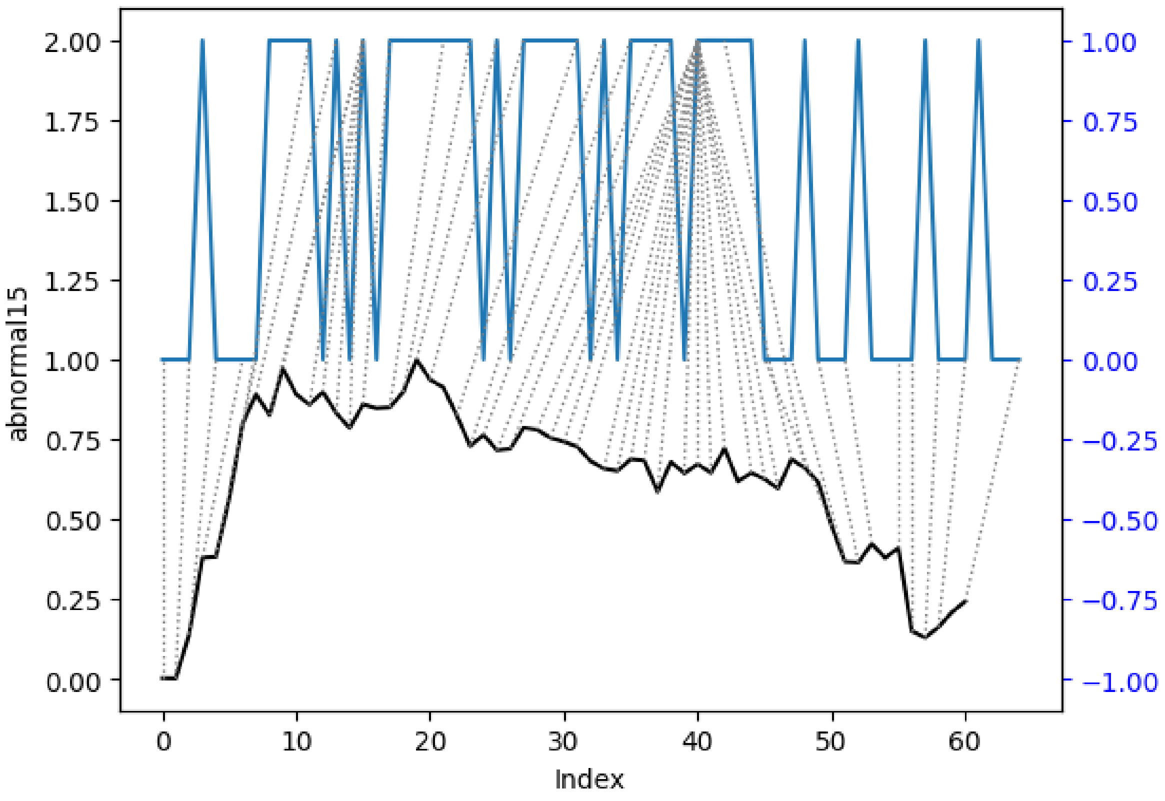

abnormal18
Index
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
60

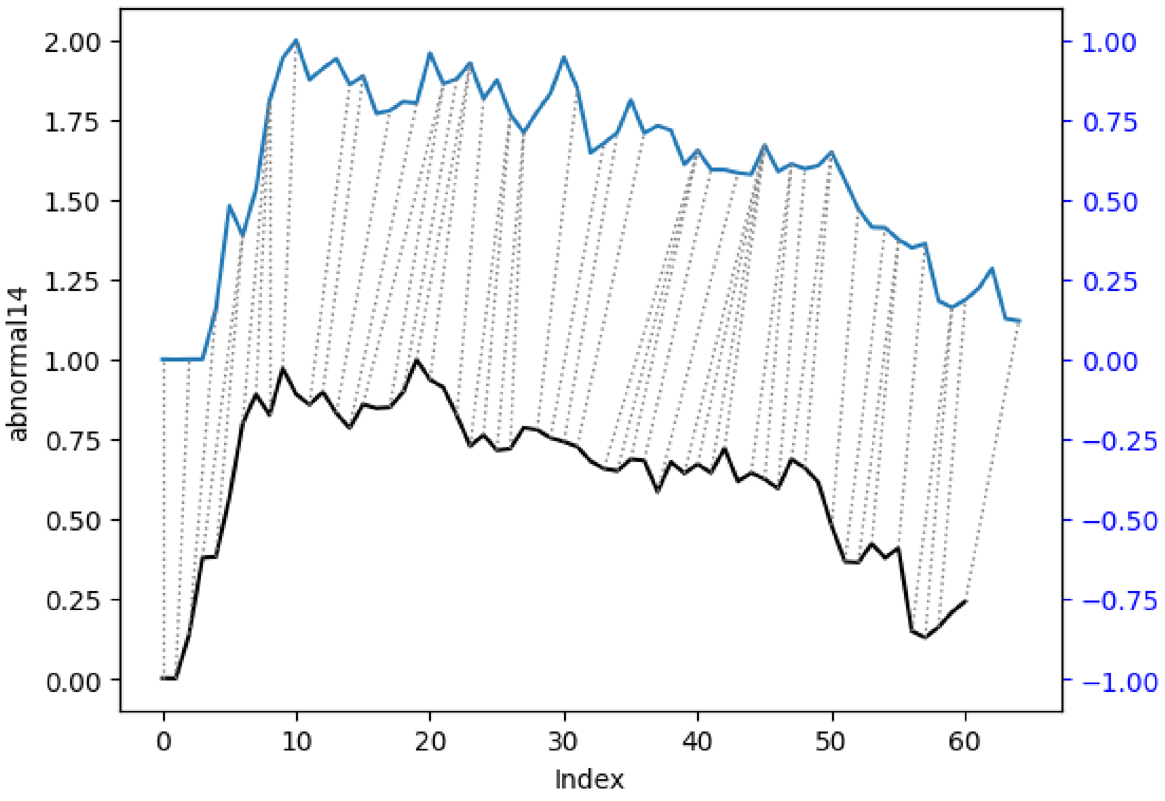


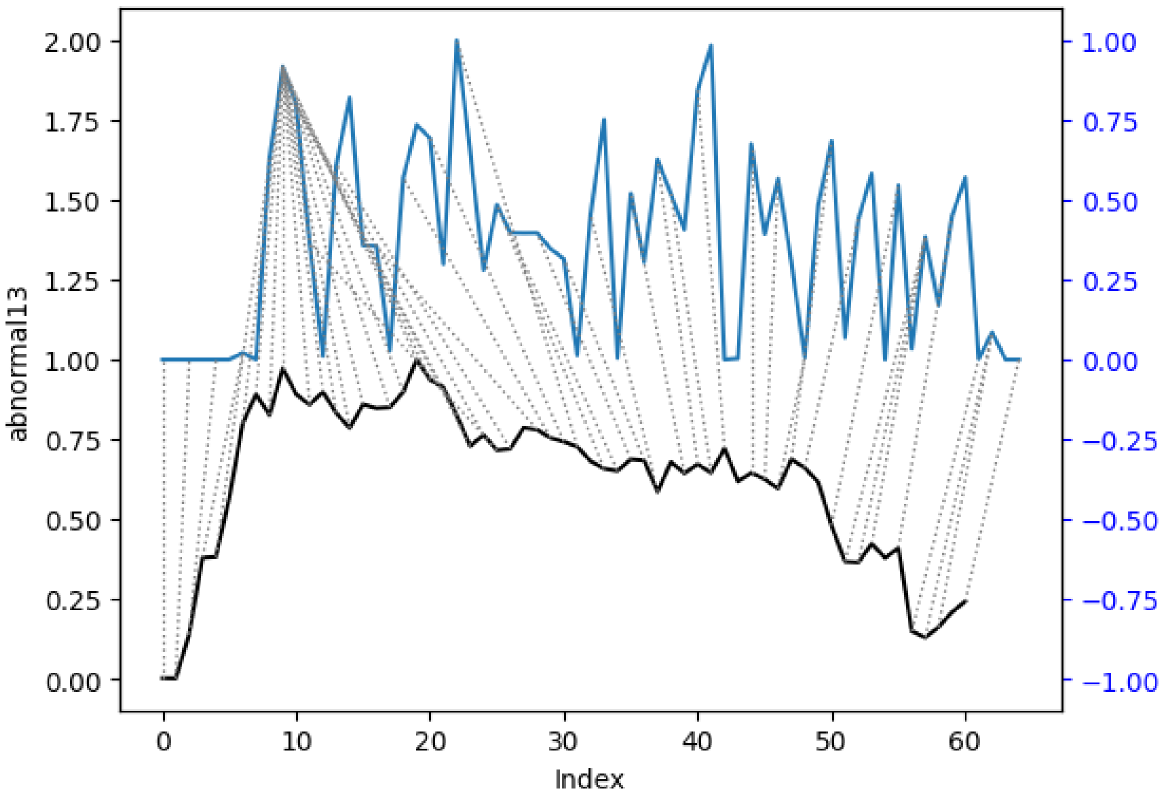
abnormal17
Index
