Projekt zaliczeniowy

Predykcja wartości opublikowanej wydajności względnej procesorów przy użyciu modelu regresji systemu ANFIS

Damian Gortych

1. Opis danych

Do mojego projektu użyłem danych na temat wydajności procesorów „Relative CPU Performance Data” pobranych ze strony UCI.

	vendor name	model	MCYT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP	ERP
0	adviser	32/60	125	256	6000	256	16	128	198	199
1	amdahl	470v/7	29	8000	32000	32	8	32	269	253
2	amdahl	470v/7a	29	8000	32000	32	8	32	220	253
3	amdahl	470v/7b	29	8000	32000	32	8	32	172	253
4	amdahl	470v/7c	29	8000	16000	32	8	16	132	132
...	...	...	...	...	...	...	...	...	...	...
204	sperry	80/8	124	1000	8000	0	1	8	42	37
205	sperry	90/80-model-3	98	1000	8000	32	2	8	46	50
206	sratus	32	125	2000	8000	0	2	14	52	41
207	wang	vs-100	480	512	8000	32	0	0	67	47
208	wang	vs-90	480	1000	4000	0	0	0	45	25

Zestaw danych składa się z następujących atrybutów:

1. “vendor name” : nazwa producenta
2. ”model”: nazwa modelu
3. ”MYCT”: czas cyklu maszyny w nanosekundach (liczba całkowita)
4. ”MMIN”: minimalna pamięć główna w kilobajtach (liczba całkowita)
5. ”MMAX”: maksymalna pamięć główna w kilobajtach (liczba całkowita)
6. ”CACH” : pamięć podręczna w kilobajtach (liczba całkowita)
7. “CHMIN”: minimalna liczba kanałów w jednostkach (liczba całkowita)
8. “CHMAX”: maksymalna liczba kanałów w jednostkach (liczba całkowita)
9. “PRP”: opublikowana wydajność względna (liczba całkowita)
10. “ERP”: estymowana względna wydajność z oryginalnego artykułu (liczba całkowita)

2. Przetwarzanie wstępne oraz podstawowe statystyki.

Moim celem będzie predykcja wartości opublikowanej wydajności względnej „PRP”.

Na początek usuwam ze zbioru zbędne atrybuty opisowe „vendor name” oraz „model” oraz atrybut „ERP”.

Dla pozostałych danych obliczam podstawowe statystyki.

```
In [12]: data = data.iloc[:,2:9]
         data.describe()
```

Out[12]:

	MCYT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP
count	209.000000	209.000000	209.000000	209.000000	209.000000	209.000000	209.000000
mean	203.822967	2867.980861	11796.153110	25.205742	4.698565	18.267943	105.622010
std	260.262926	3878.742758	11726.564377	40.628722	6.816274	25.997318	160.830733
min	17.000000	64.000000	64.000000	0.000000	0.000000	0.000000	6.000000
25%	50.000000	768.000000	4000.000000	0.000000	1.000000	5.000000	27.000000
50%	110.000000	2000.000000	8000.000000	8.000000	2.000000	8.000000	50.000000
75%	225.000000	4000.000000	16000.000000	32.000000	6.000000	24.000000	113.000000
max	1500.000000	32000.000000	64000.000000	256.000000	52.000000	176.000000	1150.000000

Sprawdzam brakujące dane.

```
In [13]: pd.isnull(data).mean()
```

```
Out[13]: MCYT     0.0
         MMIN     0.0
         MMAX     0.0
         CACH     0.0
         CHMIN    0.0
         CHMAX    0.0
         PRP      0.0
         dtype: float64
```

