Journal of Theoretical and Applied Information Technology
31st December 2018. Vol.96. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MATCHING ALGORITHMS FOR INTRUSION DETECTION SYSTEM BASED ON DNA ENCODING

OMAR FITIAN RASHID[1], ZULAIHA ALI OTHMAN[1], SUHAILA ZAINUDIN[1]

[1]Faculty of Information Science and Technology, University Kebangsaan Malaysia, Bangi, Malaysia
Email: Omaralrawi08@yahoo.com, zao@ukm.edu.my, and suhaila.zainudin@ukm.edu.my

ABSTRACT

Pattern matching algorithms are usually used as detecting process in intrusion detection system. The efficiency of these algorithms is affected by the performance of the intrusion detection system which reflects the requirement of a new investigation in this field. Four matching algorithms and a combined of two algorithms, for intrusion detection system based on new DNA encoding, are applied for evaluation of their achievements. These algorithms are Brute-force algorithm, Boyer-Moore algorithm, Horspool algorithm, Knuth-Morris-Pratt algorithm, and the combined of Boyer-Moore algorithm and Knuth–Morris–Pratt algorithm. The performance of the proposed approach is calculated based on the executed time, where these algorithms are applied on NSL-KDD dataset. The obtained results showed that the average time for matching for all NSL-KDD dataset records, based on Brute-force algorithm, Boyer-Moore algorithm, Horspool algorithm, Knuth-Morris-Pratt algorithm, and the combined of Boyer-Moore algorithm and Knuth–Morris–Pratt algorithm are equal to 18.4, 11.5, 9.23, 7.5, and 23.2 seconds respectively. These results demonstrated that using single algorithm achieved better time than combined algorithms, and Knuth-Morris-Pratt algorithm gives the best result than the rest of the other three algorithms. The results are reasonable and acceptable when they are compared with previous systems.

Keywords: *Intrusion detection, DNA Encoding, Pattern Matching Algorithm, Knuth-Morris-Pratt Algorithm, Boyer-Moore Algorithm*

1. INTRODUCTION

Various procedures have been suggested for intrusion detection system. Pattern matching algorithm is one that always used in intrusion detection system for detection operation, based on the efficiency of matching process, and it is used to calculate the intrusion detection system performance [1]. Various pattern matching algorithms are used as reliable computer system techniques that are applied for the detection of intruders. A misuse intrusion detection system is provided to monitor the network and capture packets, then analyse these packets and get report, this system used two pattern matching algorithms for the detection intrusion and these are Brute-force algorithm and Knuth-Morris-Pratt algorithm. The achieved false alarm is very high; therefore, a string matching algorithm is proposed in order to reduce the false alarming result [2]. Kala and Christy [3], discussed the signature based intrusion detection system and the different pattern matching algorithms that have been used for these systems. They used Brute-force and Knuth-Morris-Pratt algorithms and they got high false alarm results. Then, the system is enhanced by using Less False Alarm Algorithm which is proposed by [2] in order to reduce the false alarm result. A network intrusion detection system is proposed by [4] where Boyer-Moore algorithm is used. The classifier is combined based on many decision trees that obtained by individual trees.

Prabha and Sukumaran [5] described the concept of pattern matching algorithm and proposed a new method by improved the pattern matching algorithm. This method is reduced the comparison number of the characters, it is fast, and more reliable in security. An anomaly intrusion detection system based on the information that gained from the behaviour of authorized users over two years is proposed [6]. This study used data structure to encode activities and behaviours of the users. Negative selection method is applied to get a set of anomalous detectors. Then, the Knuth-Morris-Pratt algorithm is used for matching and this was tested on real data.

Rashid et al. [7] proposed a novel Intrusion Detection System based on DNA sequence, this

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

done firstly by converting network traffic to DNA sequence by using Cryptography encoding method, then extract keys, and finally used Horspool algorithm for matching process. Aburomman and Reaz [8] developed an intrusion detection system by improving the process of multiclass classification, and found a new selection model based on differential evolution. Anomalies classifier based on machine learning is presented [9] that used to classify activities automatically for intrusion detection system.