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
60



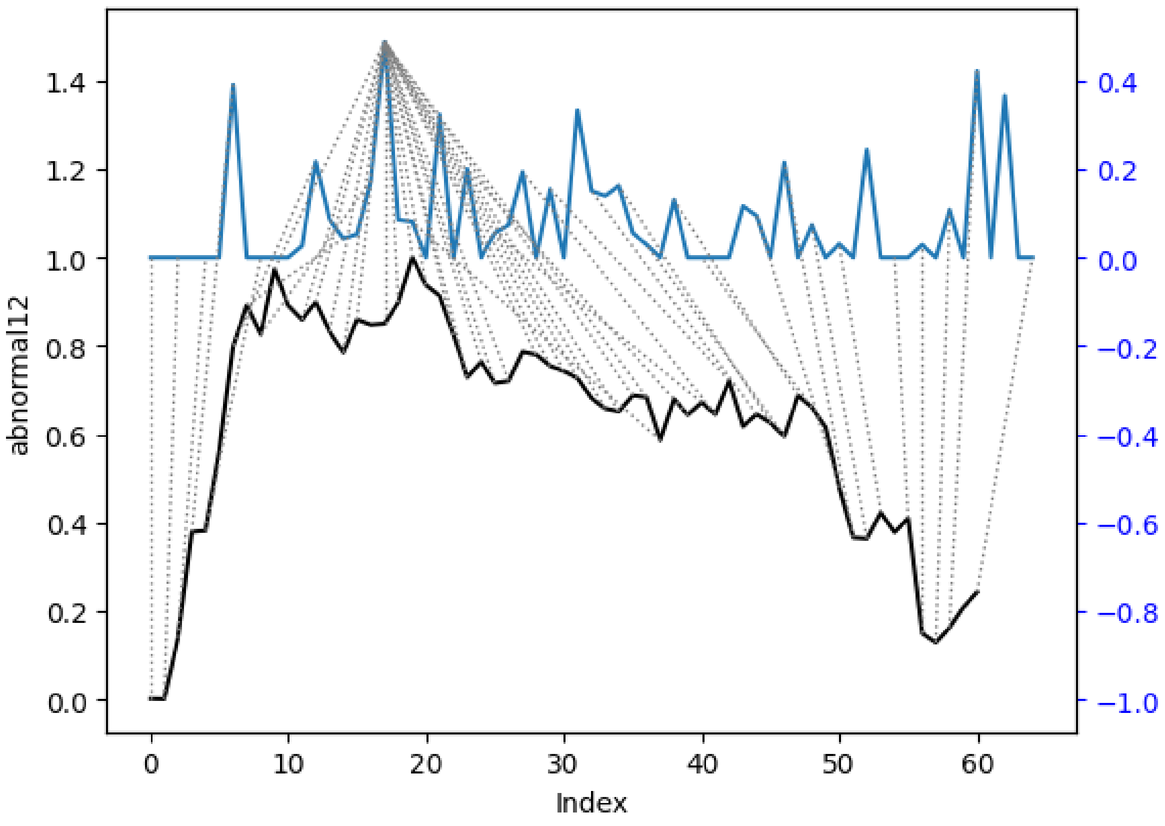
abnormal16
Index
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
60


abnormal15
Index
0
10
20
30
40
50
60
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

