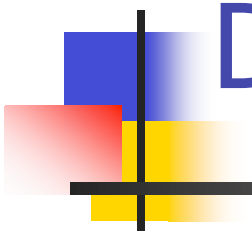
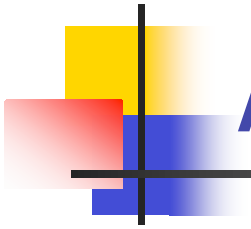


Non-Signature based Virus Detection

InSeon Yoo

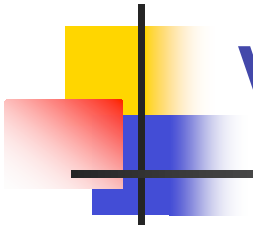
University of Fribourg

TCV Workshop, 4-5 May 2006



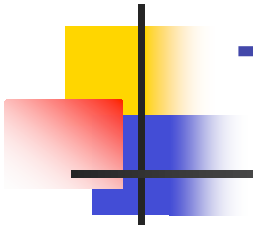
Agenda

- Virus detection methods
- Types of Viruses
- Parasitic Virus Structure
- SOM-based virus recognition method
- SOM-based Virus detection method
- Results of VirusDetector
- Conclusions



Virus Detection Methods

- Signature-based anti-virus (only for known virus-signatures)
  - Look for known patterns in malicious code
- Checksum (only provided that the record is up-to-date and not damaged)
  - Maintain record of "good" version of file
  - Check to see if changed
- Proof-carrying code *(if code modified, proof will fail)*
  - Code includes proof of correctness
  - At execution, verify proof against code
- Statistical Methods *(only works after the damage is done)*
  - High/low number of files read/written
  - Unusual amount of data transferred
  - Abnormal usage of CPU time

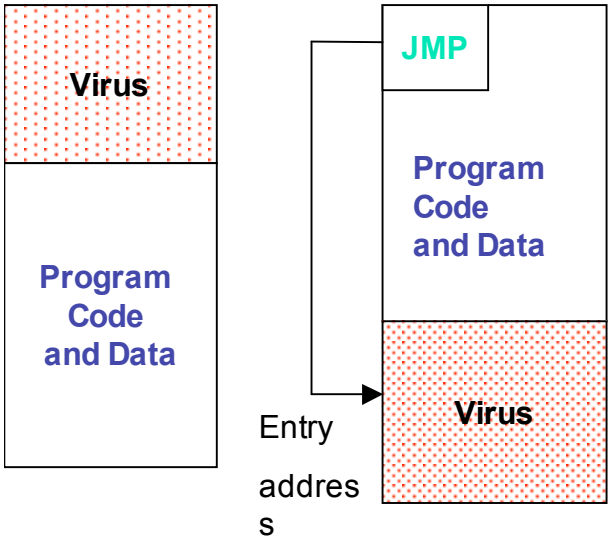


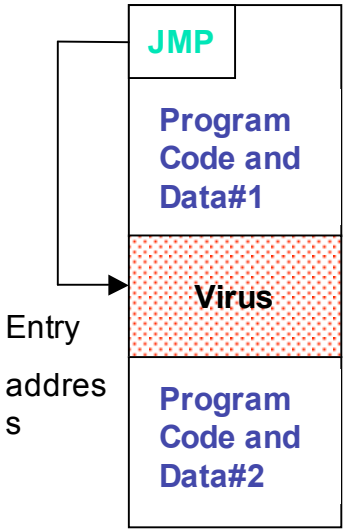
Types of Viruses

- Boot sector infectors
- Multipartite viruses:infect either boot sectors or executables
- TSR viruses: stay active in memory after the application (or bootstrapping, or disk mounting) is completed
- Stealth viruses
  - Conceal Infection
    - Trap read and disinfect (Let execute call infected file)
  - **Encrypted viruses**
    - A virus that is enciphered except for a small deciphering routine
  - **Polymorphic viruses**
- **Parasitic viruses** ← Executable infectors
- Macro viruses

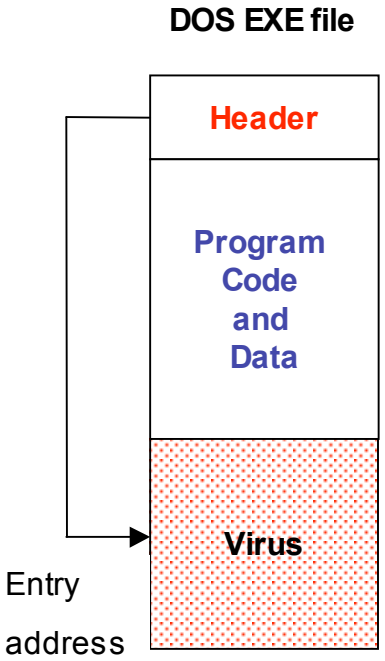
Parasitic Viruses

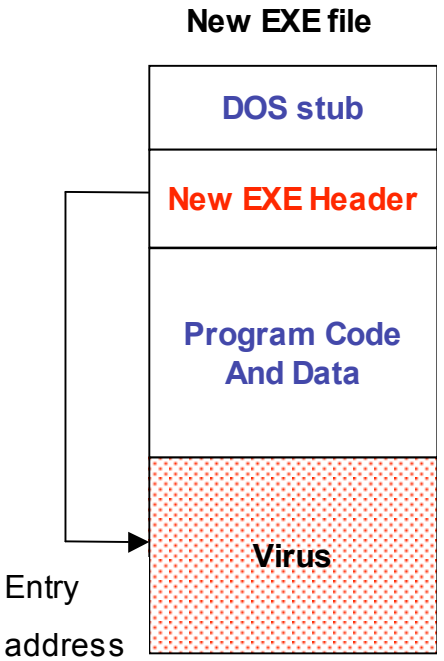
- COM format





- EXE format





Win CIH/Chernobyl virus

DOS Stub (DS)

```
0000000  M  Z 220 \0 003 \0 \0 \0 004 \0 \0 \0  ÿ  ÿ \0 \0
0000020  ¸ \0 \0 \0 \0 \0 \0 \0  @ \0 \0 \0 \0 \0 \0 \0
0000040 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
0000060 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 200 \0 \0 \0
0000100 016 037  º 016 \0  ´ \t  Í  !  ¸ 001  L  Í  !  T  h
0000120  i  s     p  r  o  g  r  a  m     c  a  n  n  o
0000140  t     b  e     r  u  n     i  n     D  O  S
0000160  m  o  d  e  .  \r \r \n  $ \0 \0 \0 \0 \0 \0 203
```

NewEXE Header (PE)

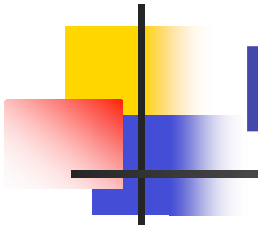
```
0000200  P  E \0 \0  L 001 004 \0  7  é  «  /  \0 \0 \0 \0
0000220 \0 \0 \0  à \0 006 003 \v 001 002  <  \0  ,  \0 \0
0000240 \0 032 \0 \0 \0 \0 \0 \0  @ 002 \0 \0 \0 020 \0 \0
0000260 \0  @ \0 \0 \0 \0 002 001 \0 020 \0 \0 \0 002 \0 \0
0000300 003 \0  3 \0 003 \0  3 \0 003 \0  3 \0 \0 \0 \0 \0
0000320 \0  p \0 \0 \0 004 \0 \0  G  I 001 \0 003 \0 \0 \0
0000340 \0 \0 020 \0 \0 020 \0 \0 \0 \0 020 \0 \0 020 \0 \0
0000360 \0 \0 \0 \0 020 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
0000400  Z  9 \0 \0  d \0 \0 \0 \0  P \0 \0  ¬ \v \0 \0
0000420 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
0000440 \0  ` \0 \0    002 \0 \0    020 \0 \0  8 \0 \0 \0
0000460 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
.....
```