Hamsaveni and Gunasekaran [10] described the concept of pattern matching algorithms and proposed Signature intrusion detection system based on Logo Pattern Matching algorithm. The proposed system is reduced the false alarm results for high level and detects the intruder efficiently and these results are implemented in SNORT. Dang et al., [11] presented a multiple pattern matching algorithms that reduce the comparison of characters and the space of the memory based on graph transition structure and search technique. Two intrusion detection system techniques based on neural network are presented that are used for feature extraction. The first one used the principal component analysis (PCA) method, and the second one is used both Bayesian method and Kernel Principal Component Analysis (KPCA). These systems are achieved by using java [12]. Amiri et al., [13] proposed two feature selection methods for intrusion detection system and these methods are applied a measure of the feature goodness and they used both a linear and a non-linear measure Then an intrusion detection system is built by utilized the Least Squares Support Vector Machine, and this system implemented based on KDDCup 99 dataset.

However, the above various techniques, that are applied based on matching algorithms, have achieved good results, but the DNA encoding does not used in such application. Therefore, the current paper is exhibited the results of the investigation of the four matching algorithms that are applied to intrusion detection system based on a new DNA encoding approach to reduce the matching time and to find the most suitable and faster one for this system. In addition, the achieved result based on the best algorithm, that is applied on KDDCup 99, is compared with the previous published results that have been used different techniques.

2. MATERIAL AND METHODS

The intrusion detection system based on DNA encoding consists of two phases: training phase and testing phase. Training phase consists of two steps; the first step is converting random network traffic from 10% KDDCup 99 dataset to DNA sequences. The following shows an example of network traffic and its equivalent DNA sequences:

Network traffic: 12,tcp,http,SF,51,8127,0,0,0,2,0,1,0,1,0,0,0,0,1,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,246,0.96,0.01,0.00,0.00,0.00,0.00,0.00,0.00

DNA sequences: TCAGTTAGCGCCAGGCGT
ACGTCAGGCTCAGTTGCCGTAGTAGTAGTT
GTATCAGTATCAGTAGTAGTAGTATCAGTA
GTAGTATCATCAGTAGTGGTAGTAGTAGTG
GTAGTAGTAGTGGTAGTAGTAGTGGTAGTA
TCAGTGGTAGTAGTAGTGGTAGTAGTAGTG
GTAGTAGTTACGACGGTTTATTAAGTAGTG
CACTAAGTAGTGGTATCAGTAGTGGTAGTA
GTAGTGGTAGTAGTAGTGGTAGTAGTAGTG
GTAGTAGTAGTGGTAGTAGTAGTGGTAGTA

The second step is used Teiresias algorithm to extract two keys and their positions. Table 1 shows the extracted keys and their positions.

Table 1. The extracted keys and their positions

Number	Keys	Positions
1	CGCCA	6
2	GCGTG	9

In testing phase, all NSL-KDD records are converted to DNA sequences, then four algorithms and with the combined of two algorithms for matching are applied. These algorithms are applied 30 times for intrusion detection system and the matching process is either based on keys only or based on both keys and their positions. Then, the matching time is determined from the above applications. From these results, the best algorithm is defined depending on running time. Matching algorithm has an important effect on the performance of the intrusion detection system in order to detect the attack quickly. These algorithms are:

Journal of Theoretical and Applied Information Technology
31st December 2018. Vol.96. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.1 Brute-force Algorithm

A string matching algorithm that checking each position from the first position of the text (position 0) to the final position (m - n), where m is the text size and n is the key length. This is done by comparing every character in the key with the corresponding character in the text. If all the characters are matched, then the key are found in text, otherwise the key are not found in text [14]. The steps of applying Brute-force algorithm on intrusion detection system based on DNA encoding are shown in Algorithm 1.

Algorithm 1: Brute-force algorithm

Input: NSL-KDD dataset, keys and positions.
Output: Classify either normal or attack.

