Tikrit Journal of Pure Science Vol. 28 (3) 2023

TJPS

https://doi.org/10.25130/tjps.v28i3.1437

المجلات الاكاديمية العراقية

IASJ

Tikrit Journal of Pure Science

ISSN: 1813 – 1662 (Print) --- E-ISSN: 2415 – 1726 (Online)

Journal Homepage: http://tjps.tu.edu.iq/index.php/j

Compute Nano topology by used the programing language python

Nawfal N. Abed , Ali.A. Shihab

Department of Mathematics, College of Education for Pure Sciences, Tikrit University, Tikrit, Iraq

ARTICLE INFO.

Article history:

-Received: 27 / 8 / 2022

-Received in revised form: 18 / 9 / 2022

-Accepted: 19 / 10 / 2022

-Final Proofreading: 2 / 6 / 2023

-Available online: 25 / 6 / 2023

Keywords: Nano Topology; Python;semi ;alpha;beta;gamma;regular;pre;equivalence relations

Corresponding Author:

Name: Nawfal N. Abed

E-mail: nawfal.n.abed@st.tu.edu.iq

mailto:draliabd@tu.edu.iq

mailto:ali.abd82@yahoo.com

Tel:

ABSTRACT

The research aims to Compute the Nano topologies of any set, consisting of five or less elements, through the Python programming language, with finding the necessary algorithms for the solution steps, We explain the mechanism of action: In first we will find the equivalence relations on set, and then we must compute the nano topology, and end know who set was semi-open, alpha-open,beta-open Regular open ,pre open and gamma-open set .

حساب النانو توبولوجي باستخدام لغة البرمجة بايثون

نوفل نجم عبد ، علي عبد المجيد شهاب

قسم الرياضيات ، كلية التربية للعلوم الصرفة ، جامعة تكريت ، تكريت ، العراق

الملخص

يهدف البحث إلى حساب النانو توبولوجي لأي مجموعة تتكون من خمسة عناصر أو أقل ، من خلال لغة البرمجة بايثون، مع إيجاد الخوارزميات اللازمة لخطوات الحل، هنا نشرح آلية العمل: في البداية سنجد صفوف التكافؤ في المجموعة ، ومن ثم يجب علينا حساب النانو توبولوجي، ونعرف في النهاية من الذي كان منها سمي أوبن، ألفا أوبن، بيتا أوبن، وبري أوبن، و جاما أوبن ودلتا أوبن.

I. Introduction

To calculate the possible equivalence relations and Nano topology on the set , we will first compute The equivalence relations and then compute the lower approximation and the upper approximation And the boundary region and then compute the Nano topology and calculate the Nano open set in set S and calculate the nano interior(NInt(S)) and then calculate the closure of set (NCl(S)) and then calculate Nano interior of Nano closure (NInt(NCl(S))) and we calculate Nano closure of Nano Interior (NCl(NInt(S))) and then we calculate(NInt(NCl(NInt(S)))) and then calculate

https://doi.org/10.25130/tjps.v28i3.1437

(NCl(NInt(NCl(S)))) and so all that to Know who set was semi open ,alpha open,beta open and gamma open

1. One nano topology

1.1 Definition [1]

Let *U* be a non-empty finite set of objects called the *universe* and *R* be an *equivalence relations* on *U* named as *in discernibility relation*.

Elements belonging to the same equivalence class are said to be *indiscernible* with one other. The pair *(U,R)* is said to be *approximation space*. Let X ⊆ U.

1) The *lower approximation* of *X* with respect to *R* is the set of all objects, which can be cer-
tain classified as *X* with respect to *R* and is denoted by $LR(x)$ and defined by $LR(x)=\cup_{x\in U}$ {R(x) : R(x) ⊆ X} where *R(x)* denotes the equivalence class determined by *X*.

2) The *upper approximation* of *X* with respect to *R* is the set of all objects, which can be
possibly classified as *X* with respect to *R* and is defined by $UR(x) = \cup_{x\in U}$ $\{R(x):R(x)\cap X\neq\phi\}$.

3) The *boundary region* of *X* with respect to *R* is denoted by $B_R(x)$, that is defined by $B_R(x) = U_R(x) - L_R(x)$.