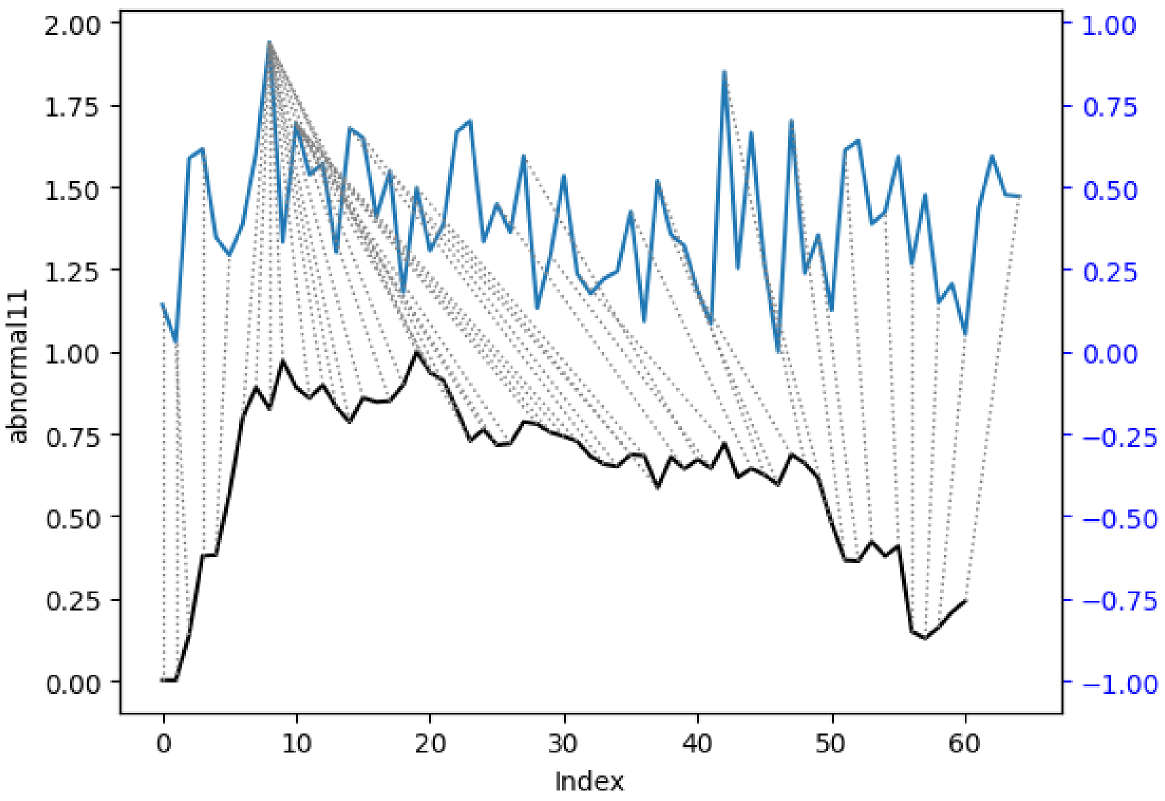

abnormal14
Index
0
10
20
30
40
50
60
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00