1. Convert records to DNA sequences.
2. Search for matching based on keys and positions.

```
i ← 1
found ← 0
while  i < =  record_length and found = 0
Block (i) ← Substring (record, i,  key_length)
count ← 0
for j ← 1 to key_length
if Substring(Block (i), j, 1) = Substring(key, j, 1)
        count ← count + 1
         if count = key_length
         found ← 1
         end if
end if
end for
i ← i + 1
end while
```

3. Determine records either normal or attack

```
if found = 0
    record is normal
else
    record is attack
end if
```

Table 2 shows an example of the application of Brute-force algorithm. It is explaining the procedure of looking for the key "TGAAC" in the sequence "TTCAGGTCTGAACA", where the Brute-force algorithm performs 16 characters comparisons.

Table 2. Example of the application of Brute force algorithm

First attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
1	2												
T	G	A	A	C									

Second attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
	1	2											
	T	G	A	A	C								

Third attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
		1											
		T	G	A	A	C							

Fourth attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
			1										
			T	G	A	A	C						

Fifth attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
				1									
				T	G	A	A	C					

Sixth attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
					1								
					T	G	A	A	C				

Seventh attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
						1	2						
						T	G	A	A	C			

Eighth attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
							1						
							T	G	A	A	C		

Ninth attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
								1	2	3	4	5	
								T	G	A	A	C	

Journal of Theoretical and Applied Information Technology
31st December 2018. Vol.96. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.2 Boyer-Moore Algorithm

Boyer-Moore matching algorithm has been considered as the standard benchmark for the practical string search literatures. This algorithm is pre-process the searched key but not the string that searched in. Therefore, it is not needing to look for every character in the string that searched in, but, skips some of the characters. As a rule, the algorithm becomes faster when the key length becomes taller [15]. The shift process is determined based on two rules; the first rule is bad character which considered with the character when the comparison is failed then found the next character occurrence to the left. The second rule is good suffix and comparisons are done at the end of pattern. The steps of applying the Boyer-Moore algorithm on intrusion detection system based on DNA encoding are shown in Algorithm 2.

Algorithm 2: Boyer-Moore algorithm

Input: NSL-KDD dataset, keys and positions.
Output: Classify either normal or attack.

1. Convert records to DNA sequences.
2. Search for matching based on keys and positions.

```
Shift ← 1
i ← 1
found ← 0
for  i  =  1 to record_length step shift
    Block (i) ← Substring (record, i, keylength)
    count ← 0
    for j ← key_length to 1 step-1
     if Substring (Block (i), j, 1) = Substring (key, j, 1)
       count ← count + 1
         if count = key_length
         found ← 1
         end if
       end if
     end for
shift ← bad_character_table (Substring (Block (i), j,
1)) – key_length + (5 - count)
end for
```

3. Determine records either normal or attack

```
if found = 0
   record is normal
else
   record is attack
end if
```

The bad character table for the same example is shown in Table 3. The following example (Table 4) is an application of Boyer-Moore algorithm. It is explaining the procedure of looking for the key "TGAAC" in the sequence "TTCAGGTCTGAACA", where the Boyer-Moore algorithm performs 10 characters comparisons.

Table 3. Bad character table

A	A	C	G	T
Bc[a]	1	5	3	4

Table 4. Example of the application of Boyer-Moore algorithm

First attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
				1									
T	G	A	A	C									
Shift by: 3 (**Gs**[4] = **Bc**[G] – 5+5)													
Second attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
						2	1						
			T	G	A	A	C						
Shift by: 3 (**Gs**[3] = **Bc**[T] – 5+4)													
Third attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
										1			
						T	G	A	A	C			
Shift by: 1 (**Gs**[4] = **Bc**[A] – 5+5)													
Fourth attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
											1		
							T	G	A	A	C		
Shift by: 1 (**Gs**[4] = **Bc**[A] – 5+5)													
Fifth attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
								5	4	3	2	1	
								T	G	A	A	C	

2.3 Horspool Algorithm

A string searching algorithm and it is pre-process the key that is being searched. Therefore, it is no need to look for every character in the string that searched [16]. The Horspool is suitable for bio application. Key is the string to be searched for, and the text is the string being searched. Horspool achieves sub-linear run time using the following

Journal of Theoretical and Applied Information Technology
31st December 2018. Vol.96. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

two ideas [17]: Right-to-left scan; shift pattern right by some amount when a mismatch occurs. Bad character rule is applied when the character comparison in text is failed. The next character occurrence in key is found. Then, shift the key to set occurrence with mismatched occurrence in the text. The steps of applying the Horspool algorithm on intrusion detection system based on DNA encoding are shown in Algorithm 3.