Program Code & Data (PR)

```
0001160 020  f 211  s 002  ^  Ì  V 213  ð 213  H  ü  ó  ¤ 203
0001200  è \b 213  0 \v  ö  t 002  ë  ð  ^  Ì  û  3  Û  ë
0001220 \a  3  Û  d 213 003 213     d 217 003  X  ]  h  P  7
0001240 002 001  Ã  t  2 017  !  Á  ã 020 203 004  $ 025  f 211
0001260  k  ü  Á  í 020  f 211  k 002  Ï 017  #  Ã  j 017  Q
0001300  j  ÿ  Q  Q  Q  j 001  j 002  Í     S \0 001 \0 203
0001320  Ä   227 215  F 235  Ï 215 207  ÷  ü  ÿ  ÿ  P  Í
0001340  g \0  @ \0 017  #  À  X 213  N  = 213 021 211  P  ü
0001360 215  @  Ö 211 001  ú  ë  ¶  S  è \0 \0 \0 \0  [ 203
0001400  Ã  $  S  Í     h \0  @ \0  X  ÿ  t  $ \b  ÿ  S
0001420  ü  Y  P  S  ÿ  S  ü  Y 017  #  À  X  [  Ã  À 222
.....
```

Virus Code (VS)

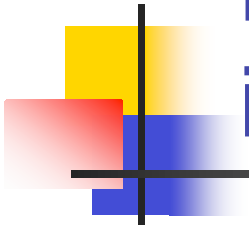
```
0044740  A \0  @ \0  2 \0  @ \0  C  I  H     v  1  .  2
0044760     T  T  I  T \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
0045000 001 \0 \0 \0 020 001 \0 \0 \0  C  P  @  e  x  e  \
0045020  a  r  p  .  D  B  G \0  .  e  x  e \0 \0 \0 \0
0045040 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
*
0045420  "     \0 \0     \0 \0 \0 \0 \0 \0 \0 \0 \0 \0 \0
0045440  °     \0 \0  , \0 \0 \0 \0 \0 \0 \0 002 \0 \a 003
0045460  Ü     \0 \0  2 \0 \0 \0 001 \0 \0 \0 001 \0 004 001
0045500 016  ! \0 \0  0 \0 \0 \0 \0 \0 \0 \0 002 \0 \0 \0
0045520  á  ! \0 \0  x \0 \0 \0 003 \0 \0 \0 \0 \0 006 003
0045540  Y  " \0 \0  Æ 001 \0 \0 002 \0 \0 \0 002 \0 026 024
0045560 037  $ \0 \0  = \0 \0 \0 \0 \0 \0 \0 001 \0 \a 003
0045600  \  $ \0 \0  G \0 \0 \0 \0 \0 \0 \0 001 \0 \0 \0
0045620  4  2 \0 \0  # 001 \0 \0 \0 \0 \0 \0 003 \0 \f 024
0045640  W  3 \0 \0 030 001 \0 \0 \0 \0 \0 \0 002 \0 016 003
.....
```



Polymorphic viruses

- A virus that changes its form each time it inserts itself into another program
- Idea is to prevent signature detection by changing the "signature" or instructions used for deciphering routine
- Toolkits to make these exist (Mutation Engine, Trident Polymorphic Engine)

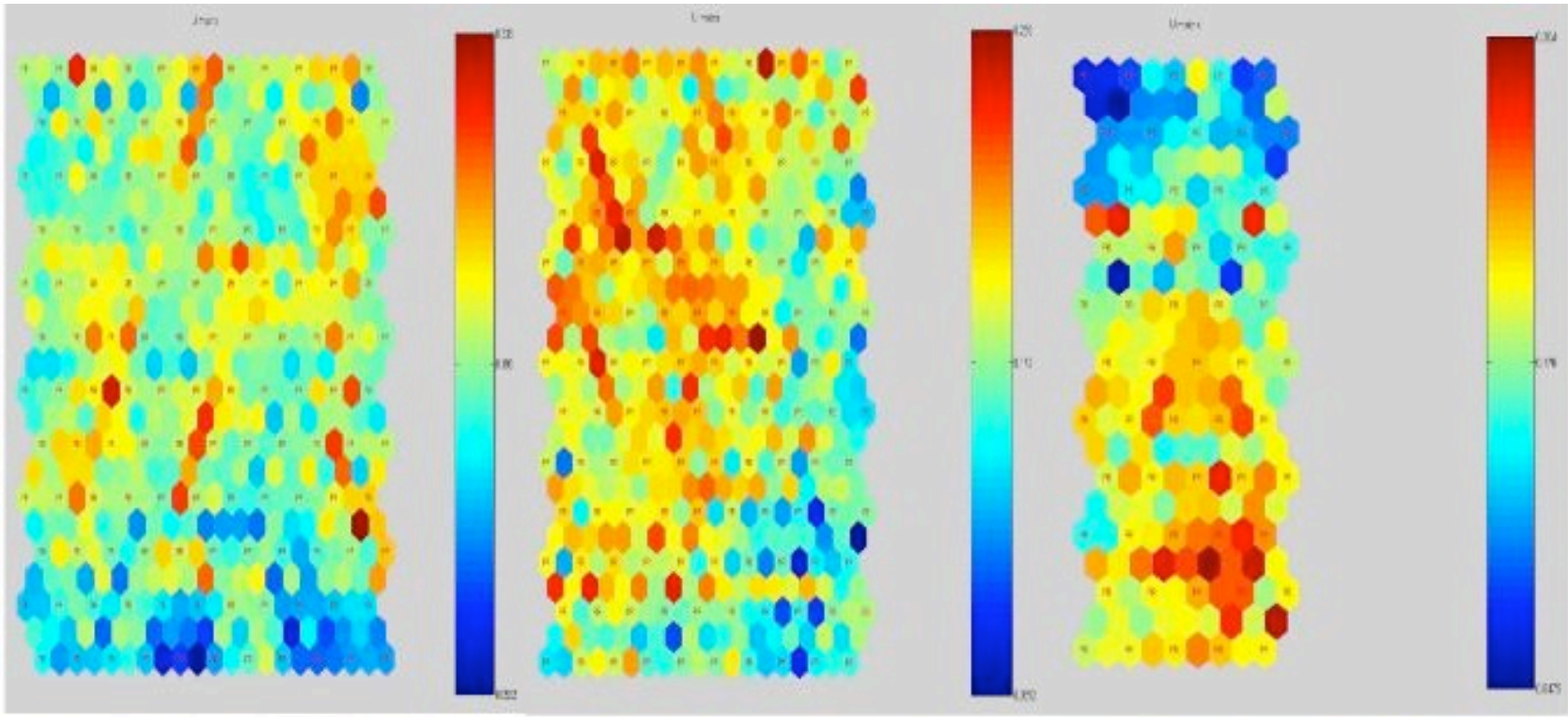
How to detect virus patterns in an infected files?