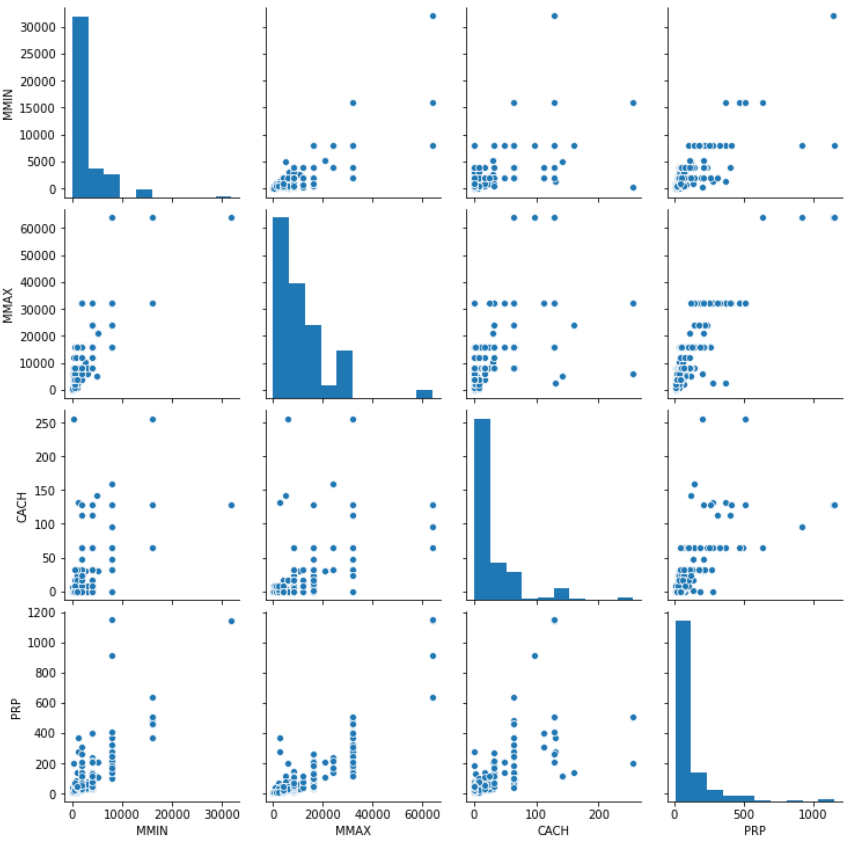
Następnie tworze macierz korelacji.

```
In [14]: sns.heatmap(data.corr().abs(),annot=True)
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x13cc0a18cc0>
```

Na jej podstawie decyduje, że do mojej predykcji użyje trzech zmiennych z największą i zadowalającą korelacją ze zmienną na wyjściu czyli „MMIN” , „MMAX’ , „CACH” .

Dla pozostałych danych tworze pairplot.

```
In [16]: import seaborn as sns
         sns.pairplot(data)
Out[16]: <seaborn.axisgrid.PairGrid at 0x13cc0132a20>
```



Z powodu niewielkiej ilości danych, wykresy nie są idealne, natomiast można zauważyć liniową zależność.

Dodatkowo dołączam zakresy zmiennej wyjściowej dostępne wraz z pobranymi danymi.

0-20	31
21-100	121
101-200	27
201-300	13
301-400	7
401-500	4
501-600	2
above 600	4

3. Teoretyczny opis wykorzystywanego modelu ANFIS.

Wykorzystywany przeze mnie w kolejnych etapach model ANFIS czyli adaptacyjny system wnioskowania neuro-rozmytego jest rodzajem sztucznej sieci neuronowej bazujący na rozmytym wnioskowaniu Takagi-Surgeno.
Modele rozmyte cechują się mniejszą potrzebą otrzymywania informacji co jest ich istotną zaletą w porównaniu z metodami probabilistycznymi.
Wykorzystywane są one w wielu różnych dziedzinach takich jak ekonomia lub przetwarzanie obrazów w problemach klasyfikacyjnych i regresyjnych.
W mojej pracy posłuży mi do przewidzenia wartości wydajności