Algorithm 3: Horspool algorithm

Input: NSL-KDD dataset, keys and positions.
Output: Classify either normal or attack.

1. Convert records to DNA sequences.
2. Search for matching based on keys and positions.

```
i ← 1
found ← 0
while  i < =  record_length and found = 0
    Block (i) ← Substring (record, i, keylength)
    count ← 0
for j ← key_length to 1 step-1
if Substring (Block (i), j, 1) = Substring (key, j, 1)
count ← count + 1
if count = key_length
found ← 1
end if
end if
end for
i ← i + bad_character_table (5 – count)
end while
```

3. Determine records either normal or attack

```
if found = 0
record is normal
else
record is attack
end if
```

The following example (Table 5) is an application of the Horspool algorithm. It is explaining the procedure of looking for the key "TGAAC" in the sequence "TTCAGGTCTGAACA", where the Horspool algorithm performs 9 characters comparisons.

Table 5. Example of the application of Horspool algorithm

First attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
				1									
T	G	A	A	C									
Shift by: 3 (**Bc**[G])													

Second attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
			2				1						
			T	G	A	A	C						
Shift by: 4 (**Bc**[T])													

Third attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
											1		
							T	G	A	A	C		
Shift by: 1 (**Bc**[A])													

Fourth attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
								2	3	4	5	1	
								T	G	A	A	C	

2.4 Knuth-Morris-Pratt Algorithm

A string matching algorithm that converts the search string to a finite state machine, and then runs the machine with the string to be searched as the input string. This algorithm is created information array that contains information about how the keyword matches against shifts of itself [10]. The steps of applying the Knuth-Morris-Pratt algorithm on intrusion detection system based on DNA encoding are shown in Algorithm 4.

Algorithm 4: Knuth-Morris-Pratt algorithm

Input: NSL-KDD dataset, keys and positions.
Output: Classify either normal or attack.

1. Convert records to DNA sequences.
2. Search for matching based on keys and positions.

```
Shift ← 1
i ← 1
found ← 0
for  i  =  1 to record_length step shift
Block (i) ← Substring (record, i, keylength)
count ← 0
for j ← 1 to key_length
if Substring (Block (i), j, 1) = Substring (key, j, 1)
count ← count + 1
if count = key_length
found ← 1
end if
end if
end for
shift ← j - kmpNext (count)
```

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

end for

3. Determine records either normal or attack
if found = 0
record is normal
else
record is attack
end if

The kmpNext table for the same example is shown in Table 6. The following example (Table 7) is an application of Knuth-Morris-Pratt algorithm. It is explaining the procedure of looking for the key "TGAAC" in the sequence "TTCAGGTCTGAACA", where the Knuth-Morris-Pratt algorithm performs 16 characters comparisons.

Table 6. The KmpNext table

I	0	1	2	3	4
x [i]	T	G	A	A	C
KmpNext [i]	-1	0	0	0	0

Table 7. Example of the application of Knuth-Morris-Pratt algorithm

First attempt

T	T	C	A	G	G	T	C	T	G	A	A	C	A
1	2												
T	G	A	A	C									

Shift by: 1 (i – **kmpNext** [i] = 1- 0)

Second attempt

T	T	C	A	G	G	T	C	T	G	A	A	C	A
	1	2											
	T	G	A	A	C								

Shift by: 1 (i – **kmpNext** [i] = 1- 0)

Third attempt

T	T	C	A	G	G	T	C	T	G	A	A	C	A
		1											
		T	G	A	A	C							

Shift by: 1 (i – **kmpNext** [i] = 0 - -1)