- Self-Organizing Maps
  - Recognize patterns
    - Patterns represent maliciousness
  - Deal with email attachment files

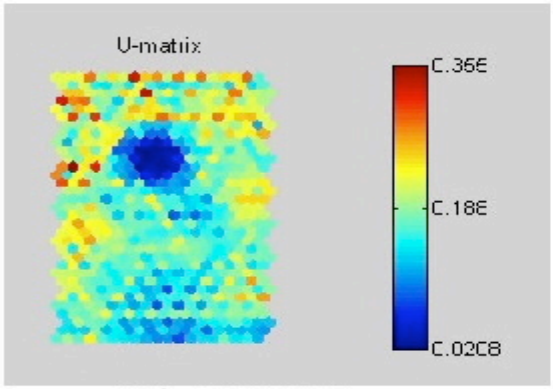
- Before introducing my approach to detect viruses, existing methods are presented first.

SOM of Win files before CIH virus infection

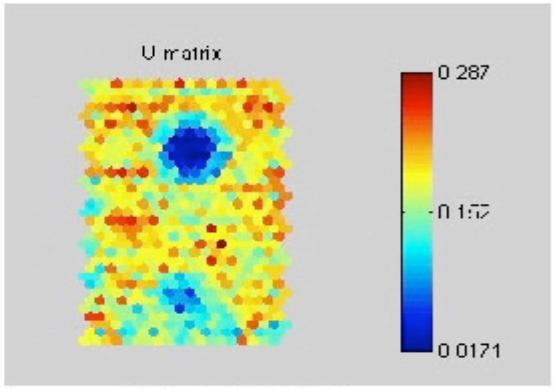


(a) before CIH1.2 (b) before CIH1.3 (c) before CIH1.4

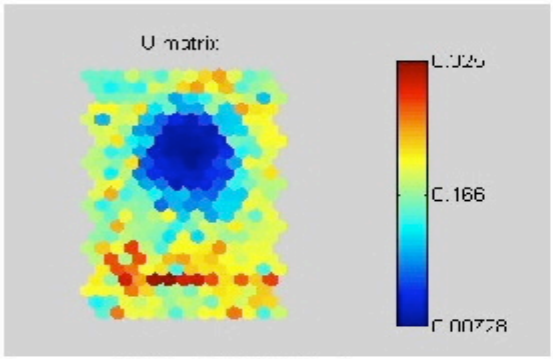
SOM of Win95.CIH virus