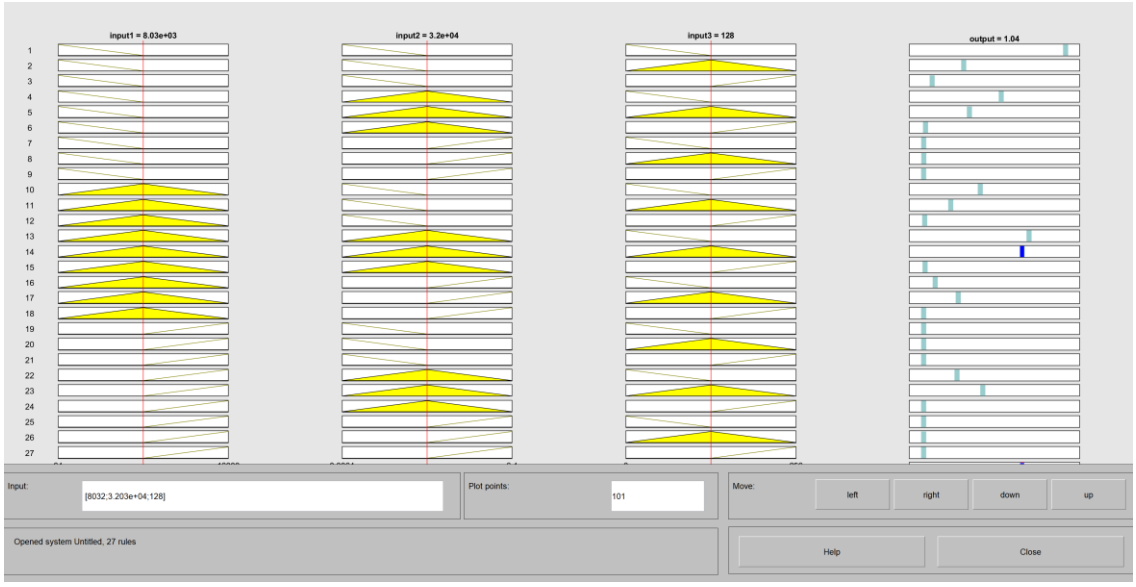
4. Stworzenie modeli ANFIS.

Wczytuje dane, a następnie dziele je na 4 zbiory testowe i treningowe, które posłużą mi do walidacji krzyżowej.

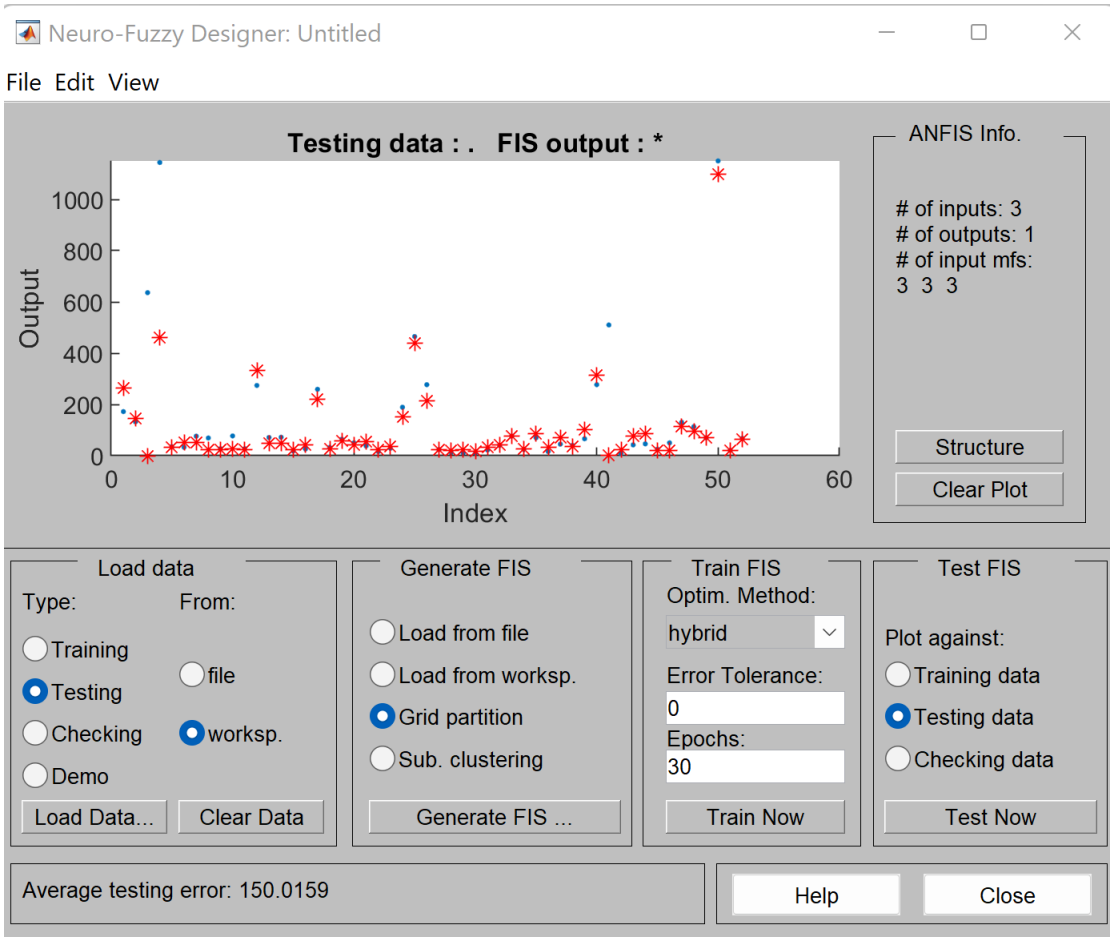
```
%% Wczytanie danych
close all;clear;clc;
data = readtable("data.csv");
data = table2array(data);
data(:,1) = [];
cv = cvpartition(size(data,1),'KFold',4);
idx1 = cv.test(1);
idx2 = cv.test(2);
idx3 = cv.test(3);
idx4 = cv.test(4);

%%
% Separate to training and test data
dataTrain1 = data(~idx1,:);
dataTest1  = data(idx1,:);
dataTrain2 = data(~idx2,:);
dataTest2  = data(idx2,:);
dataTrain3 = data(~idx3,:);
dataTest3  = data(idx3,:);
dataTrain4 = data(~idx4,:);
dataTest4 = data(idx4,:);
%%
```