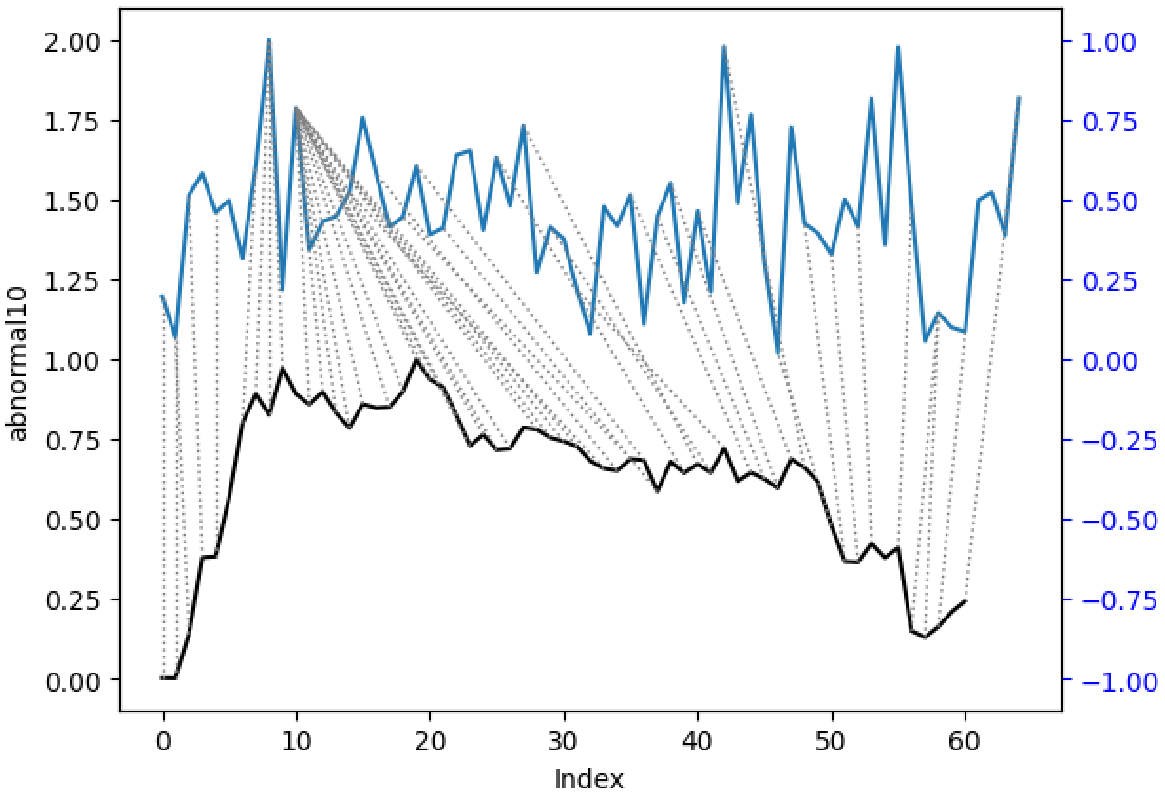
abnormal13
Index



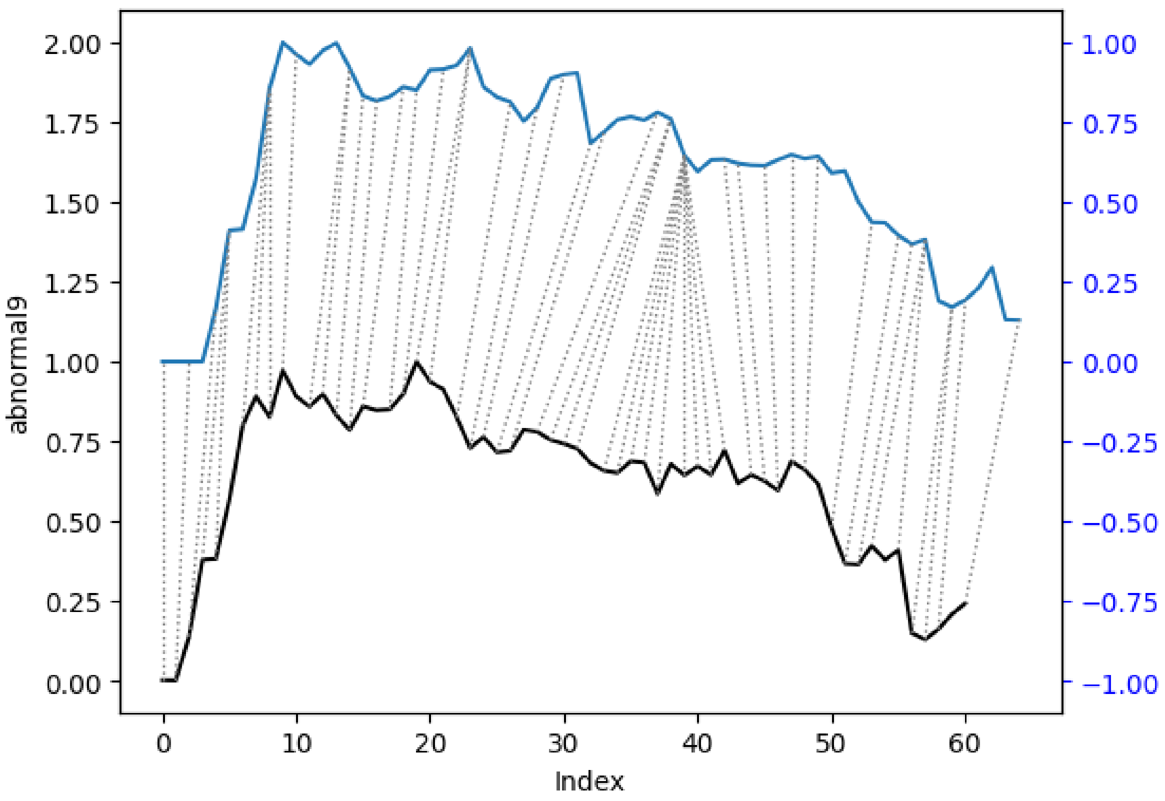
abnormal12
Index



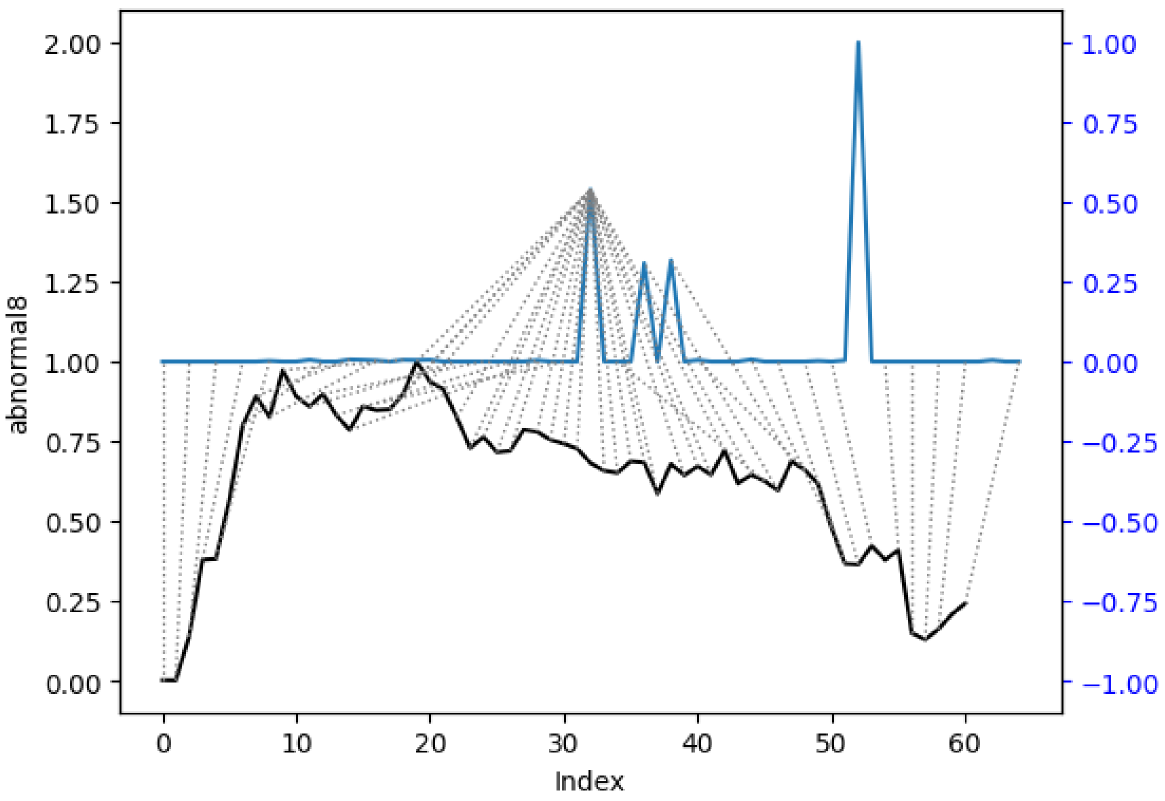
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
abnormal11
0
10
20
30
40
50
60
Index
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00



