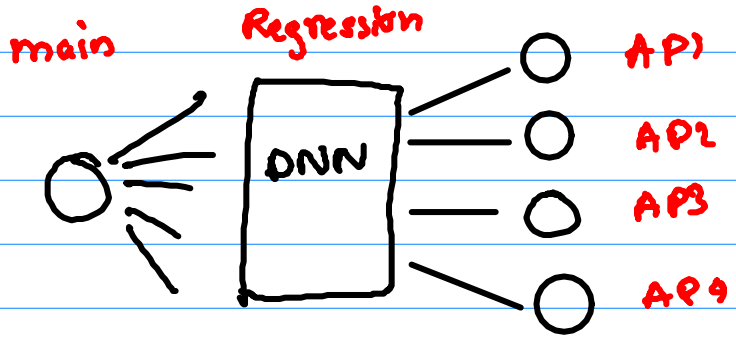
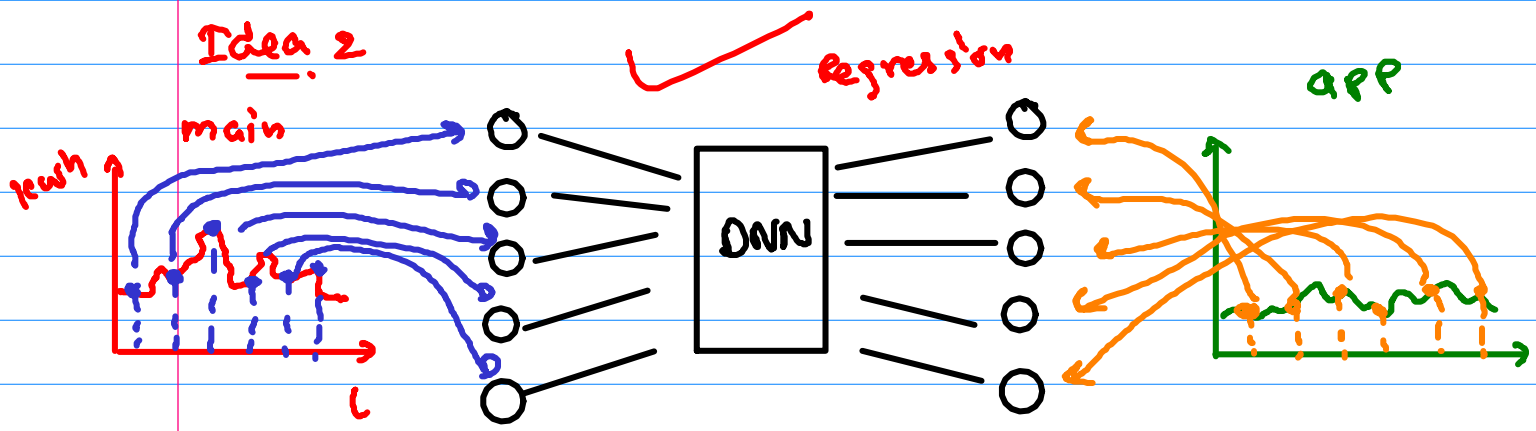
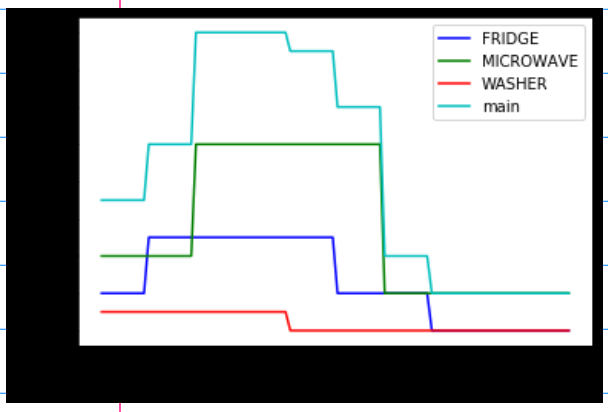
Idea 1