1. Model 1

Typ funkcji : trójkątna
Liczba funkcji : 3 na każdym inpucie
Liczba reguł : 27




```
Minimal training RMSE = 24.6182
```

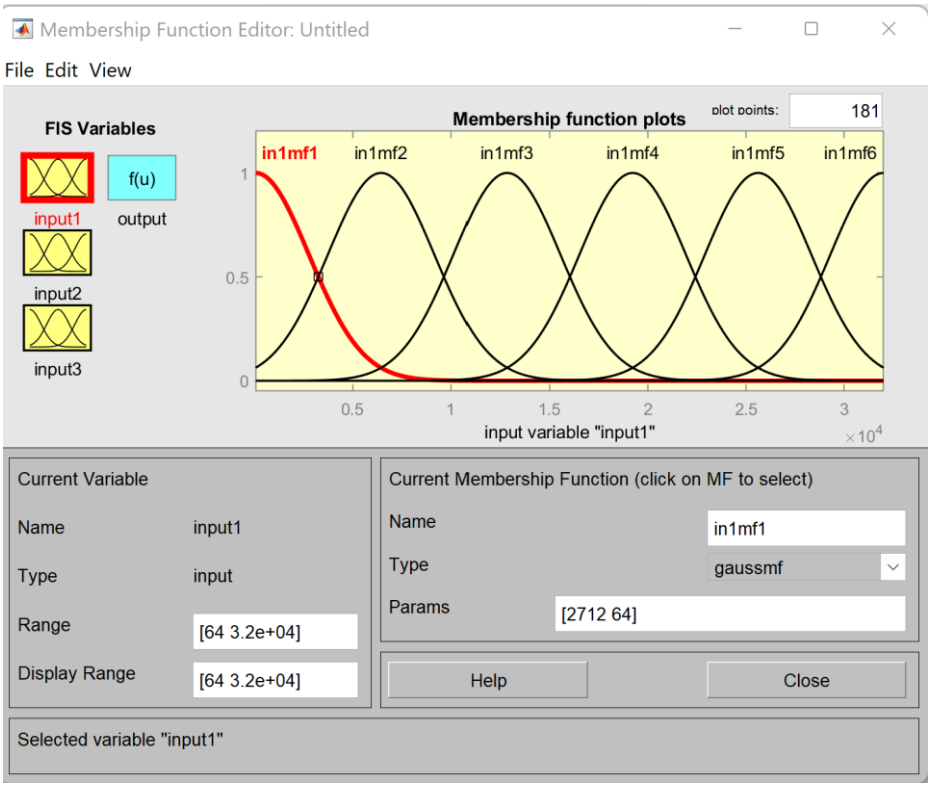


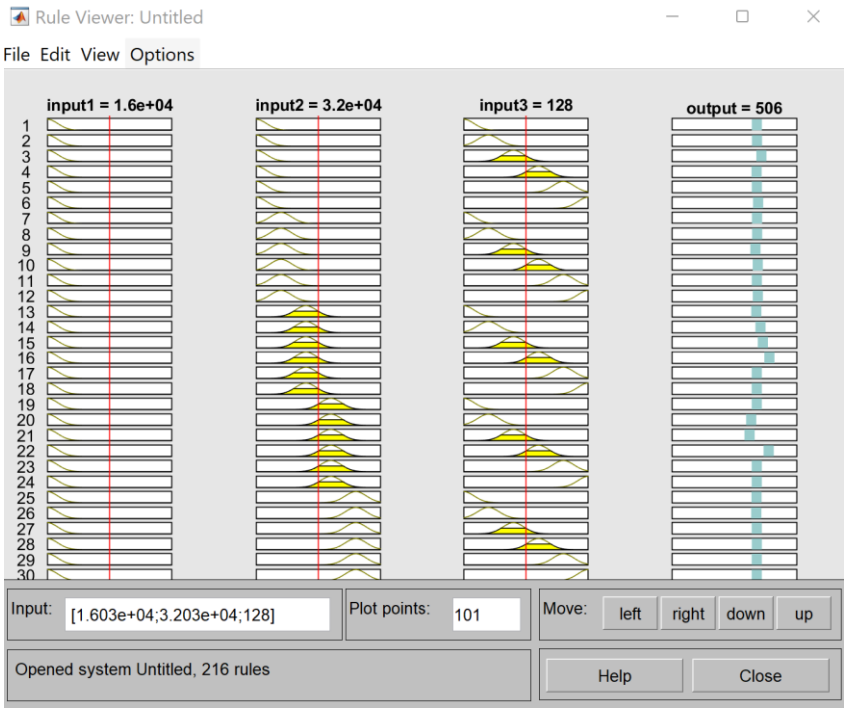
Wyniki walidacji krzyżowej :
RMSE test = 151.14
RMSE train = 24.91

W tym modelu otrzymaliśmy niezadowalające wyniki zarówno dla zbioru treningowego jak i testowego. Nie jest on optymalny dla naszych danych.