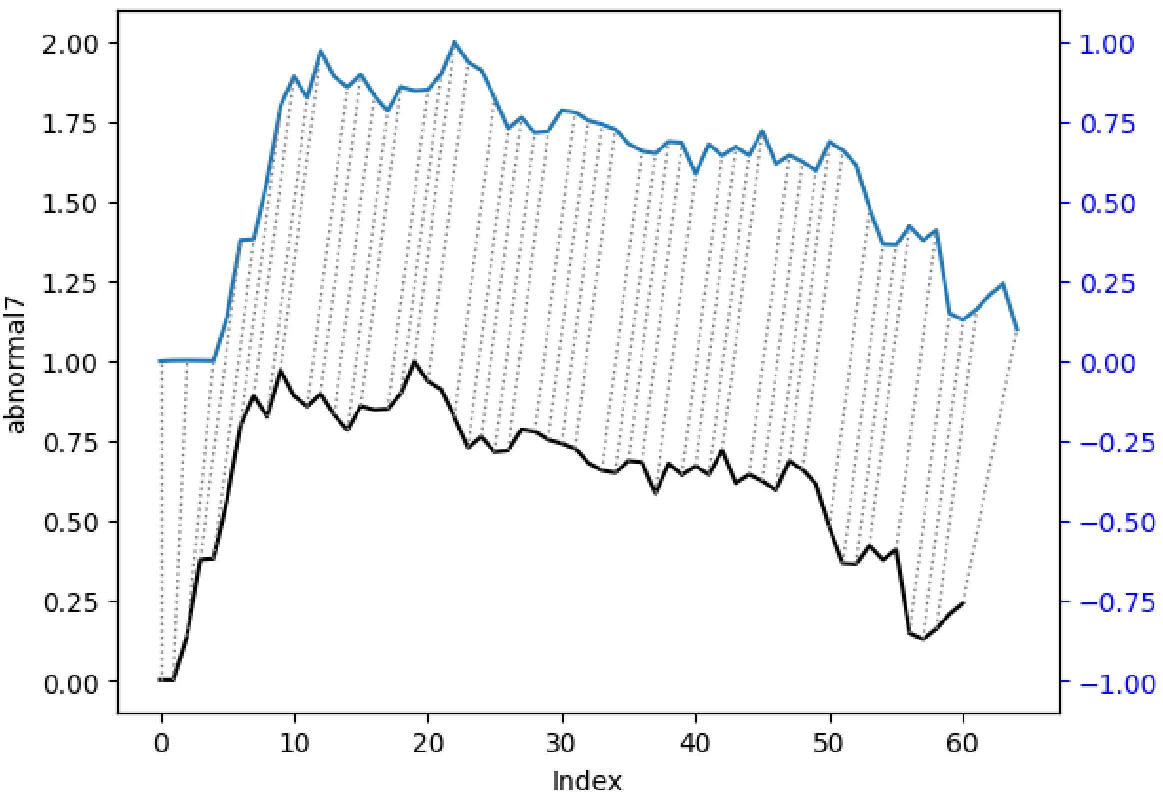
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
abnormal10
0
10
20
30
40
50
60
Index
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00