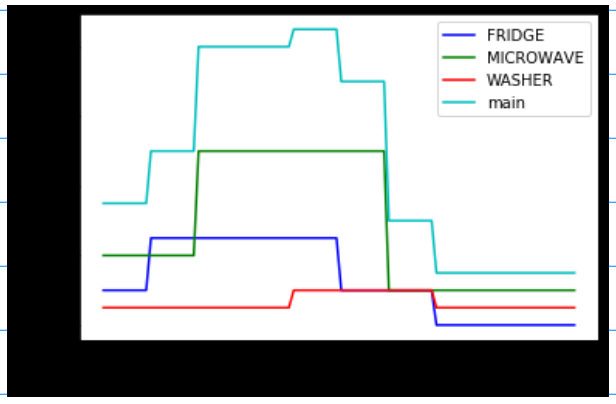


30	8,7,10,5
30	2,8,2,13
30	0,5,13,10

Idea 2

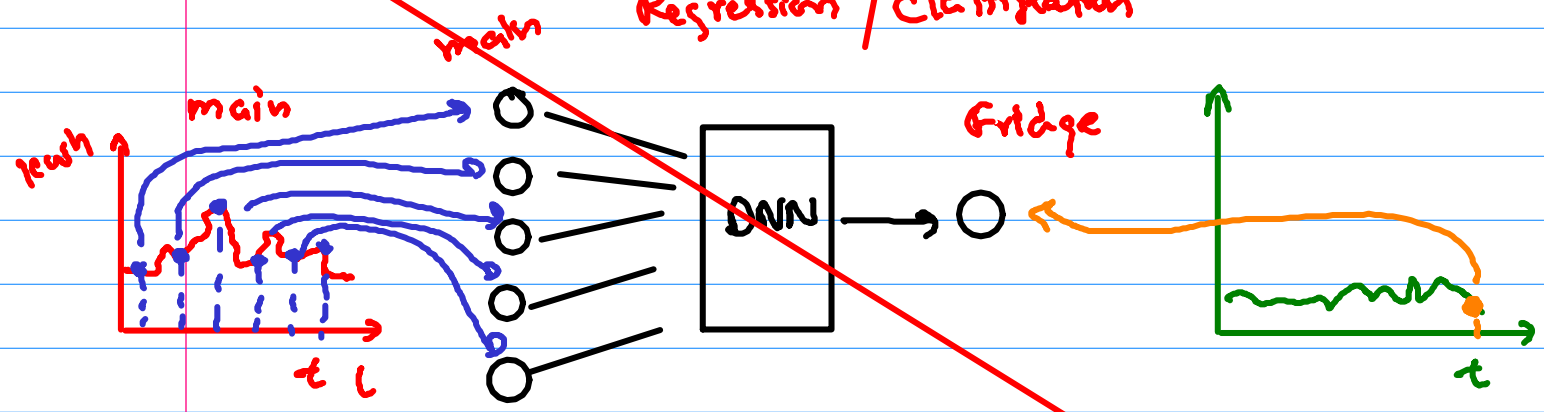






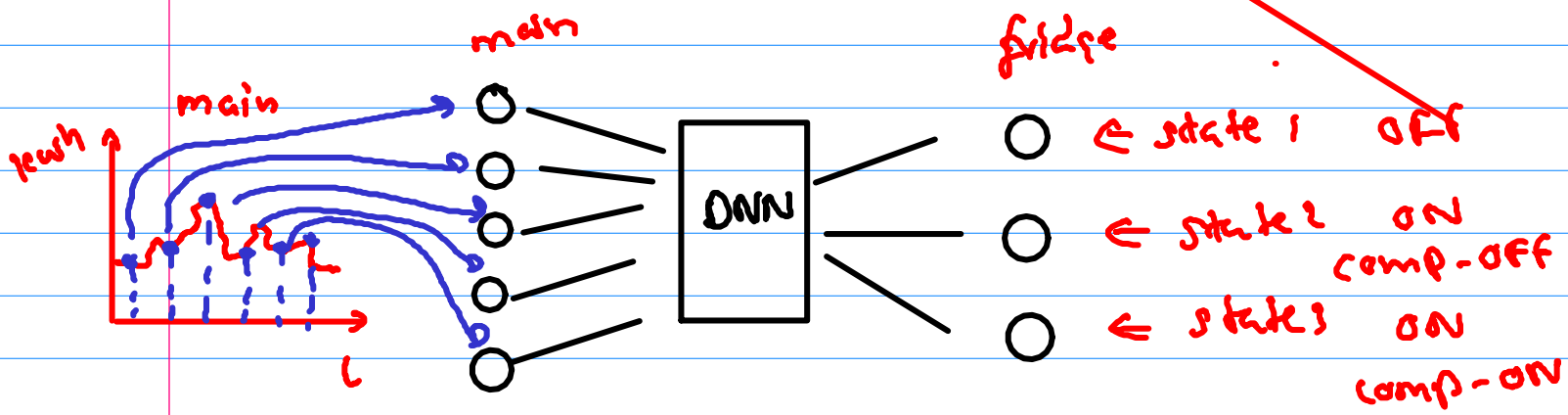
Idea 3

Regression / Classification



Idea 4

Classification



Data Preprocessing

1. Forward Filling

xxx933	6
xxx936	7
xxx939	7

→

xxx933	6
xxx934	6
xxx935	6
xxx936	6
xxx937	7
xxx938	7
xxx939	7

```python
for i in range(len(time)-1):
    if(time[i+1]-time[i]!=1):
        gap=time[i+1]-time[i]
        if(gap<20):
            for j in range(gap):
                new_time.append(time[i]+j)
                new_data.append(data[i])
    else:
        new_time.append(time[i])
        new_data.append(data[i])
```

i=0
i=1
i=5

```
1303132929 224.19
1303132930 225.57
1303132931 226.09
1303132932 222.74
1303132933 222.20
1303132934 222.11
1303132935 223.14
1303132938 222.64
1303132939 221.88
1303132940 223.60
1303132941 222.21
1303132942 222.82
1303132943 222.91
1303132944 222.81
```

new time = xx29, xx30, ⋮, xx39, xx35, xx36, xx37

new data = 224.19, ⋮, 222.11, 223.14, 223.19, 223.19

time ↑ data

new dataset =

main power	appliance power

```python
new_dataset=[]

for i,timestamp_main in enumerate(main_dataset[:,0]):
    available=np.where(appliance_dataset[:,0]=timestamp_main)[0]
    #checking whether the main timestamp is availble in the appliance[]
    if(avalable.size!=0):
        index=avalable[0]
        new_dataset.append([main_dataset[i,1],appliance_dataset[i,1]])
```

i=0
i=1
i=2

main

```
1303132932 222.74
1303132933 222.20
1303132934 222.11
1303132935 223.14
1303132936 223.17
1303132937 222.25
1303132938 222.64
1303132939 221.88
1303132940 223.60
1303132941 222.21
1303132942 222.82
1303132943 222.91
1303132944 222.81
1303132945 221.64
1303132946 222.94
```