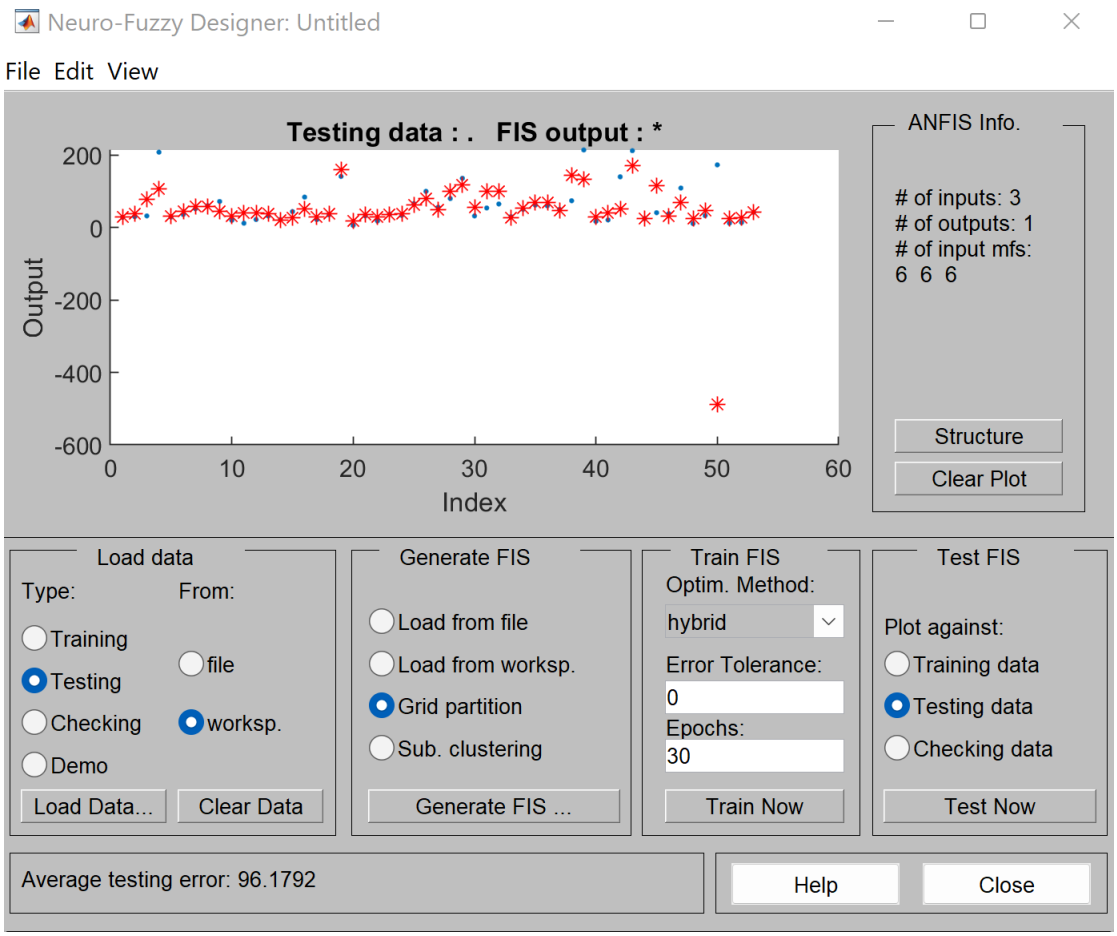
2. Model 2

Typ funkcji : gauss
Liczba funkcji : 6 na każdym inpucie
Liczba reguł : 216





```
Minimal training RMSE = 21.3379
```