Fourth attempt

T	T	C	A	G	G	T	C	T	G	A	A	C	A
			1										
			T	G	A	A	C						

Shift by: 1 (i – **kmpNext** [i] = 0 - -1)

Fifth attempt

T	T	C	A	G	G	T	C	T	G	A	A	C	A
				1									
				T	G	A	A	C					

Shift by: 1 (i – **kmpNext** [i] = 0 - -1)

Sixth attempt

T	T	C	A	G	G	T	C	T	G	A	A	C	A
					1								
					T	G	A	A	C				

Shift by: 1 (i – **kmpNext** [i] = 0 - -1)

Seventh attempt

T	T	C	A	G	G	T	C	T	G	A	A	C	A
						1	2						
						T	G	A	A	C			

Shift by: 1 (i – **kmpNext** [i] = 1 - 0)

Eighth attempt

T	T	C	A	G	G	T	C	T	G	A	A	C	A
							1						
							T	G	A	A	C		

Shift by: 1 (i – **kmpNext** [i] = 0 - - 1)

Ninth attempt

T	T	C	A	G	G	T	C	T	G	A	A	C	A
								1	2	3	4	5	
								T	G	A	A	C	

2.5 Combined Two Algorithms

Apply the combined of two algorithms that proposed by [18], these algorithms are Boyer Moore algorithm and Knuth Morris Pratt algorithm. The main idea is to find matches between the pattern and the string with a smaller number of shifts, larger shift is chosen from two shifts that got from Knuth-Morris-Pratt table (KmpNext) and Boyer-Moore tables (bad character rule and the good suffix rule). The steps of applying the combined of Boyer-Moore algorithm and Knuth-Morris-Pratt algorithm on intrusion detection

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

system based on DNA encoding are shown in Algorithm 5.

Algorithm 5: Combination of Boyer Moore and Knuth–Morris–Pratt

Input: NSL-KDD dataset, STR (keys and positions)
Output: Classify either normal or attack.

1. Convert records to DNA.
2. Search for matching based on STR.

```
Shift ← 1
i ← 1
found ← 0
for  i  =  1 to record_length step shift
    Block (i) ← Substring (record, i, keylength)
    count ← 0
    for j ← 1 to 2
      if Substring (Block (i), i+j-1, 1) = Substring (key,
       j, 1)  and Substring (Block (i),
 i+ key_length  - j, 1) = Substring (key, key_length +1
- j, 1)
       count ← count + 2
        if count = key_length -1
          if Substring (Block (i), i+2, 1) = Substring (key,
3, 1)
          count ← count + 1
          end if
        end if
    if count = key_length
    found ← 1
    end if
  end if
 end for
 shift1 ← j - kmpNext (count)
 shift2 ← bad_character_table (Substring (Block (i), j,
1)) – key_length + (5 - count)

   if shift1 >= shift2 then shift =shift1
   else shift =shift2
   end if
end for
```

3. Determine records either normal or attack

```
    if found = 0 record is normal
    else record is attack
    end if
```

The following example (Table 8) is an application of the combined algorithms. It is explaining the procedure of looking for the key "TGAAC" in the sequence "TTCAGGTCTGAACA", where the algorithms perform 11 characters comparisons.

Table 8. Example of the application of combined algorithms

First attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
1				1									
T	G	A	A	C									
Shift based on KmpNext = 1, Shift based on **Gs** = 3, then Shift by 3 characters													
Second attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
			1				1						
			T	G	A	A	C						
Shift based on KmpNext = 1, Shift based on **Gs** = 4 , then Shift by 4 characters													
Third attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
							1				1		
							T	G	A	A	C		
Shift based on KmpNext = 1, Shift based on **Gs** = 1, Shift by 1 characters													
Fourth attempt													
T	T	C	A	G	G	T	C	T	G	A	A	C	A
								1	2	3	2	1	
								T	G	A	A	C	

The NSL-KDD dataset is proposed by Tavallaee et al. [19] which reduced the number of the original KDD 99 dataset. It contains 41 features like the 99 KDDCup dataset features. Both KDDCup 99 and NS-KDD datasets records are classified either normal or attacks, and attacks records can be classified in to four types. These types are Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R). The advantages of NSL- KDD dataset are:

- There are no duplicate records in training and testing datasets.
- Makes the evaluation more accurate and efficient.
- The train and test records number are acceptable, which mean that all the records can

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

be used instead of choosing some random records.