X is called *rough set* if $LR(X)\neq UR(X)$

1.2 Definition [2]

Let U be the universe and R be an equivalence relation on U and τ R(X) = {U, ϕ, $L_R(x)$, $U_R(x)$, $B_R(x)$}.

where X⊆U, then τ R(X) satisfies the following axioms:

1) U and $\phi\in\tau R(x)$
2) The union of elements of any sub collection of *τ R*(*X*) is in *τ R*(*X*).
3) The intersection of the elements of any finite sub collection of *τ R*(*X*) is in *τ R*(*X*).

That is *τ R*(*X*)) forms a topology on *U* called as the nano topology on U with respect to *X*.

We call (*U*,*τ R*(*X*)) as the nano-topological space.

The elements of *τ R*(*X*) are called as *Nano open set*

And the complement of a *Nano open sets* is called *nano closed sets*

1.3 Definition [3]

If (U,*τ R*(*X*)) is a *nano topological space* with respect to *X* and if *A* ⊆ *U* , then the
nano interior of *A* is defined as the union of all *nano open subsets* of *A* and it is denoted by
NInt(A) That is, *NInt(A)* is the largest *nano open* subset of *A*.

The *nano closure* of *A* is defined as the intersection of all nano closed sets containing *A* and
it is denoted by *NCl(A)*.

That is, *NCl(A)* is the smallest nano closed set containing *A*.

1.4 Definition [4]

Let (*U*, *τR*(*x*)) be a nano topological space the set *B* = {*U*, *LR*(*x*), *BR*(*x*)} is called a *bases* for the nano topology *τ R*(*x*) on *U* with respect to *X*.

1.5 Definition[5][6]

Let (*U*, *τR*(*x*)) be a nano topological space ,and *A* ⊆ *U* Then *A* is said to be [1][4]

1. Nano regular open if A = nint (ncl (A)),
2. Nano *α*-open if A⊆(nint (ncl(nint(A))),
3. Nano Semi-open if A ⊆ncl (nint (A))
4. Nano pre-open if A ⊆nint (ncl (A))
5. Nano *γ*-open (or nano b-open) if A ⊆ncl (nint(A) ∪nint (ncl(A)))
6. Nano *β*-open (or nano semi-pre-open) if A⊆ncl(nint (ncl (A)))

The family of all nano regular open (resp. nano *α*-open, nano semi-open, nano pre-open
, nano *γ*-open and nano *β*-open) sets in anano topological space (*U*, *τR*(*X*)), is denoted by NRO(U,X)(resp. N*α*O(U,X), NSO(U,X), NPO(U,X) , N*γ*O(U,X) and N*β*O(U,X))[7].

1.6 Example[9]

Let U={a,b,c,d} with U/R = {{a},{d},{b,c}} , and A = {a,d}.

Then we can deduce that *τR*(*A*) = { U ,*ϕ*,{a,d}} hear the set {a,b,d} is nano *α*-open but not nano open in(U,*τR*(*A*))

1.7 Example

Let U={a,b,c,d} with U/R = {{a},{c},{b,d}} , and A = {a,d}.

Then the nano topological space *τR*(*A*) = { U ,*ϕ*,{a},{b,d},{a,b,d}} then we have the following

1. If B = {a, b,d}, then B is nano open,but not nano regular open.
2. If C = {a,c} then C is nano-semiopen but not nano *α*-open[10].
3. If D = {a,b} then D is nano pre-open but not nano *α*-open[11].
4. If E = {a,b,c}, then E is nano *γ*-open but not nano semi-open.
5. If G = {b, c} then G is nano *β*-open but not nano *γ*-open.
6. If F = { b,c,d }, then F is nano *γ*-open but not nano pre-open.