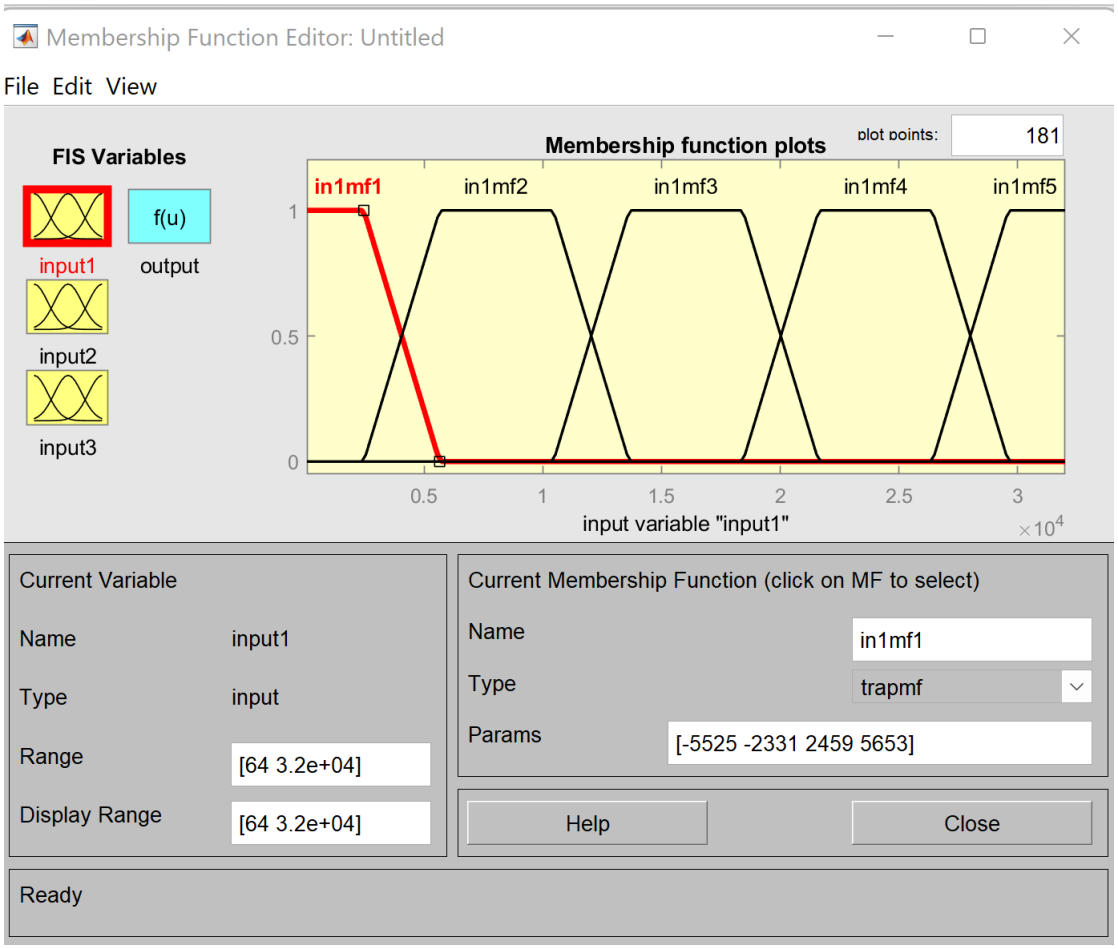


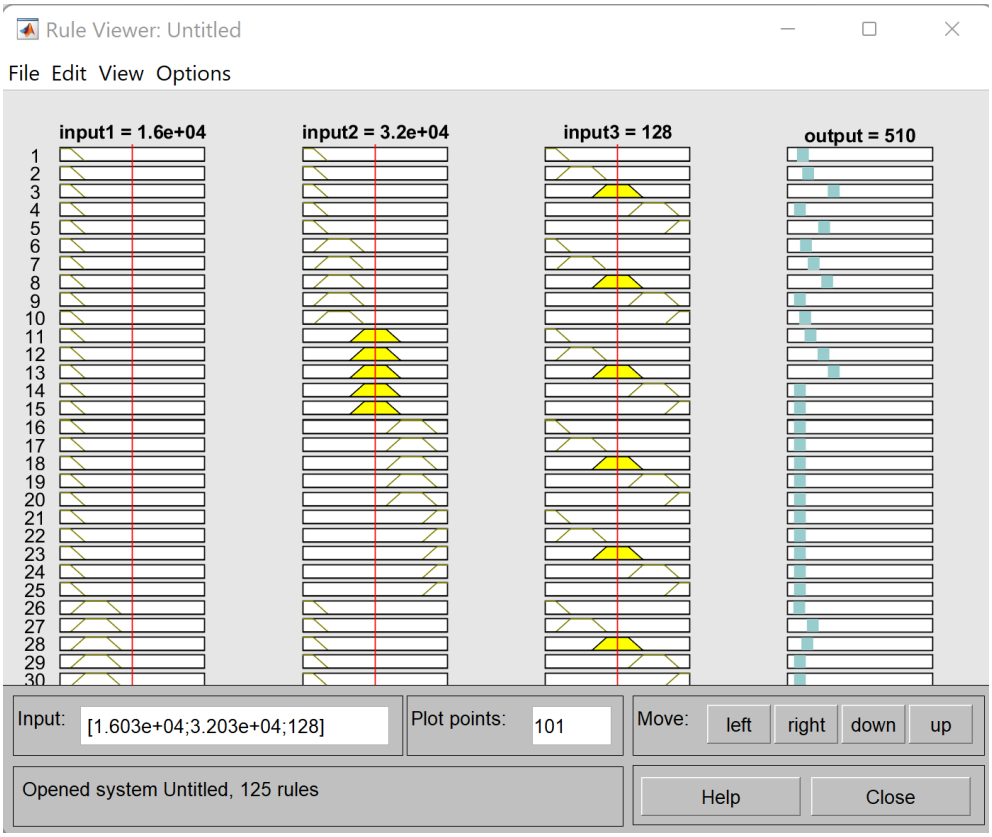
Wyniki walidacji krzyżowej :
RMSE test = 98.11
RMSE train = 22.01

W tym modelu otrzymaliśmy najlepsze wyniki dla zbioru treningowego, jednakże model okazał się nieodpowiedni w predykcji zbioru testowego.