The matching time for all four algorithms is calculated by using all records of NSL-KDD (KDDTest+) dataset that equal to 22544 records.

3. RESULTS AND DISCUSSIONS

Experimental environment: operating system is Microsoft Windows 10 Professional, CPU is Intel 2.50GHz, and memory is 4.00 GB. Four algorithms are applied; each one is run 30 times. The results of matching time for the whole 30 attempts for intrusion detection system based on DNA encoding data by using Brute-force, Boyer-Moore, Horspool, Knuth-Morris-Pratt, and the combined of Boyer-Moore algorithm and Knuth–Morris–Pratt algorithm are determined. The matching time (in term of seconds) obtained by these algorithms for each attack using keys only (K) and keys and their positions (P) on NSL-KDD dataset and the summary of these results are presented in Tables 9 and 10 respectively.

Table 9. Time results (in term of seconds) obtained by applying matching algorithms

No.	Matching Time (Sec)					
	Brute force (K)	Boyer-Moore (K)	Horspool (K)	KMP (K)	KMP & BM (K)	(P)
1	22	12	11	9	24	3
2	23	13	8	8	22	3
3	20	13	9	9	26	3
4	20	13	8	8	27	3
5	18	13	14	8	21	3
6	16	10	8	6	25	**2**
7	17	10	9	8	22	3
8	15	11	9	7	18	**2**
9	14	10	8	6	17	3
10	16	12	10	7	20	4
11	17	10	7	8	22	3
12	18	10	7	6	22	3
13	21	12	9	8	22	3
14	18	11	7	7	18	3
15	21	15	13	10	28	3
16	13	8	6	5	16	**2**
17	18	14	10	9	23	3
18	18	13	13	8	26	3
19	14	9	8	6	24	**2**
20	18	12	14	8	22	4
21	20	12	9	7	26	3
22	19	12	9	8	22	3
23	13	8	7	5	**16**	3
24	20	12	10	7	32	3
25	20	12	10	8	29	3
26	23	12	9	9	23	3
27	19	10	8	6	24	4
28	19	12	8	7	27	3
29	21	12	9	9	27	3
30	21	12	10	8	25	3

Table 10. The summary of time results obtained from the application of matching algorithms

Matching Time(Sec)				
Method	**Best**	**Worst**	**Average**	**Standard deviation**
Brute-force	13	23	18.4	2.76
Boyer-Moore	8	15	11.5	1.63
Horspool	6	14	9.23	2.04
KMP	5	10	7.5	1.25
KMP and Boyer Moore	16	32	23.2	3.85
Keys & Positions	2	4	2.96	0.49

A comparison between the matching times resulted from the application of all algorithms shown in Table 10 are presented in Figure 1. As outlined in the figure below, the best achieved average time is obtained by using Knuth-Morris-Pratt algorithm and it is equal to 7.5 seconds for all NSL-KDD dataset records.

Journal of Theoretical and Applied Information Technology
31st December 2018. Vol.96. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

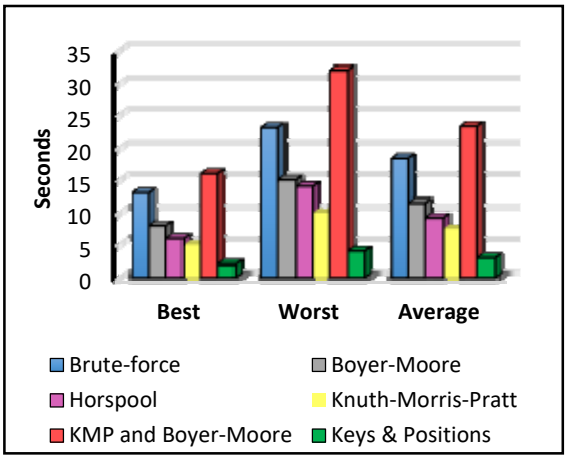

Figure 1. Illustrated the summary of the matching time results by the application of the four algorithms