1.8 proposition [1][5]

If (U, R) is an approximation space and X, Y ⊆ U, then

1. $L_R(X) \subseteq X \subseteq U_R(X)$
2. $L_R(0) = U_R(\theta)$ and $L_R(U) = U_R(U) = U$
3. $U_R(XUY) = U_R(X)$ $U_R(Y)$
4. $U_R(X\cap Y) \subseteq U_R(X) \cap_U R\ (Y)$
5. $L_R(XUY) = L_R(x) \cup L_R(Y)$
6. $L_R(X^{\cap}Y) = L_R(X) \cap L_R(Y)$
7. $L_R(X) \subseteq L_R(Y)$ and $U_R(X) \subseteq U_R(Y)$ whenever *X*
8. $U_R(X^c) = [LR(x)]c$ and $L_R(X^c) = [UR(x)]c$
9. $U_RU_R(X) = L_RUR(X) = UR(X)$
10. L *R*(*X*) = URL *R*(*X*) = *LR*(*X*)

Proof see [1][5]

1.9 Definition

Let (*U*, $\tau_R(X)$) be a *nano topological space* and *A* ⊆ *U*

The *nano δ-closure* of *A* is defined by ncl*δ* (*A*) = $\{x\in U : A\cap nint(ncl(G))\neq\phi,\ G\in\tau_R(X)$ and $x\in G\}$. A set *A* is called *nano δ-closed* if *A* = *nclδ*(*A*).

The complement of a *nano δ-closed* set is *nano δ-open*. Notice that nint *δ* (*A*) = *U* − *ncl δ* (*U* −

https://doi.org/10.25130/tjps.v28i3.1437

A), where $U - A = A'$ is the complement of A.

1.10 Definition

Let $(U, \tau_R(X))$ be a nano topological space and $A \subseteq U$. The set A is said to be[11]:

1. nano δ-regular open, if $A = nint(ncl\delta(A))$[7].
2. nano $\delta\alpha$-open, if $A \subseteq nint(nlc(nint\ \delta(A))$[12].
3. nano δ-semi open, if $A \subseteq ncl(nint\delta(A))$[7].
4. nano δ-pre-open, if $A \subseteq nint(ncl\delta(A))$[7].

2 nano semi

2.1 Definition

A subset S of a Nano topological space (X,T) is called [12]

1. $nano\text{-}semi - open$ if $S \subseteq ncl(nint(S))$**[10][8]**
2. $nano\text{-}\alpha - open$ if $S \subseteq nint(ncl(nint(S)))$ [**10**]
3. $nano\text{-}\beta - open(or\ nano\text{-}semi - pre - open$ [**13**]) if $S \subseteq ncl(nint(ncl(S)))$ [**15**]
4. $nano\text{-}b - open$ **[4][8]** (or $nano\text{-}\gamma - open$[**9**] or $nano\text{-}SP - open$[**14**]) if $S \subseteq nint(ncl(S)) \cup$ ncl(nint(S))
5. $nano\text{-}pre - open$ if $S \subseteq nint(ncl(S))$ [**6**]
6. $nano\text{-}\delta - pre - open$[**8**] if $S \subseteq nint(ncl\delta(S))$ [**12**]
7. $nano\text{-}\delta - semi - open$ if $S \subseteq ncl(nint\delta(S))$ [**16**]
8. $nano\text{-}e - open$ if $S \subseteq ncl(nint\delta(S)) \cup nint(ncl\delta(S))$ [**17**]
9. $nano\text{-}e^{*}\text{-}open$ (or $nano\text{-}\delta\text{-}\beta\text{-}open$) if $S \subseteq ncl(nint(ncl\delta(S)))$ [**6**]
10. $nano\text{-}\theta - semi - open$ if $S \subseteq ncl(nint\theta(S))$ [**18**]
11. $nano\text{-}\theta - pre - open$ if $S \subseteq nint(ncl\theta(S))$ [**19**]
12. $nano\text{-}\theta\beta - open$ if $S \subseteq ncl(nint(ncl\theta(S)))$ [**19**]

2.2 Example

Let U ={a, b, c, d} with $U/$ R={{a},{b},{c, d}} and X $= \{a, c\} = U$.

Then the Nano topology $\tau_R(X) = \{U, \phi, \{a\}, \{c, d\}, \{a, c, d\}\}$.

3 The algorithm needed to compute Nano topology

3.1 The function RE3

The aim of this function to find the Relations Equivalence OF set X.