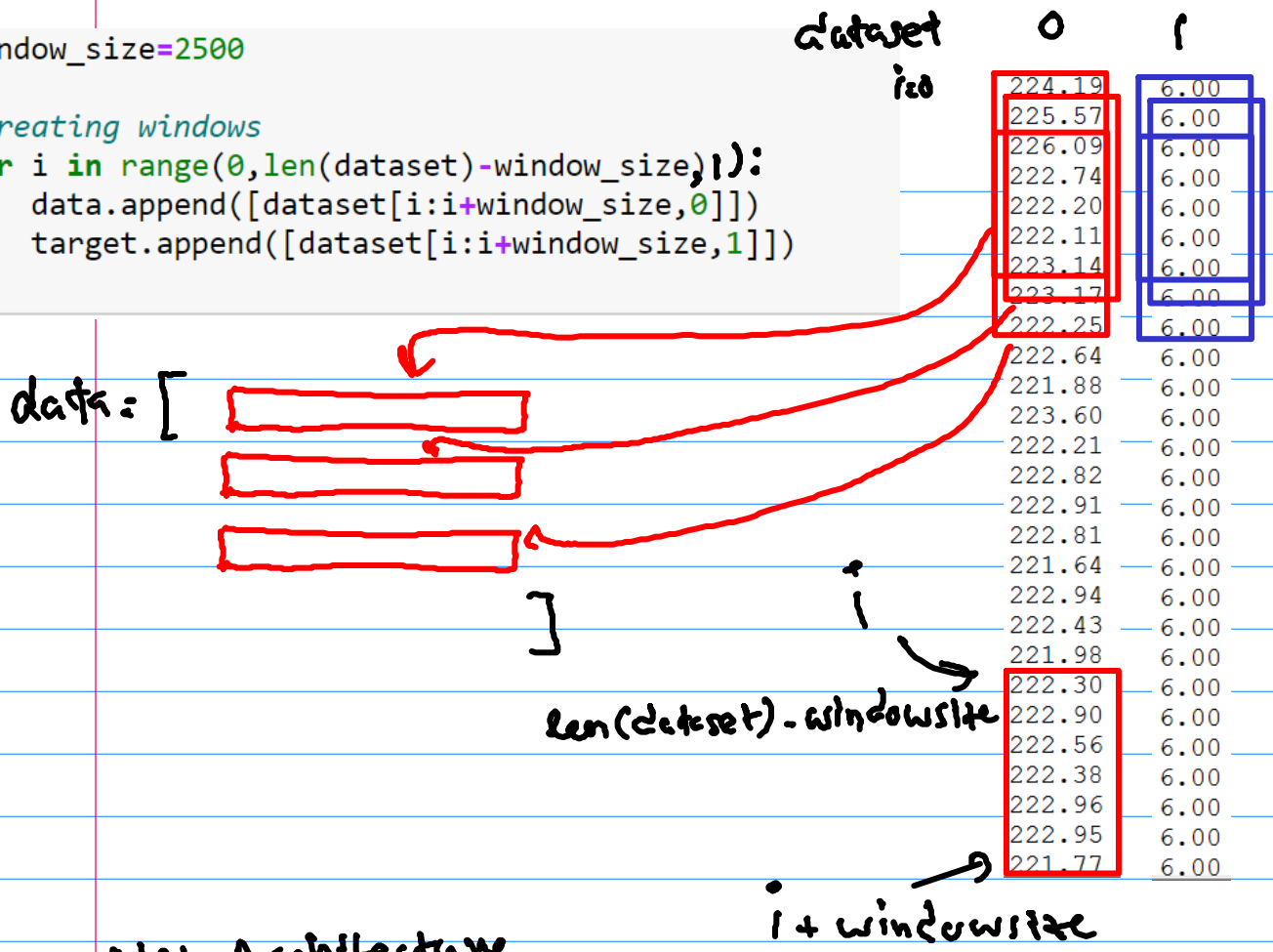
appliance

```
1303132933 6.00
1303132936 6.00
1303132940 6.00
1303132943 6.00
1303132946 6.00
1303132950 6.00
1303132953 6.00
1303132957 6.00
1303132960 6.00
1303132964 6.00
1303132967 6.00
1303132971 6.00
```

new dataset = [[222.2 , 6.00]]

```python
window_size=2500

#creating windows
for i in range(0,len(dataset)-window_size,1):
    data.append([dataset[i:i+window_size,0]])
    target.append([dataset[i:i+window_size,1]])
```



NN Architecture

2500

1.	Input (length T is appliance specific window size)
2.	Parallel 1D convolution with filter size 3, 5, and 7 respectively, stride=1, number of filters=32, activation type=linear, border mode=same
3.	Merge layer which concatenates the output of parallel 1D convolutions
4.	Dense layer, output_dim=128, activation type=ReLU
5.	Dense layer, output_dim=128, activation type=ReLU
6.	Dense layer, output_dim=T, activation type=linear