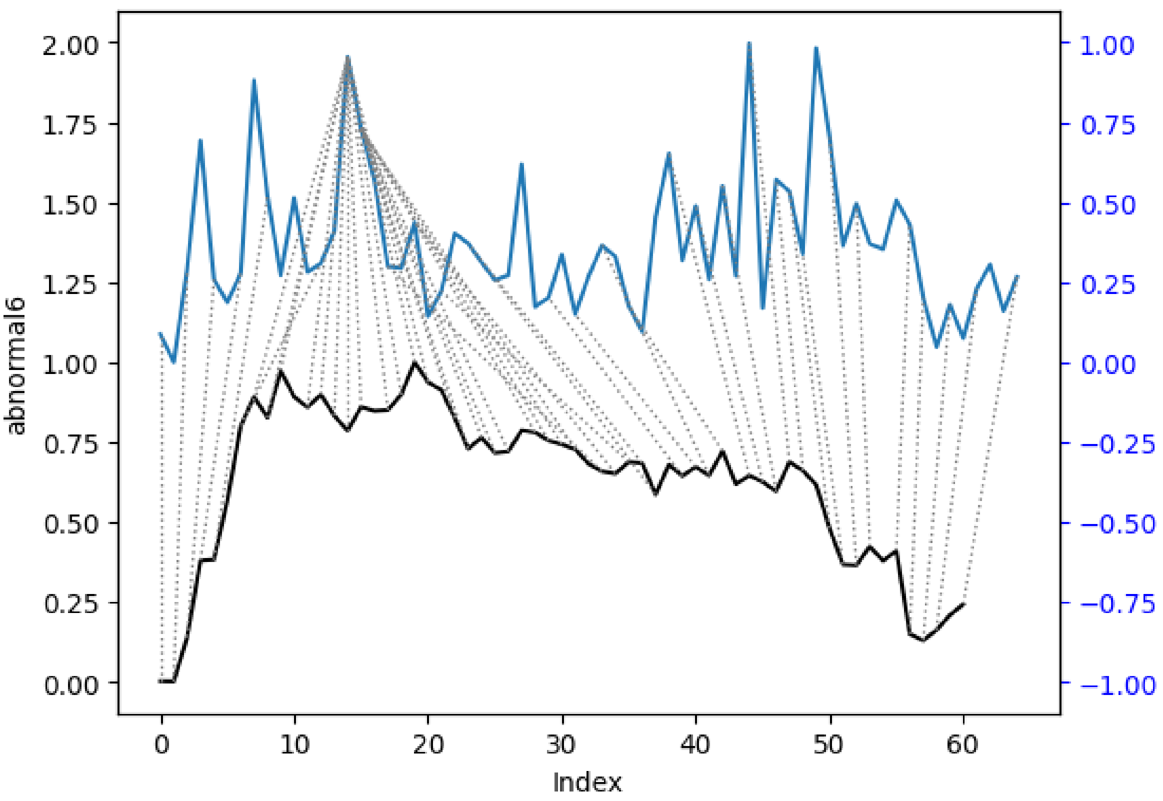
abnormal9
Index
0
10
20
30
40
50
60
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00



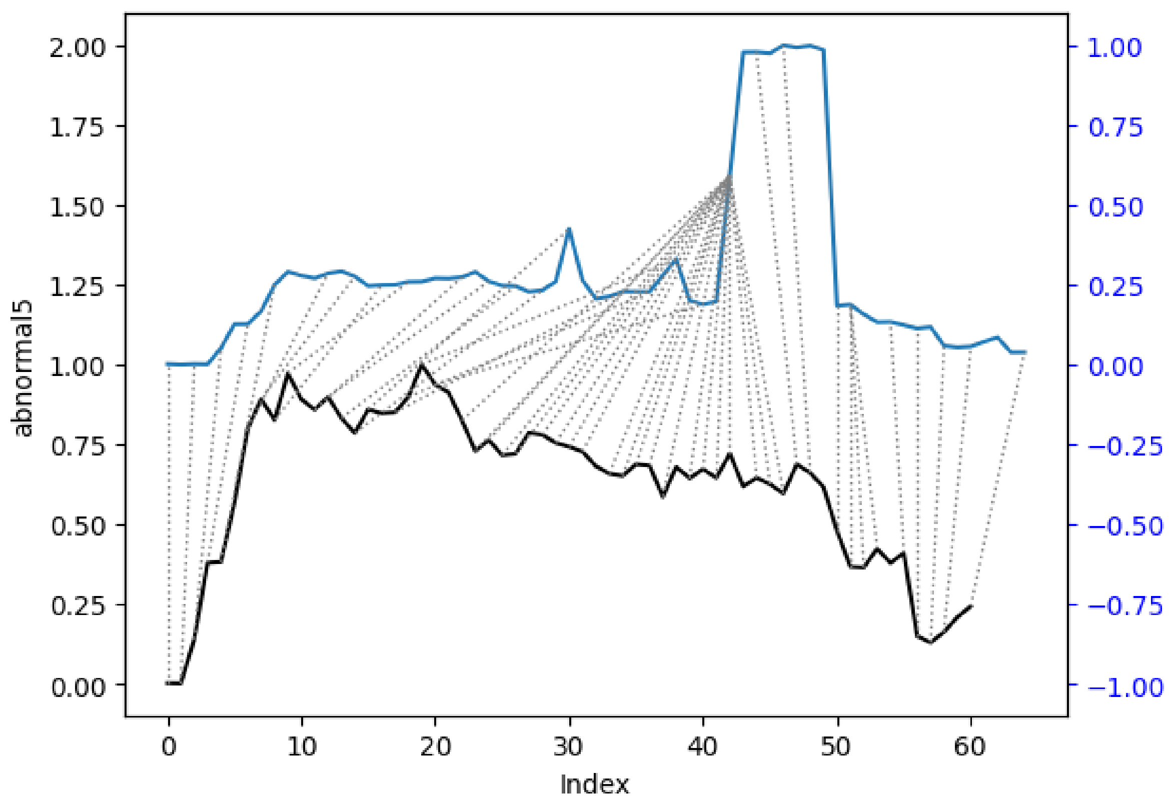
abnormal8
Index
0
10
20
30
40
50
60
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00