3.1.1 The Program

```
from  SPX import *
def R1(list1_int,list2_int):
        list3 = []
        for i in range(len(list1_int)):
          for j in range(len(list2_int)):
                list3.append([list1_int[i],list2_int[j]])
        return(list3)
# test R1
#x = [1,2];y=[10,20]
#r1 = R1(x,y);print('R1_____')
#print(r1)
def R2(list_list,list_int):
        list3=[]
        for i in range(len(list_list)):
          li = list_list[i]
          for j in range(len(list_int)):
                l3 = [list_int[j]]
                for k in range(len(li)):
                l3.append(li[k])
                list3.append(l3)
        return(list3)
# Test R2
#z = [100,200]
#r2 = R2(r1,z);print('R2_______')
#print(r2)
def M_l(x):
        n = len(x) ; nn = 2**n ;ml=[]
        for i in range(n):
          start = 2**(n-1-i)
          end   = (2**(n-i))
          l=[k for k in range(start,end)]
          if i > 0 :
                l.append(0)
          ml.append(l)
        return(ml)
#Test M_L (multi list)
#ml = M_l([1,2,3,4]);print('M_l___')
#print(ml)
def R(X):
        n = len(X); ml = M_l(X)
        if n == 1 :
          r2 = [1]
        elif n == 2 :
          r2 =[[1],[2,3]]
        else :
          r1 = R1(ml[0],ml[1]) ; r2 = r1
          for i in range(2,len(ml)):
                r2 = R2(r2,ml[i])
        r3 = []
        for i in range(len(r2)):
          if (sum(r2[i])) == (2**n)-1 :
                s = (2**n)-1
                for j in range(n):
                if r2[i][j] != 0 :
                   s = s & r2[i][j]
                if s == 0:
                r2[i].sort() ; r3.append(r2[i])
        r3end = [0 for i in range(n-1)]
        r3end.append((2**n)-1)
        r3.append(r3end)
        return(r3)
#R3 = R([1,2,3,4]);print('R____')
#print(R3);print(len(R3))
def f1(r3):
        l =len(r3);r4=[]
        for k in range(l):
          r3k = r3[k]
          while 0 in r3k :
                r3k.remove(0)
          r4.append(r3k)
        return(r4)
#F1 = f1(R3);print('f1___')
#print(F1);print(len(F1))
def ff(r3) :
        r4 = []
        for k in range(len(r3)):
          r3k = r3[k] ; le=[]
          for i in range(0,len(r3k)-1):
                li=r3k[i]
```

https://doi.org/10.25130/tjps.v28i3.1437

```
                for j in range(i+1,len(r3k)):
                lj = r3k[j]
                e = li & lj
                le.append(e)
          if sum(le) == 0 :
                r4.append(r3k)
        return(r4)
#FF = ff(R3);print('ff___')
#print(FF);print(len(FF))
def pr(list1):
        for i in range(len(list1)):
          print(i,list1[i])
x=[1,2,3,4] ; n = len(x);#print('R_____')
RX = R(x); #print(len(RX))
#pr(RX)
RX4 = f1(RX);#print('RX________')
#pr(RX4)
ff4 = ff(RX4);#print('ff4______')
#pr(ff4)
def All_Eq_R(X):
        lpx = Lpx(X); All_R = []
        RX = R(X) ; RX4 = f1(RX) ; ff4 = ff(RX4)
        for i in range(len(ff4)):
          li = ff4[i] ; s = []
          for j in range(len(li)):
                lj = li[j] ; s0 = set(lpx[lj]) ;
s.append(s0)
          All_R.append(s)
        return(All_R)
#print('All_R_____')
#print("All Eqwevalens relation was")
#allR = All_Eq_R(x)
#pr(all)
```

3.1.2 Example Test the function RE3

If we test the function RE3 on set have three element we have

0 [{3}, {2}, {1}]
1 [{2, 3}, {1}]
2 [{2}, {1, 3}]
3 [{3}, {1, 2}]
4 [{1, 2, 3}]