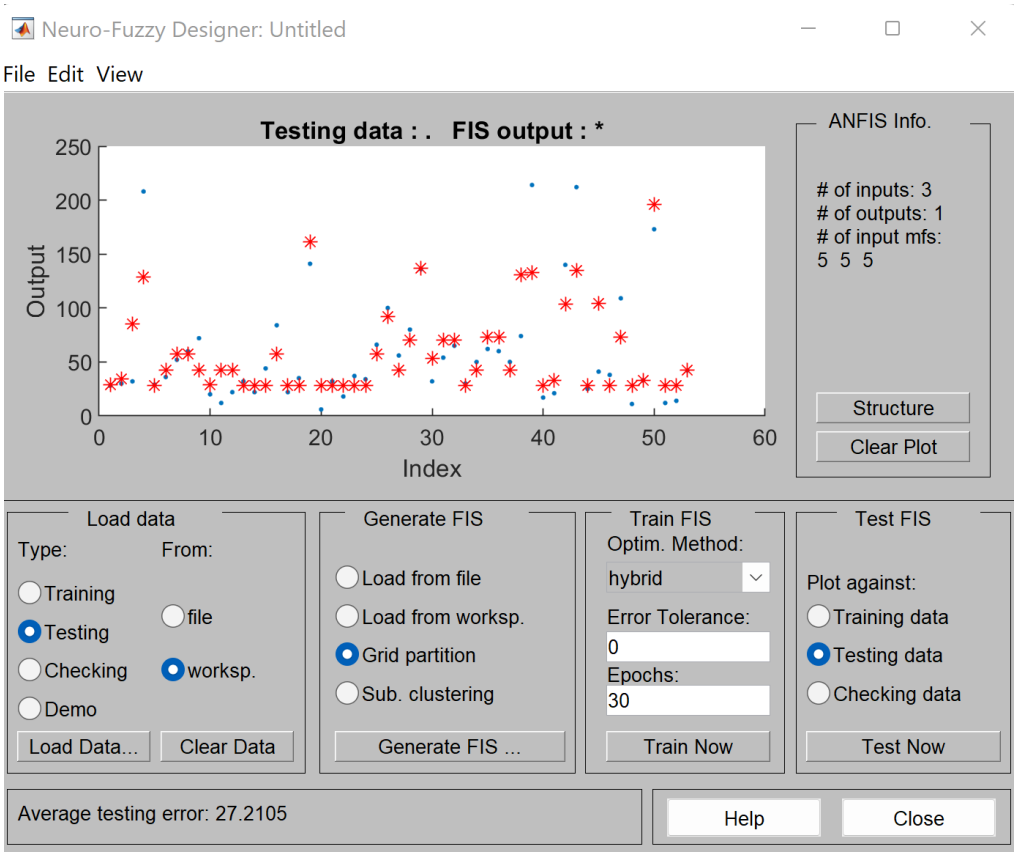
3. Model 3

Typ funkcji : trapez
Liczba funkcji : 5 na każdym inpucie
Liczba reguł : 125


Membership Function Editor: Untitled
File Edit View
FIS Variables
input1
output
input2
input3
f(u)
Membership function plots
plot points:
181
in1mf1
in1mf2
in1mf3
in1mf4
in1mf5
1
0.5
0
0.5
1
1.5
2
2.5
3
input variable "input1"
×10^4
Current Variable
Name
input1
Type
input
Range
[64 3.2e+04]
Display Range
[64 3.2e+04]
Current Membership Function (click on MF to select)
Name
in1mf1
Type
trapmf
Params
[-5525 -2331 2459 5653]
Help
Close
Ready


Rule Viewer: Untitled
File Edit View Options
input1 = 1.6e+04
input2 = 3.2e+04
input3 = 128
output = 510
Input:
[1.603e+04;3.203e+04;128]
Plot points:
101
Move:
left
right
down
up
Opened system Untitled, 125 rules
Help
Close

```
Minimal training RMSE = 25.3846
```



Wyniki walidacji krzyżowej :
RMSE test = 27.92
RMSE train = 25.52

Model 3 okazał się najlepszy dla naszych danych, predykcja zarówno na zbiorze treningowym jak i testowym dała bliskie rezultaty.

5. Wnioski

Analizując różne modele największą obserwacją, był ich problem w przewidywaniu zbioru testowego, pomimo zadowalających wyników na zbiorze treningowym. Ostatecznie najlepszym modelem okazał się model 3, który najlepiej poradził sobie z problemem i uzyskał zadowalające wyniki.