Table 11 presented the results of the matching time obtained by the present system and these are compared with the published time of the intrusion detection system based on neural network that has been used two techniques for feature extraction. The first one is used principal component analysis (PCA) method and the second one used both Bayesian method and KPCA. The total records number that have been used are equal to 18571 records (13000 for training + 5571 for testing) [12]. These results are compared with the results based on keys only and on both keys and their positions that have been obtained in the current work by using Knuth-Morris-Pratt algorithm which gives the best result. From the table, it is clear that the matching time obtained by the method of the present system, are good. This system gives a better matching time than the previous system. The results of the proposed system and the published one are illustrated in Figure 2.

Table 11. Comparison between the matching times of the proposed system with the published ones

Method		Time (Sec)
Pattewar and Sonawane [12]	PCA	33.39
	Bayesian and KPCA	28.677
Proposed System	**Keys only**	**7.5**
	Keys and Positions	**2.96**

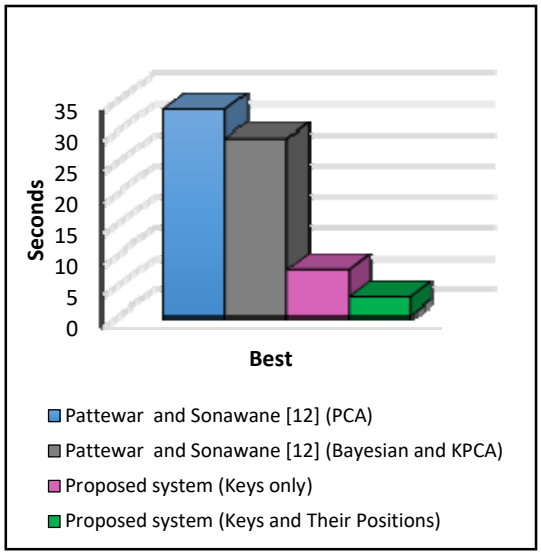

Figure 2. Comparison between the matching times of the proposed system with the published ones

Table 12 exhibited the time values (in term of minutes) obtained from the present system. These values are compared with intrusion detection system that has been used the least squares support vector machine classification method in order to improve machine learning where the experiments are based on KDDCup 99 dataset mentioned by (Amiri et al., [13]). Therefore, the current system is applied on KDDCup 99 dataset for different classes (Normal, DoS, Probe, R2L, and U2R) to compare the obtained results based on keys only and on both keys and their positions by using Knuth-Morris-Pratt algorithm which gives the best result with the published values. From the table, it is clear, that the matching time obtained by the method of the present system is good. This system gives a better matching time than the previous one and these results highlighted the success of the proposed system. The results of the proposed system and the published one are illustrated in Figure 3.

Table 12. Comparison between the matching times of the proposed system with the published ones based on every dataset class separately (time in minutes)

Method		Normal	DoS		Probe	R2L	U2R
Amiri et al., [13]		20	20		20	21	20
Proposed System	**K**	0.3	3.15		0.05	0.2	0.01
	P	0.08	0.98		0.01	0.05	0.01

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

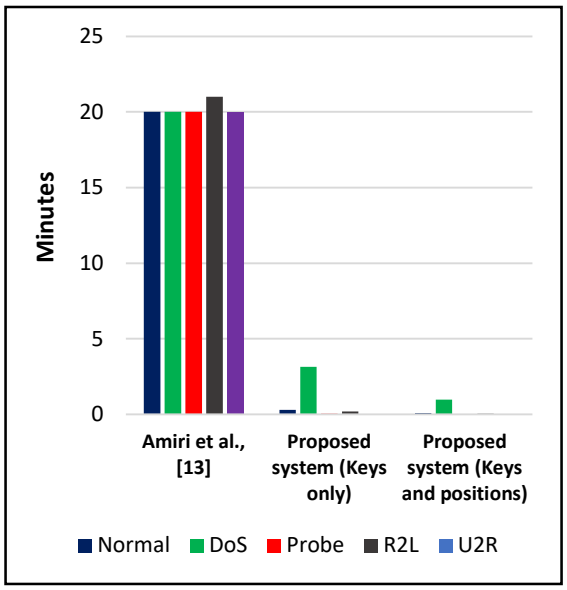

Figure 3. Comparison between the matching times of the proposed system with the published ones based on every dataset class separately