And if we test the function RE3 on set have four element we have

0 [{4}, {3}, {2}, {1}]
1 [{3, 4}, {2}, {1}]
2 [{3}, {2, 4}, {1}]
3 [{4}, {2, 3}, {1}]
4 [{2, 3, 4}, {1}]
5 [{3}, {2}, {1, 4}]
6 [{2, 3}, {1, 4}]
7 [{4}, {2}, {1, 3}]
8 [{2, 4}, {1, 3}]
9 [{2}, {1, 3, 4}]
10 [{4}, {3}, {1, 2}]
11 [{3, 4}, {1, 2}]
12 [{3}, {1, 2, 4}]
13 [{4}, {1, 2, 3}]
14 [{1, 2, 3, 4}]

3.2 The function Nano

The aim from this function to fined the Nano topology ON

3.2.1 The Program

```
from RE3 import *
U = {1,2,3,4,5,6,7}
X = {1,2,3,4}
A = {1,2,3}
S
=[{1},{2},{3},{4},{1,2},{1,3},{1,4},{2,3},{2,4},{3,4},{1
,2,3},{1,2,4},{1,3,4},{2,3,4}]
R = [{1,2,3},{4,5},{6,7}]
print('X=',X)
print('A=',A)
print('U=',U)
print('U/R =',R)
def LR(X,U,R):
   Lr = set()
   for x in U:
      for Rx in R:
         if (x in Rx)and(Rx.issubset(X)):
            Lr.update(Rx)
   return(Lr)
print('Lower approximation =',LR(X,U,R))
def UR(X,U,R):
   Ur = set()
   for x in U:
      for Rx in R:
         if Rx.intersection(X)!= set():
            Ur.update(Rx)
   return(Ur)
print('Upper approximation =',UR(X,U,R))
def BR(X,U,R):
   Lr = LR(X,U,R)
   Ur = UR(X,U,R)
   Br = Ur.difference(Lr)
   return(Br)
print('The boundary Region =',BR(X,U,R))
def NTS(X,U,R):
   Ur = UR(X,U,R)
   Lr = LR(X,U,R)
   Br = BR(X,U,R)
   nts =[set(),U,Ur,Lr,Br]
   return(nts)
print('Nano topology =',NTS(X,U,R))
def Nbasis(X,U,R):
   Lr = LR(X,U,R)
   Ur = UR(X,U,R)
   Nb=[U,Lr,Ur]
   return(Nb)
print('Nano basis =',Nbasis(X,U,R))
def NOinA(A,NTS):
   noinA=[]
   for nts in NTS :
      if nts.issubset(A):
         noinA.append(nts)
   return(noinA)
Nts = NTS(X,U,R)
print('Nano Open subset A',NOinA(A,Nts))
def NInt(A,U,NTS):
   nint = set()
```

