

Міністерство освіти та науки України Львівський  
національний університет імені Івана Франка

# Звіт

## Лабораторна робота №3

Виконав Кравчук Олександр  
Перевірів Сінкевич О.О.

## Мета :

“Порахувати потужність(Вт) котла(чи якогось іншого опалювального пристрою) та спрогнозувати настільки необхідно збільшити/зменшити потужність опалювального пристрою на 1-3 кроки наперед використовуючи метод для прогнозування часових рядів ARIMA”

## Зауваження :

Систему з декількох пристроїв яку буде представлено нижче можна буде використовувати в технологіях реального часу за допомогою фреймворка FLASK(python) який в свою чергу буде підключений до Raspberry Pi - який вимірює температуру та вологість за допомогою сенсора



Система полягає в наступному, ми сенсором міряємо температуру на 100(інтервал сталий допустим 15 хв.) кроків вперед. Для кожного кроку вимірюємо потужність - Q. Берем перші 50 кроків та для них прогнозуємо потужність на крок вперед. Далі так продовжуємо циклічно для кожного кроку вимірюємо потужність, та для кожного наступного 50 кроку прогнозуємо. Це все підключене до Flask.

## Реалізація :

Все реалізовано на python використовуючи такі бібліотеки як numpy(для розв’язування систем),pandas(Для читання csv даних), sympy(для розв’язування диференціальних рівнянь).

Почнемо з ARIMA :

```

8     return function
9
10    def ARIMA_with_Prediction(step, x,y,p,q,d):
11        new_y = [0]
12        new_y1 = []
13
14        resut_arr = []
15        MA_plust_ar = 0
16
17        if d == 0:
18            new_x = []
19            new_y = []
20            new_x = x
21            new_y = y
22            my_x_x = 199
23            for i in range(step):
24                MA_res = MA_foresee(new_y,q)
25                AR_res = auto_reg(new_x,new_y,p=p)
26                MA_plust_ar = MA_res + AR_res
27                resut_arr.append(MA_plust_ar)
28                new_y.append(MA_plust_ar)
29                my_x_x +=1
30                new_x.append(my_x_x)
31            return resut_arr
32
33    # data = pd.read_csv("mainData.csv")

```

Тут реалізована Аріма .

```

def minMNM(a,b):
    function = 0
    w0, w1 = symbols('x y', real=True)

    for i in range(len(a)):
        function +=(w1*a[i] + w0 - b[i]) ** 2

    #print(function)

    function1w0 = diff(function,w0)
    function2w1 = diff(function,w1)

    #print(function1w0)
    # print(function2w1)

    function1w0 = str(function1w0)
    function2w1 = str(function2w1)

    result1 = re.findall(r'\d+',function1w0)
    result2 = re.findall(r'\d+',function2w1)

    for i in range(len(result1)):
        result1[i] = int(result1[i])

    for i in range(len(result2)):
        result2[i] = int(result2[i])

    matrix = [result1[0:2],result2[0:2]]
    matrix_result = [result1[2],result2[2]]

    resultAll = np.linalg.solve(matrix,matrix_result)

    new_w0, new_w1 = resultAll

    return new_w0, new_w1

```

Тут реалізований метод найменших квадратів для оптимізації аріми.

```

import new_arima
import newq
import pandas as pd

paht_to_data = "mainData.csv"

def CountQAndArima(step,path,p,q,d=0):
    data = pd.read_csv(path)
    temp_y = []
    temp_y.extend(data['value'])
    y = []
    for i in range(len(temp_y)):
        temp = temp_y[i]
        y.append(temp)

    q_list = newq.gloabalTemperature(y,1)
    x = []
    for i in range(len(q_list)):
        x.append(i)

    result = new_arima.ARIMA_with_Prediction(step=step,x=x,y=q_list,p=p,q=q,d=0)
    return result

result = CountQAndArima(2,paht_to_data,1,7)
print("Predict = ",result)

```

Ось тут поєднання знаходження потужності та прогнозування.

```

File Edit View Search Terminal Help
ostap@ostap:~/Documents/LNU/kiberg/QWithArima$ python3 main_laboratorna2.py
Predict = [0.9806108368302218, 0.982208968511009]
ostap@ostap:~/Documents/LNU/kiberg/QWithArima$

```

Результати роботи.

## Висновок :

В цій лабораторній роботі я ознайомився з написанням коду для машин реального часу на опрацювання даних.