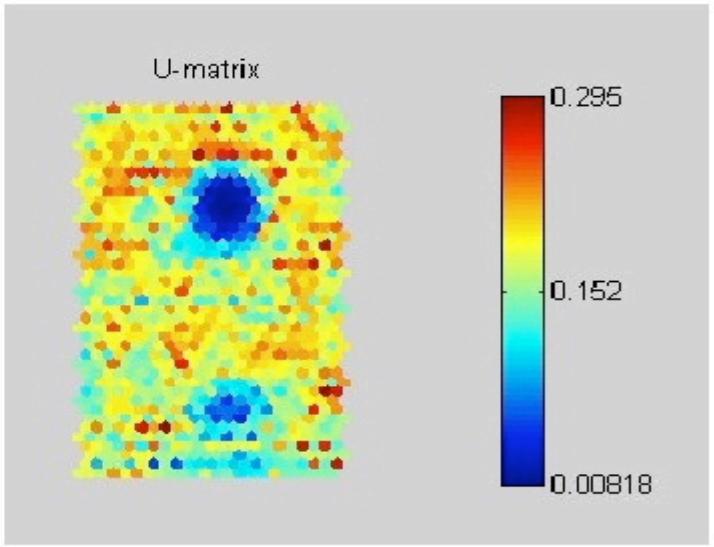
Win95CIH12



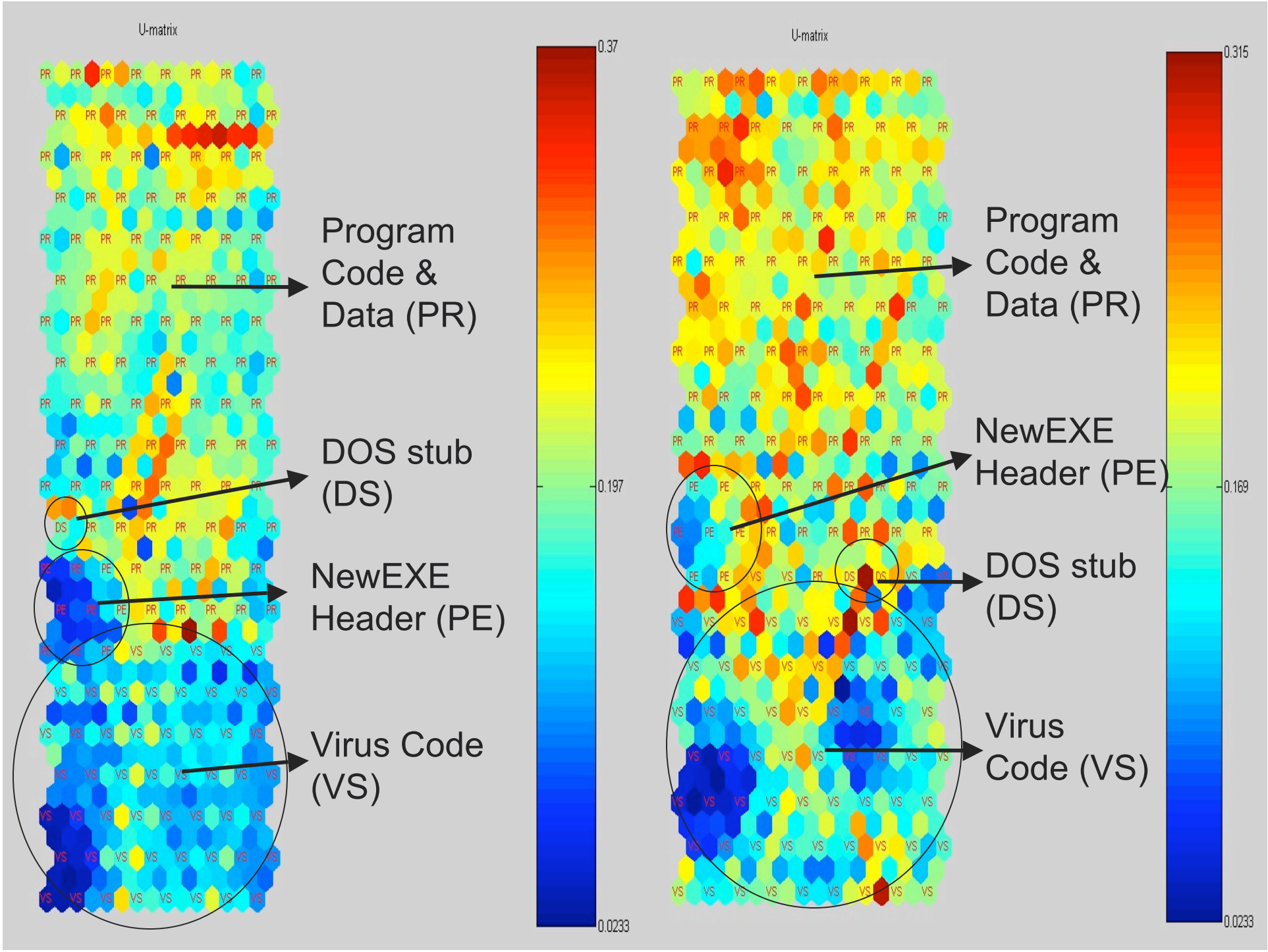
Win95CIH13



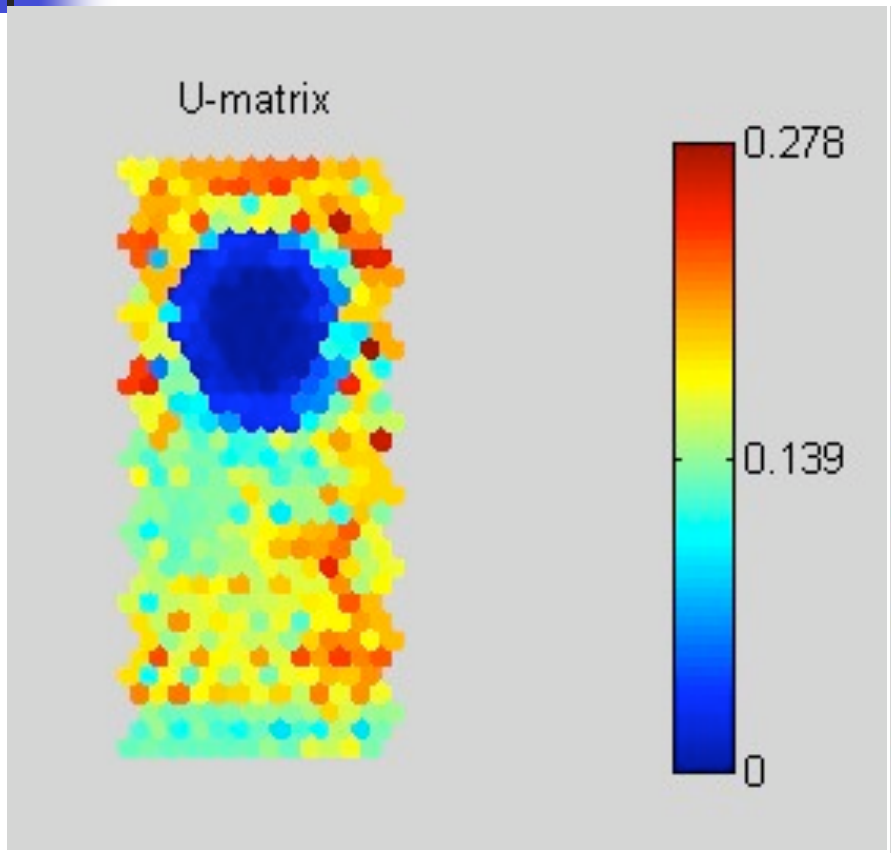
Win95CIH14



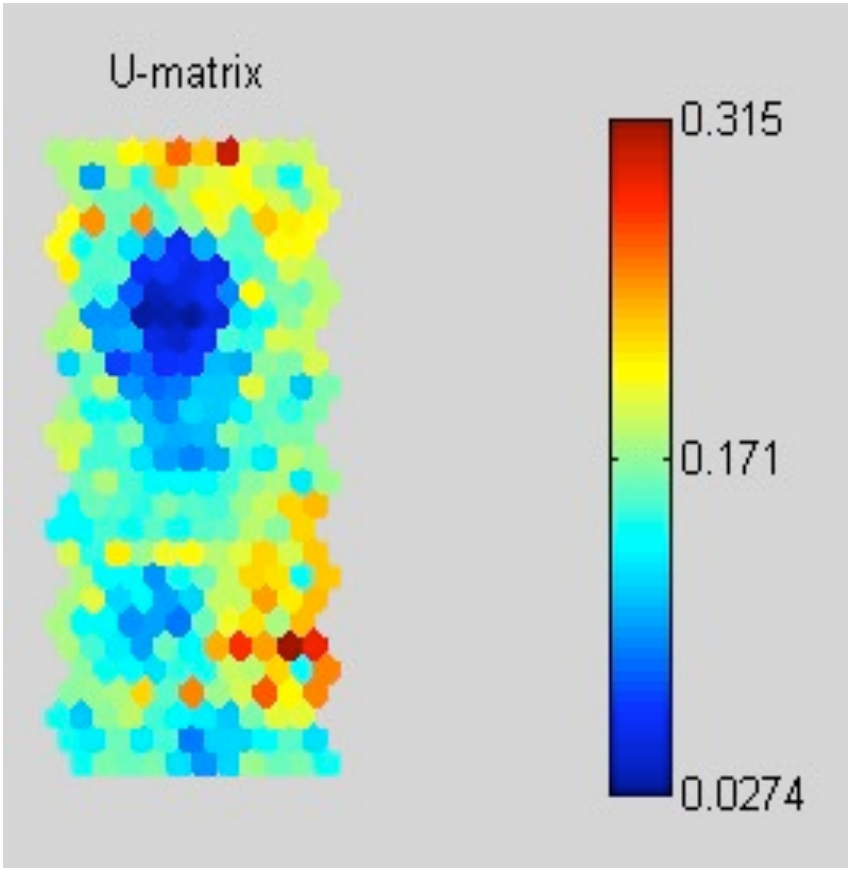
Win95 CIH Virus (by all data set)


U-matrix
0.37
0.197
0.0233
Program Code & Data (PR)
DOS stub (DS)
NewEXE Header (PE)
Virus Code (VS)
U-matrix
0.315
0.169
0.0233
Program Code & Data (PR)
NewEXE Header (PE)
DOS stub (DS)
Virus Code (VS)