https://doi.org/10.25130/tjps.v28i3.1437

```
    noinA = NOinA(A,NTS)
    for no in noinA :
        nint.update(no)
    return(nint)
print('Nano interior of A =',NInt(A,U,Nts))
def NCCA(A,U,NTS):#nano closed contined set A
    ncca = []
    for onts in NTS :
        cnts = U.difference(onts)
        if A.issubset(cnts):
            ncca.append(cnts)
    return(ncca)
print('Nano closed set containing A=',NCCA(A,U,Nts))
def NCl(A,U,NTS):
    ncl = U
    ncca = NCCA(A,U,NTS)
    for nc in ncca :
        ncl = ncl.intersection(nc)
    return(ncl)
print('Nano closed set = ',NCl(A,U,Nts))
def NclNint(A,U,NTS):
    nint = NInt(A,U,NTS)
    nclnint = NCl(nint,U,NTS)
    return(nclnint)
print('Ncl(Nint(A,U,NTS))=',NclNint(A,U,Nts))
def NintNcl(A,U,NTS):
    ncl = NCl(A,U,NTS)
    nintncl = NInt(ncl,U,NTS)
    return(nintncl)
print('Nint(Ncl(A))=',NintNcl(A,U,Nts))
def NintNclNint(A,U,NTS):
    nclnint = NclNint(A,U,NTS)
    nintclint = NInt(nclnint,U,NTS)
    return(nintclint)
print('Nint(Ncl(Nint(A)))=',NintNclNint(A,U,Nts))
def NclNintNcl(A,U,NTS):
    nintncl = NintNcl(A,U,NTS)
    nclincl = NCl(nintncl,U,NTS)
    return(nclincl)
print('Ncl(Nint(Ncl(A)))=',NclNintNcl(A,U,Nts))
def IsNRegulaO(A,U,NTS):
    S = NintNcl(A,U,NTS)
    if A == S :
        return(True)
    else:
        return(False)
def IsNAlphaO(A,U,NTS):
    S = NintNclNint(A,U,NTS)
    if A.issubset(S) :
        return(True)
    else:
        return(False)
def IsNSemiO(A,U,NTS):
    S = NclNint(A,U,NTS)
    if A.issubset(S) :
        return(True)
    else:
        return(False)
def IsNPreO(A,U,NTS):
    S = NintNcl(A,U,NTS)
    if A.issubset(S) :
        return(True)
    else:
        return(False)
def IsNbeta(A,U,NTS):
    S = NclNintNcl(A,U,NTS)
    if A.issubset(S) :
return(True)
    else:
return(False)
def IsNgamma(A,U,NTS):
    S1 = NclNint(A,U,NTS)
    S2 = NintNcl(A,U,NTS)
    S1.update(S2)
    if A.issubset(S1) :
return(True)
    else:
return(False)
print('test regular')
for i in range(len(S)):
print(S[i],IsNRegulaO(S[i],U,Nts))
print('test Nano Alpha Open')
for i in range(len(S)):
print(S[i],IsNAlphaO(S[i],U,Nts))
print('test Nano Semi Open')
for i in range(len(S)):
print(S[i],IsNSemiO(S[i],U,Nts))
print('test Nano Pre Open')
for i in range(len(S)):
print(S[i],IsNPreO(S[i],U,Nts))
print('test Nano beta open')
for i in range(len(S)):
print(S[i],IsNbeta(S[i],U,Nts))
print('test Nano gamma open')
for i in range(len(S)):
print(S[i],IsNgamma(S[i],U,Nts))
#print(UR(X,U,R))
#print(BR(X,U,R))
```

3.2.2 Example Test the function Nano

X= {1, 3}
A= {1, 2, 3}
U= {1, 2, 3, 4}
U/R = [{1}, {2}, {3, 4}]
Lower approximation = {1} Upper approximation = {1, 3, 4}
The boundary Region = {3, 4}
Nano topology = [set(), {1, 2, 3, 4}, {1, 3, 4}, {1}, {3, 4}]
Nano basis = [{1, 2, 3, 4}, {1}, {1, 3, 4}]
Nano Open subset A [set(), {1}] Nano interior of A = {1}
Nano closed set containing A= [{1, 2, 3, 4}]
Nano closed set = {1, 2, 3, 4}
Ncl(Nint(A,U,NTS))= {1, 2}
Nint(Ncl(A))= {1, 2, 3, 4}
Nint(Ncl(Nint(A)))= {1}
Ncl(Nint(Ncl(A)))= {1, 2, 3, 4}
test Nano regular
{1} True

https://doi.org/10.25130/tjps.v28i3.1437

{2} False
{3} False
{1, 2} False
{1, 3} False
{2, 3} False
{1, 2, 3} False
test Nano Alpha Open
{1} True
{2} False
{3} False
{1, 2} False
{1, 3} False
{2, 3} False
{1, 2, 3} False
test Nano Semi Open
{1} True
{2} False
{3} False
{1, 2} True
{1, 3} False
{2, 3} False
{1, 2, 3} False
test Nano Pre Open
{1} True
{2} False
{3} True
{1, 2} False
{1, 3} True
{2, 3} False
{1, 2, 3} True
test Nano beta open
{1} True
{2} False
{3} True
{1, 2} True
{1, 3} True
{2, 3} True
{1, 2, 3} True
test Nano gamma open
{1} True
{2} False
{3} True
{1, 2} True
{1, 3} True
{2, 3} False
{1, 2, 3} True
That is meaning
All Nano regular -Open sets ={{1}}
All Nano alpha-Open sets = {{1}}
All Nano semi-Open sets = {{1},{1, 2}}
All Nano Pre-Open sets = {{1},{3},{1, 3},{1, 2, 3} }
All Nano beta-Open sets = {{1},{3},{1, 2},{1, 3},{2, 3},{1, 2, 3} }
All Nano Gamma-Open sets = {{1},{3},{1, 2},{1, 3},{1, 2, 3} }

