```
import keras
from keras.preprocessing.sequence import pad_sequences
import math
import collections
import numpy as np

length_max = 1001 #max length of NN 1
length_max _ 2 = 100 #max length of NN 2
length_max _ 3 = 20 #max length of NN 3
splitted_length=500 #length of the splitted sequences

def kolmo (seq, counter = 1, complexity = 0) :
    if (len (seq[0]) < length_max) :
        if (len (seq[0]) > length_max _ 2) :
            sequence = pad_sequences (seq, value = -1, maxlen = length_max)
            complexity = NN_ 1000. predict (sequence) + math.log (counter, 2) #BDMNN
        elif (len (seq[0]) <= length_max _ 2 and len (seq[0]) > length_max _ 3) :
            sequence = pad_sequences (seq, value = -1, maxlen = length_max _ 2)
            complexity = NN_ 100. predict (sequence) + math.log (counter, 2)
        elif (len (seq[0]) <= length_max _ 3 and len (seq[0]) > 11) :
            sequence = pad_sequences (seq, value = -1, maxlen = length_max _ 3)
            complexity = NN_ 20. predict (sequence)[0] + math.log (counter, 2)
        else :
            complexity = 2.285794 #complexity of sequences 0 and 1
    elif (len (seq[0]) >= length_max) :
        splitted = np.array_split (seq[0], math.ceil (len (seq[0])/(splitted_length)))
        splitted_tuple = map (tuple, splitted)
        counts = collections.Counter (splitted_tuple) #count repeated sequences
        complexity_list = list (map (kolmo, [splitted], counts.values ()))
        complexity = sum (complexity_list[0])
    else :
        complexity = 2.285794 #complexity of sequences 0 and 1
    return complexity
```