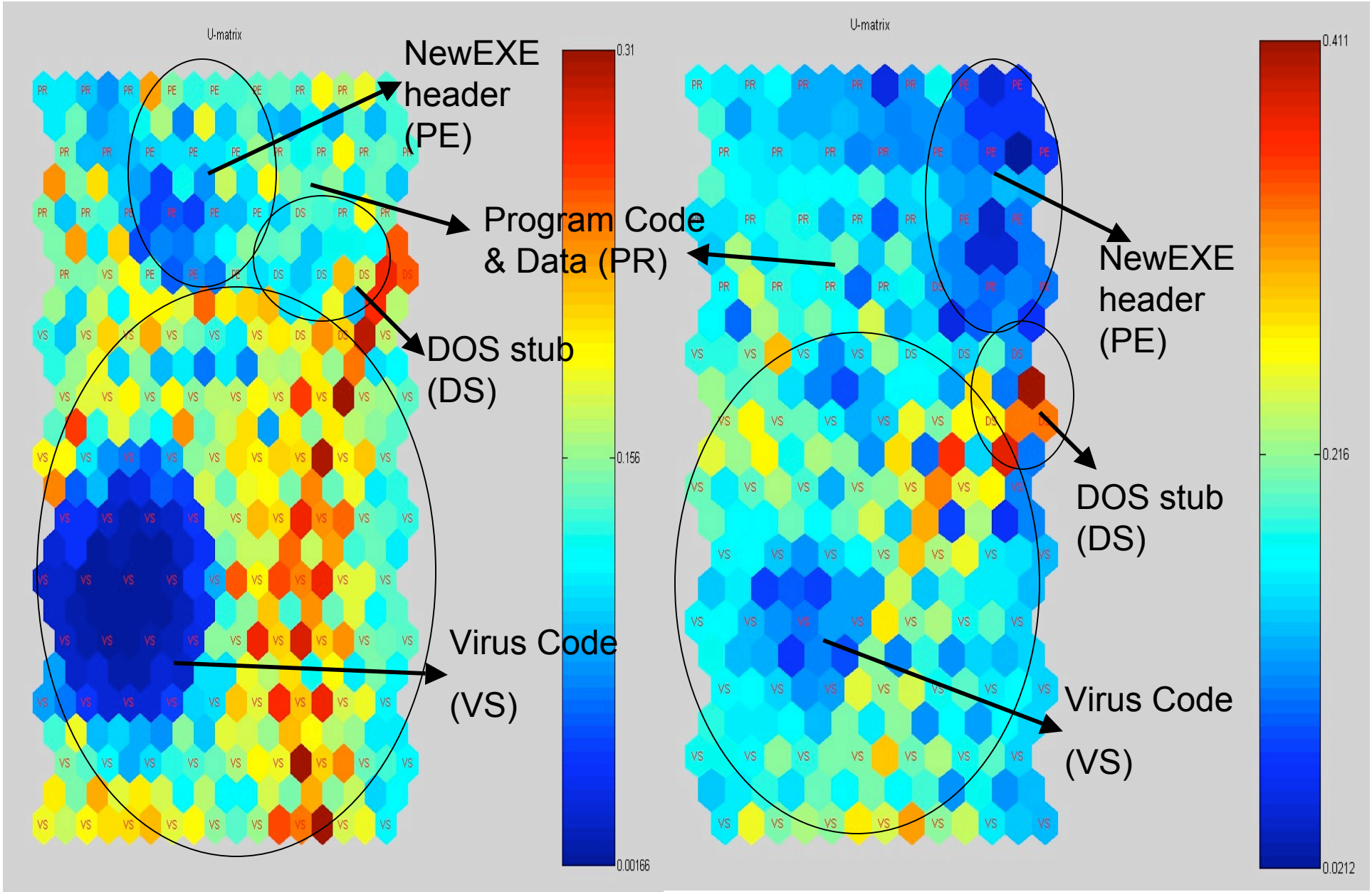
SOM of Boza virus



(a) SOM projection of Boza.A

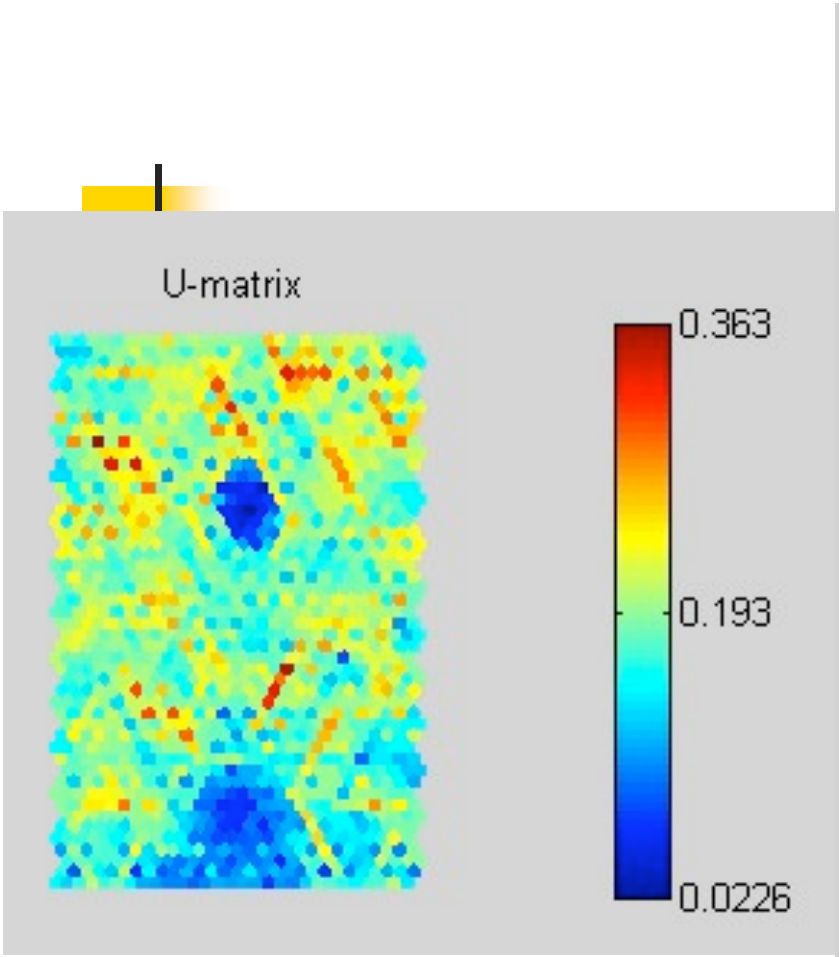


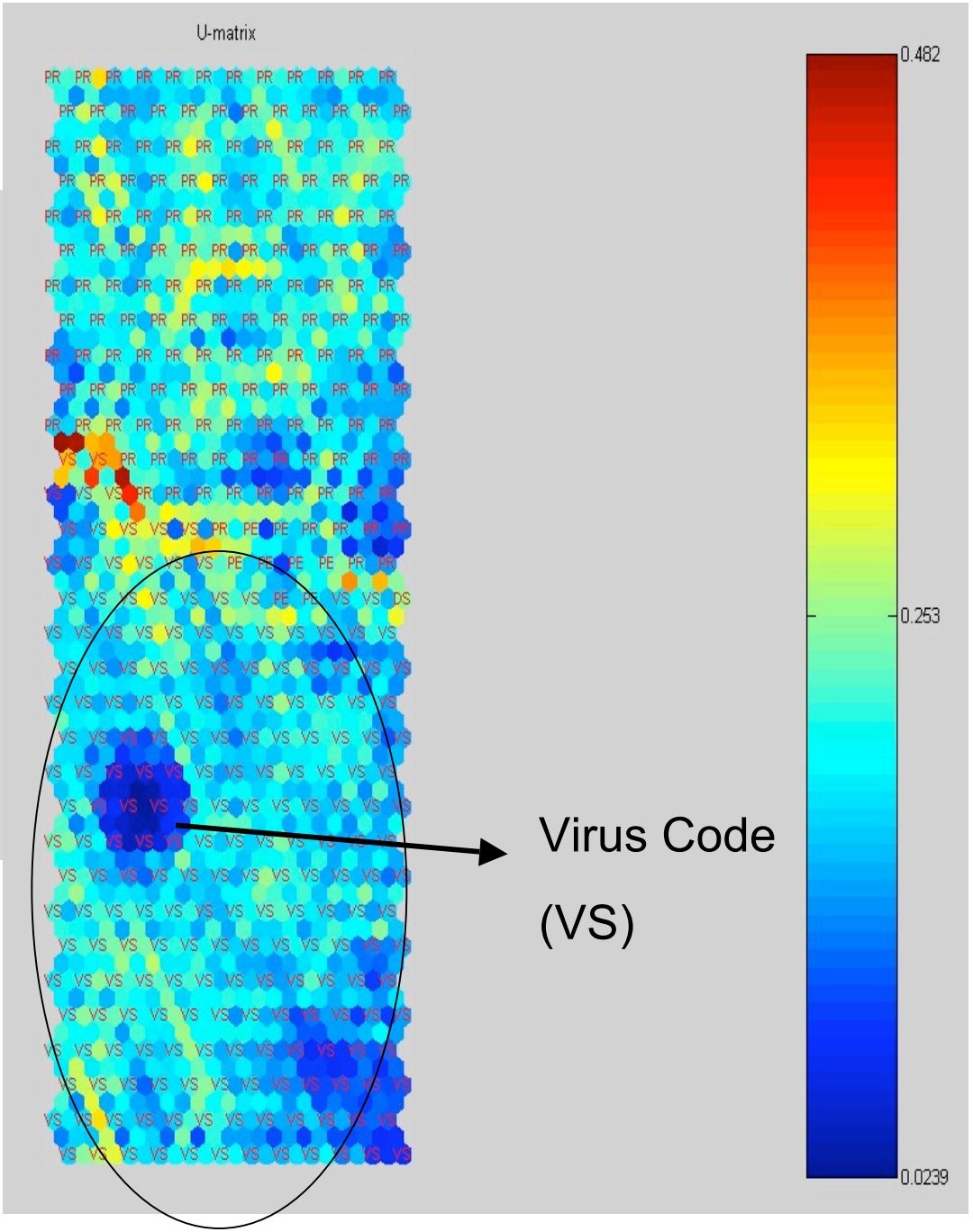
(b) SOM projection of Boza.C



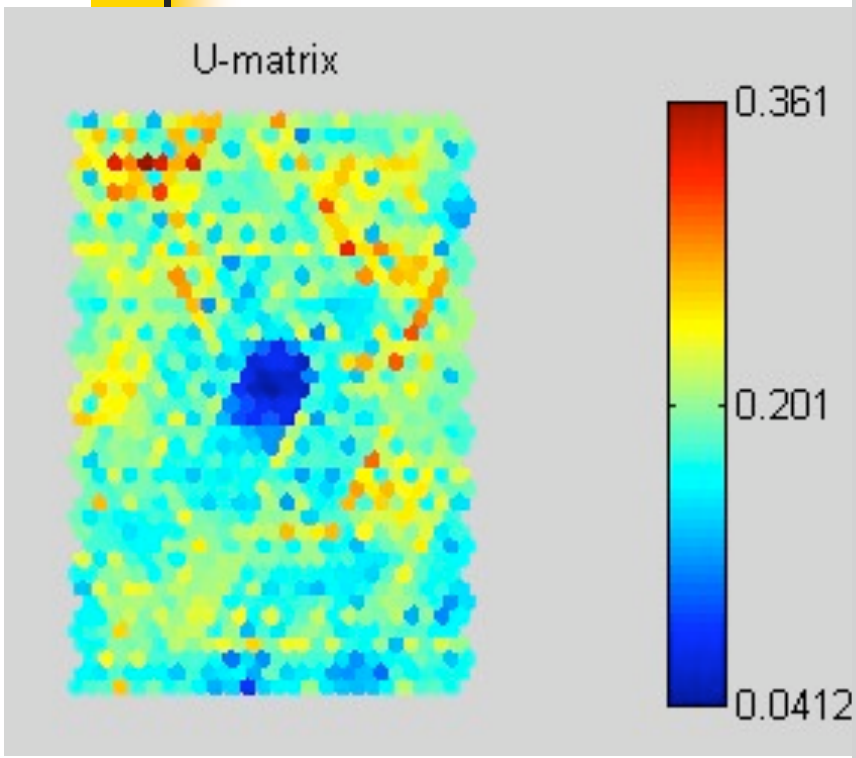
(a) Boza.A

(b) Boza.C

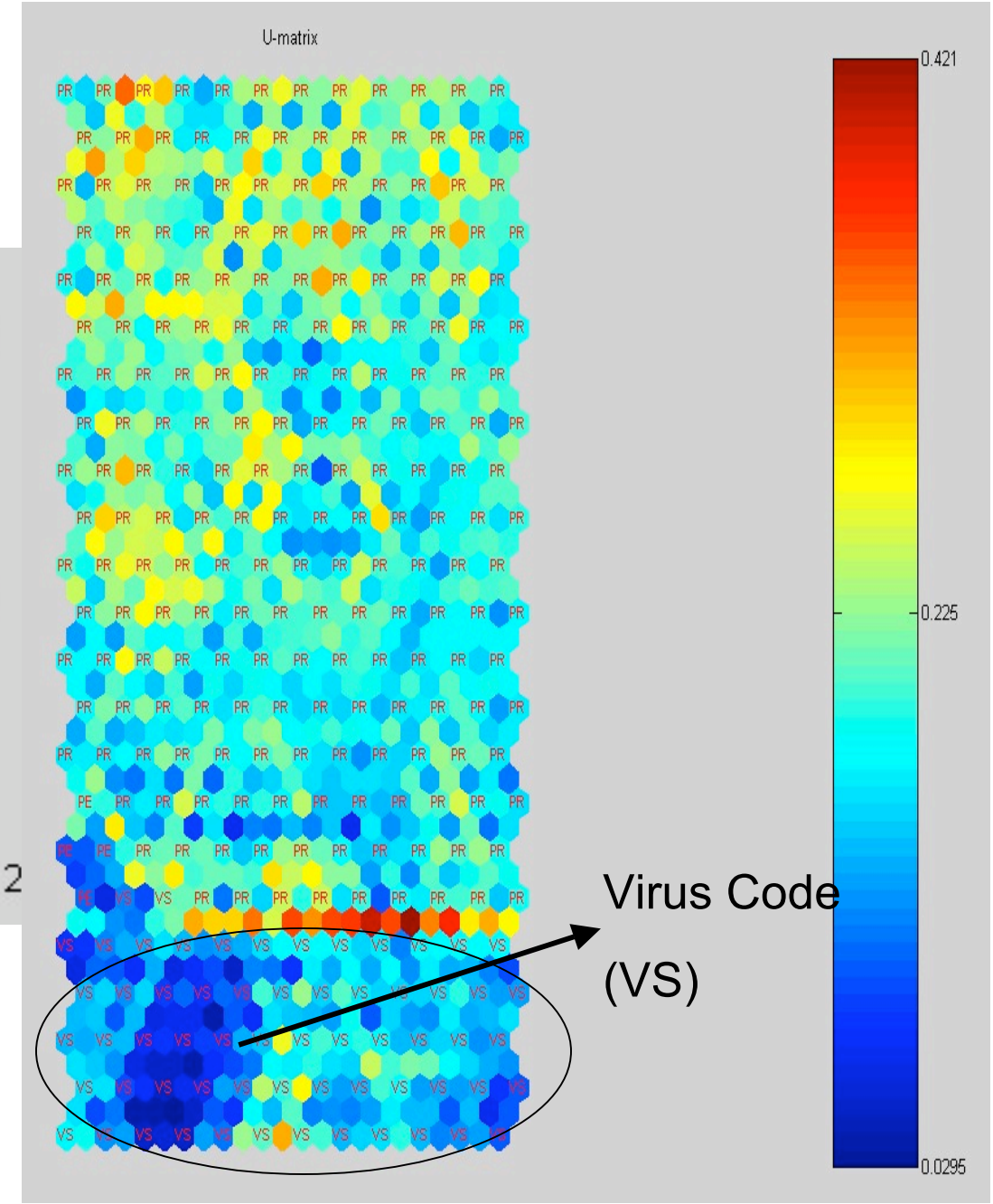




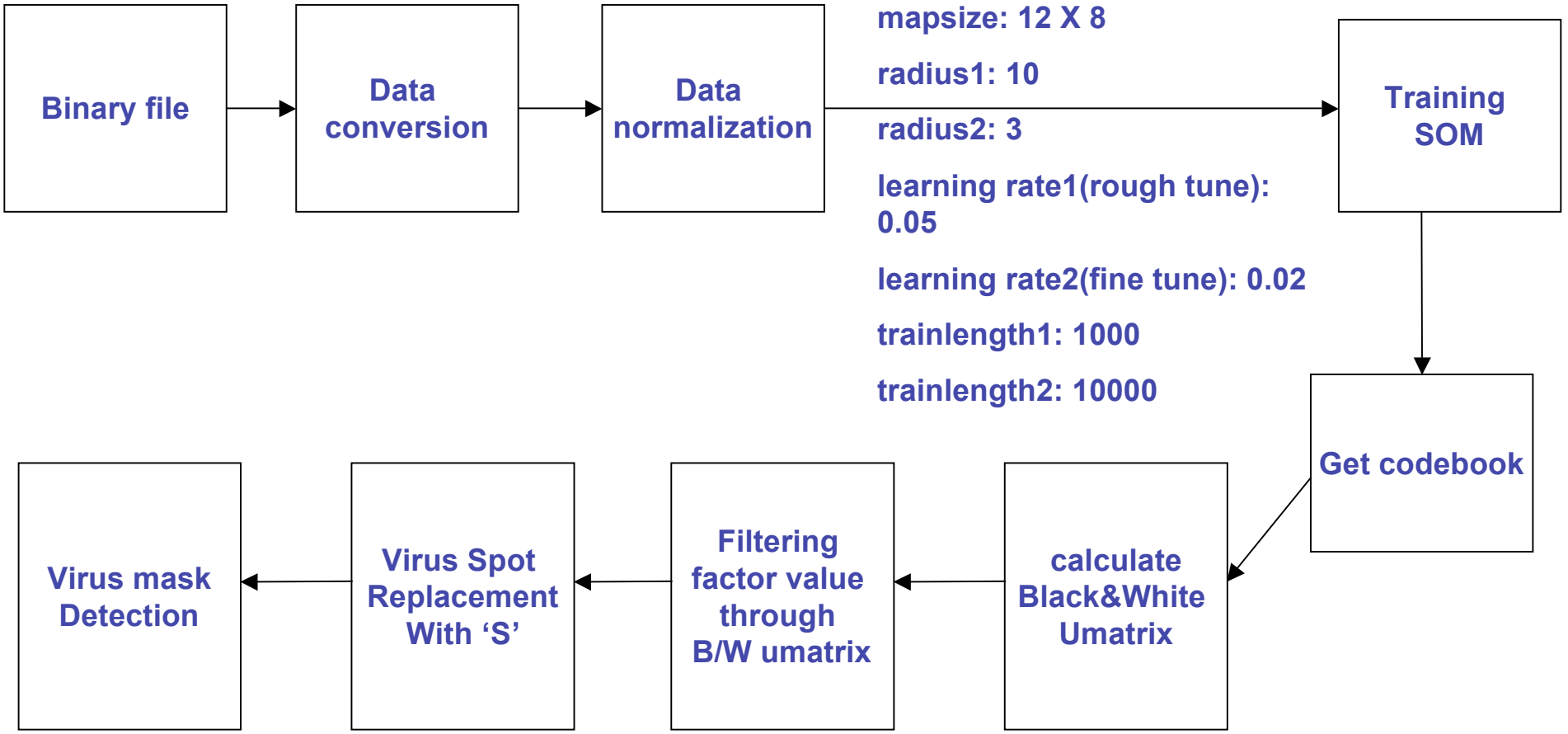
Win32.apparition virus



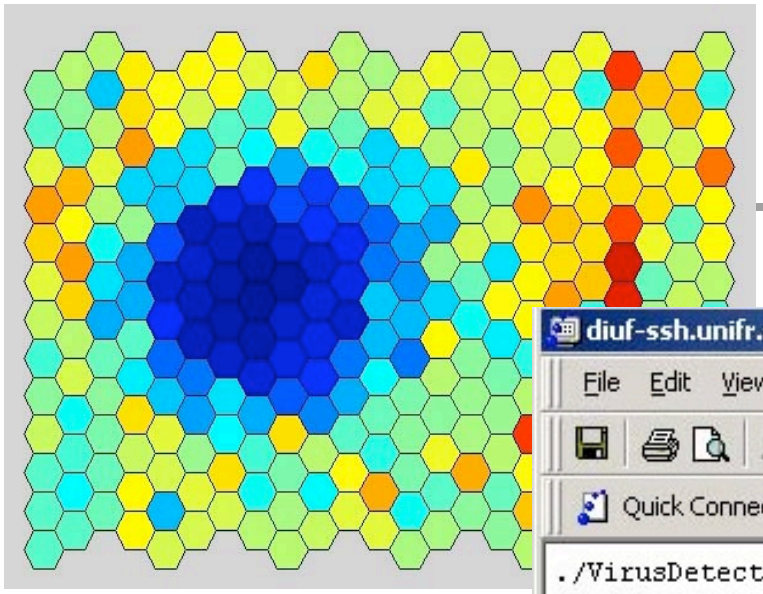
Win32.HLLP.Semisoft virus



Virus Detection Process in *VirusDetector*



Virus Detection Ex. CIH 1.4



diuf-ssh.unifr.ch - default - SSH Secure Shell

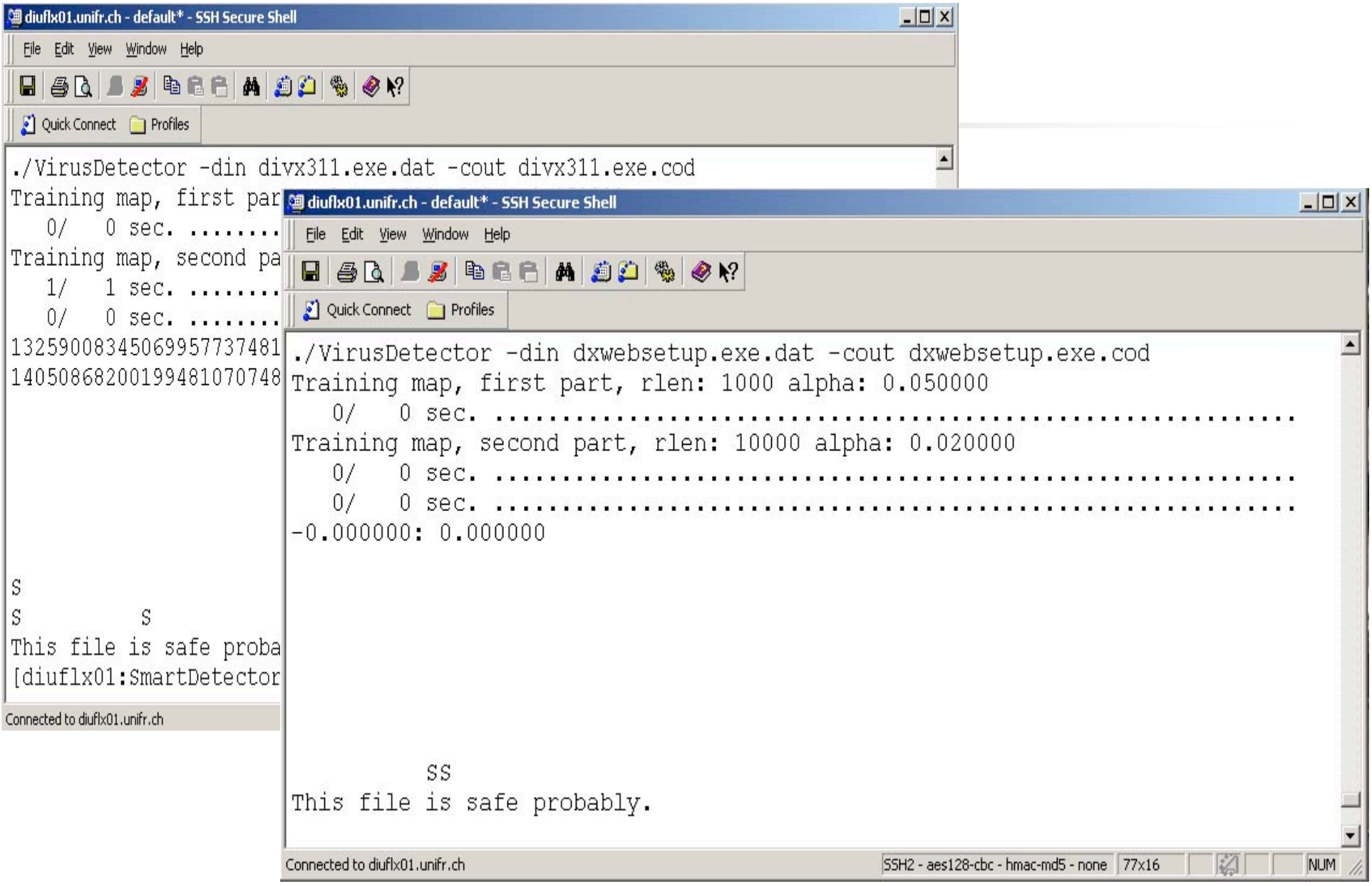
File Edit View Window Help