Reference

[1] Bhuvaneswari, K., & Gnanapriya, K. M. (2014). Nano generalized closed sets in nano topological spaces. *International Journal of Scientific and Research Publications*, *4*(5), 1-3.
[2] Parimala, M., Jafari, S., & Murali, S. (2017). Nano ideal generalized closed sets in nano ideal topological spaces. In *Annales Univ. Sci. Budapest* (Vol. 60, pp. 3-11).
[3] Bhuvaneswari, K., & Ezhilarasi, A. (2014). On nano semi-generalized and nano generalized-semi closed sets in nano topological spaces. *International Journal of Mathematics and Computer Applications Research*, *4*(3), 117-124.
[4] Parimala, M., Indirani, C., & Jafari, S. (2016). On nano b-open sets in nano topological spaces. *Jordan Journal of Mathematics and Statistics*, *9*(3), 173-184.
[5] Revathy, A., & Ilango, G. (2015). On nano β-open sets. *Int. J. Eng. Contemp. Math. Sci*, *1*(2), 1-6
[6]Hosny, M. (2020). Nano∧ β-sets and nano∧ β-continuity. *Journal of the Egyptian Mathematical Society*, *28*(1), 1-11.
[7] Pankajam, V., & Kavitha, K. (2017). δopen sets and δ nano continuity in δ nano topological space. *International Journal of Innovative Science and Research Technology*, *2*(12), 110-118.
[8] Parimala, M., Indirani, C., & Jafari, S. (2016). On nano b-open sets in nano topological spaces. *Jordan Journal of Mathematics and Statistics*, *9*(3), 173-184.
[9] Nasef, A. A., Aggour, A. I., & Darwesh, S. M. (2016). On some classes of nearly open sets in nano topological spaces. *Journal of the Egyptian Mathematical Society*, *24*(4), 585-589.
[10] Thivagar, M. L., & Richard, C. (2013). On nano forms of weakly open sets. *International journal of mathematics and statistics invention*, *1*(1), 31-37.
[11] Benchalli, S. S., Patil, P. G., Kabbur, N. S., & Pradeepkumar, J. (2017). Weaker forms of soft nano open sets. *Computer Math. Sci*, *8*(11), 589-599.
[12] Parimala, S., & Chandrasekar, V. (2019). Nano δ open sets and their notions. *Malaya Journal of Matematik*, *1*, 664-672.
[13] Sathishmohan, P., Rajendran, V., Kumar, C. V., & Dhanasekaran, P. K. (2018). On nano semi pre neighbourhoods in nano topological spaces. *Malaya Journal of Matematik*, *6*(1), 294-298.
[14] Pirbala, O. T., & Ahmedb, N. K. On Nano Sβ-Open Sets In Nano Topological Spaces.
[15] Revathy, A., & Ilango, G. (2015). On nano β-open sets. *Int. J. Eng. Contemp. Math. Sci*, *1*(2), 1-6.
[16] Parimala, S., Sathiyaraj, J., & Chandrasekar, V. New notions via nano δ-open sets with an application in diagnosis of type-II diabetics.
[17] Manivannan, P., Vadivel, A., Saravanakumar, G., & Chandrasekar, V. (2020). Nano generalized e-closure and nano generalized e-interior. *Malaya Journal of Matematik (MJM)*, *8*(1, 2020), 89-98.
[18] Padma, A., Saraswathi, M., Vadivel, A., & Saravanakumar, G. (2019). New notions of nano M-open sets. *Malaya Journal of Matematik, S (1)*, 656-660 .

https://doi.org/10.25130/tjps.v28i3.1437

[19] Sujatha, M., Vadivel, A., Rangarajan, R. V. M., & Angayarkanni, M. (2019, December). Nano continuous mappings via nano θ open sets. In *AIP Conference Proceedings* (Vol. 2177, No. 1, p. 020100). AIP Publishing LLC.