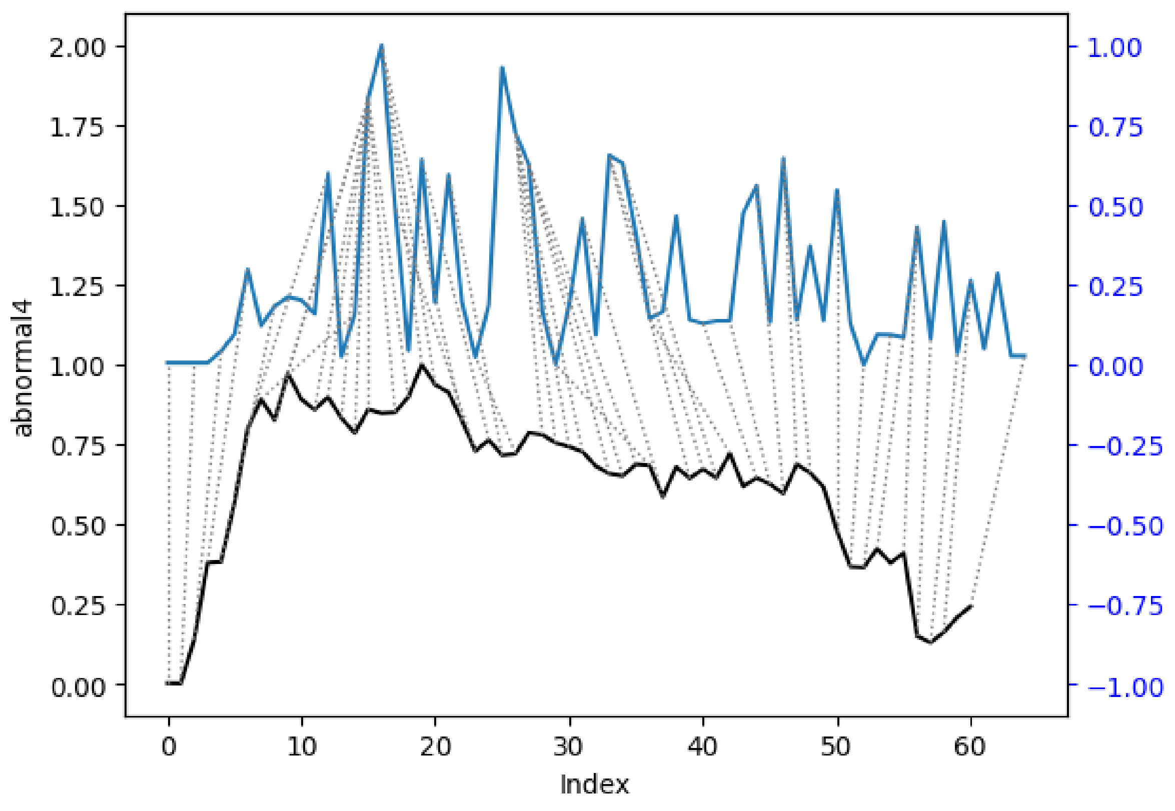
abnormal7
Index



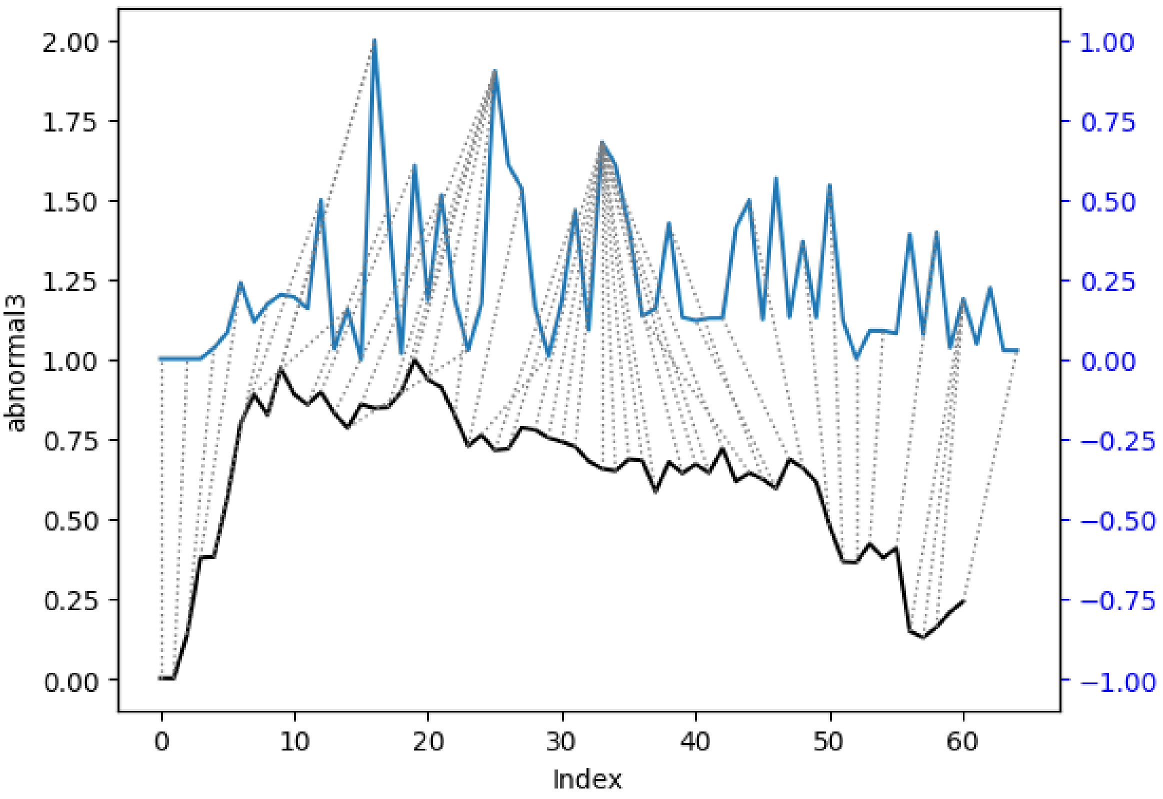
abnormal6
Index

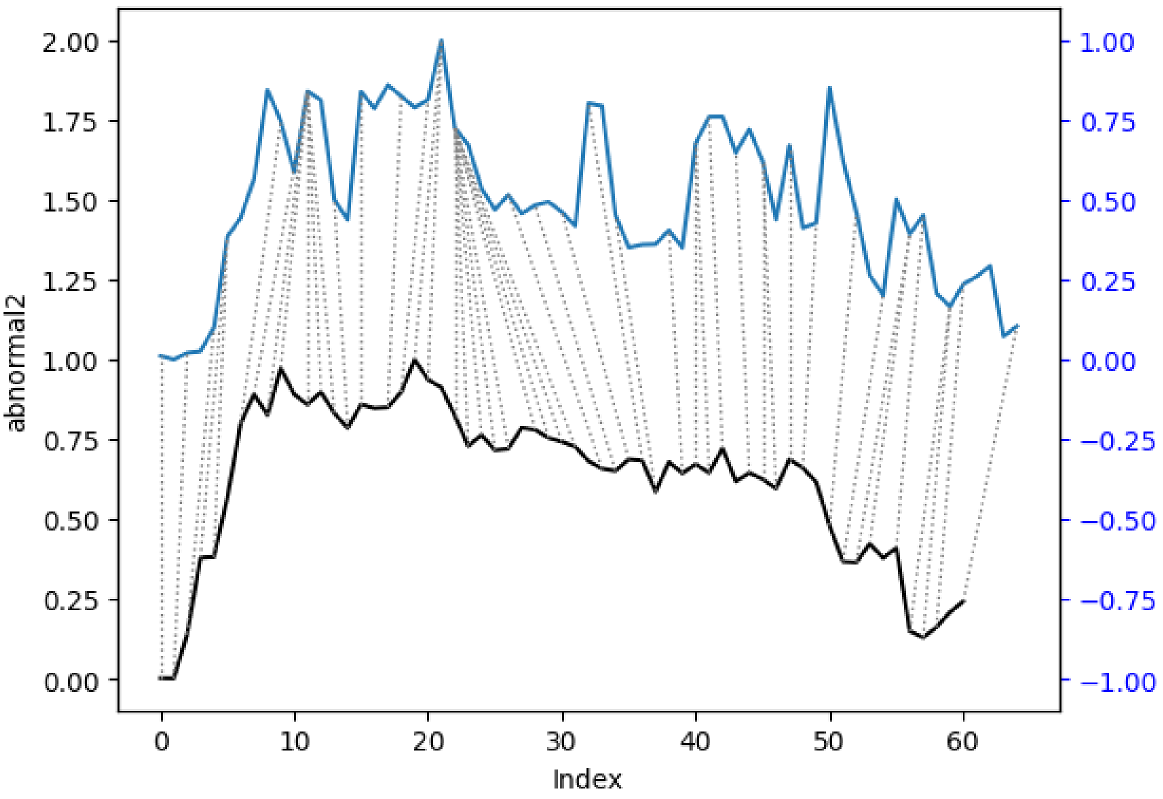


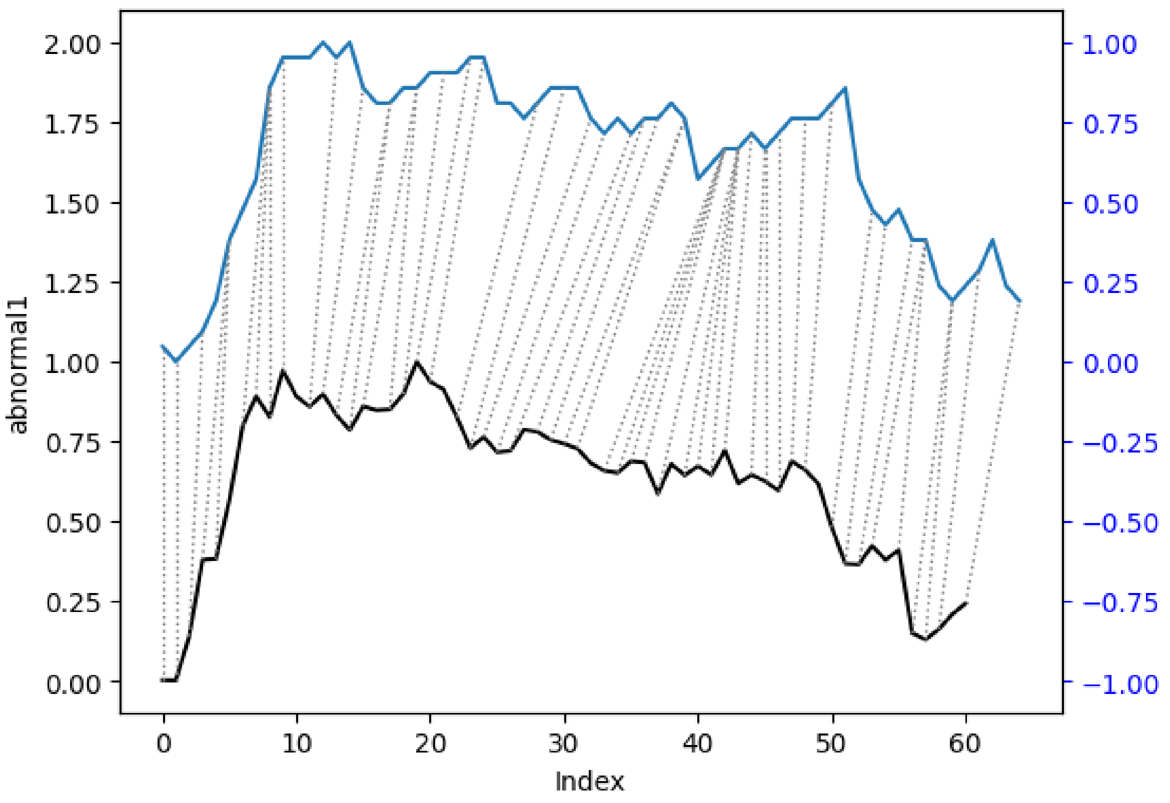
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
abnormal5
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
60
Index



2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
abnormal4
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
60
Index


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
abnormal3
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
60
Index


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
abnormal2
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
60
Index



abnormal1
Index
0
10
20
30
40
50
60
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00