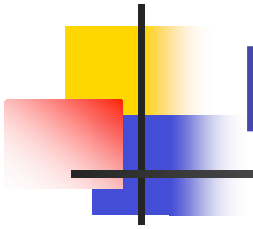
Quick Connect Profiles

```
./VirusDetector -din testcihl4.dat -cout testcihl4.cod
Training map, first part, rlen: 1000 alpha: 0.050000
   0/   0 sec. ..............................................................
Training map, second part, rlen: 10000 alpha: 0.020000
   1/   1 sec. ..............................................................
   0/   0 sec. ..............................................................
244863529380724078120358040912891766422947846441016171404634415901312630339916174
422649182169624316977263517440351606667514741691832987177935177655851386142 72.00
0000: 0.000000


      SS
     SSSS
      SSS
     SSSS
      SS

This is a virus-infected file!
```

Connected to diuf-ssh.unifr.ch | SSH2 - aes128-cbc - hmac-md5 - none | 81x18

Ex. Normal executable files

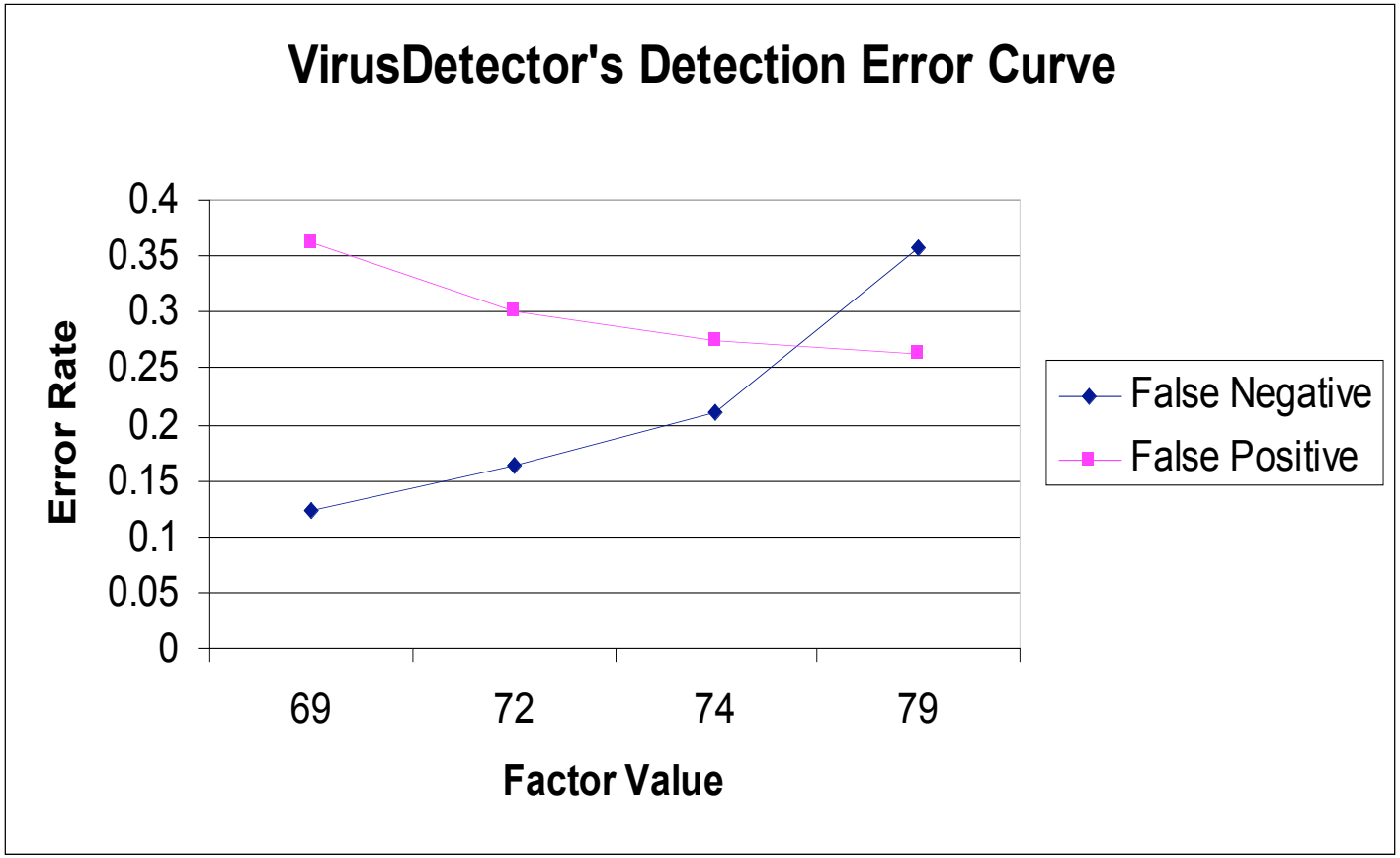


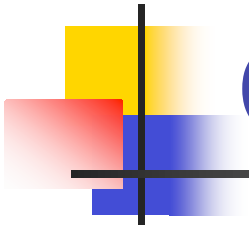


Result of Virus Detection

- Test data: 790 virus-infected files
- Unencrypted parasitic viruses
  - Win9x viruses: 291
    - False negative: 0.09 (approx. 9%)
  - Win32 viruses: 499
    - False negative:0.2065 (approx. 21%)
- Normal Win exe files: 80
  - False positive: 0.3 (approx. 30%)
- Polymorphic & Encrypted parasitic viruses
  - Win9x Encrypted viruses: 30
    - False negative: 0.03 (3%)
  - Win9x Polymorphic viruses: 15
    - False negative: 0.13 (13%)
  - Win32 Encrypted viruses: 30
    - False negative: 0.13 (13%)
  - Win32 Polymorphic viruses: 50
    - False negative: 0.42 (42%)

Error Rate of *VirusDetector*





Conclusions

- Towards unknown or variants of known viruses detection approach with SOM
- Classical virus detection mostly relies on virus-signatures
- Unknown or variants of known viruses are presented (84 % detection rate with 30% false positive)
- Combination of the classical methods and this SOM based non-signature detection can help systems more secure