4. CONCLUSION

The present paper has shown the different matching times for intrusion detection obtained by applying four matching algorithms and the combined of two algorithms based on new DNA encoding. These algorithms are Brute-force algorithm, Boyer-Moore algorithm, Horspool algorithm, Knuth-Morris-Pratt algorithm, and the combined of Boyer-Moore algorithm and Knuth–Morris–Pratt algorithm. The system is executed by using NSL-KDD dataset as source information. The performance of this system is evaluated based on matching time, and the calculating time is applied either based on keys only or based on keys and their positions.

The achieved results from the present experiment showed that the average matching time by using Brute-force algorithm, Boyer-Moore algorithm, Horspool algorithm, Knuth-Morris-Pratt algorithm, and the combined of Boyer-Moore algorithm and Knuth–Morris–Pratt algorithm based on keys only are equal to 18.4, 11.5, 9.23, 7.5, and 23.2 seconds respectively. While, the matching time for all applied matching algorithms by using the keys and their positions is equal to 2.96 seconds.

Therefore, the new finding of this study indicated that the Knuth-Morris-Pratt algorithm is the most suitable and fast algorithm that can be used for intrusion detection system based on DNA encoding. Also, when both keys and their positions are used, the detection time is faster than only keys are used. In addition, the results of the current system are better than the published ones. The execution of the applied matching algorithms on intrusion detection system based on DNA encoding do not have any difficulties to be implemented in the future, because it is easy to apply and has fast matching time.

ACKNOWLEDGEMENT

This research is funded by KPT research grant FRGS/1/2016/ICT02/UKM/02/8. Also. we would like to thank to University Kebangsaan Malaysia (UKM) for supporting researchers in conducting research

REFERENCES:

[1] Z. Qu, and X. Huang, "The improving pattern matching algorithm of intrusion detection", *Procedia Engineering* 15: 2841-2846, 2011.

[2] Lata, and K. Indu, "Novel algorithm for intrusion detection system", International Journal of Advanced Research in Computer and Communication Engineering 2(5), May 2013.

[3] T. S. Kala, and A. Christy, "A Pattern Matching Algorithm for Reducing False Positive in Signature Based Intrusion Detection System", *International Journal of Engineering and Technology (IJET)* 8(2), April - May 2016.

[4] K. V. Kumar, and B. Veerendranath, "Data Mining Model for Network Intrusion Detection Using Boyer-Moore Algorithm", *International Journal of Advanced Research in Computer Science & Technology* 2(4), October - December 2014.

[5] K. Prabha, and S. Sukumaran, "Improved Single Keyword Pattern Matching Algorithm for Intrusion Detection System", *International Journal of Computer Applications* 90(9), March 2014.

[6] C. B. G. Maldonado, M. S. Penas, and M. V. L. Lopez, ""Negative Selection and Knuth Morris Pratt Algorithm for Anomaly Detection", *IEEE Latin America Transactions* 14(3): 1473-1479, March 2016.

[7] O. F. Rashid, Z. A. Othman, and S. Zainudin, "A Novel DNA Sequence Approach for Network Intrusion Detection System Based on Cryptography Encoding Method", *International Journal on Advanced Science,*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Engineering and Information Technology 7(1): 183-189, 2017.

[8] A. A. Aburomman, and M. B. Reaz, "A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems", *Information Sciences* 414: 225–246, 2017.

[9] Q. S. Qassim, A. M. Zin, and M. J. Ab Aziz, "Anomalies Classification Approach for Network-based Intrusion Detection System", *International Journal of Network Security* 18(6): 1159-1172, 2016.

[10] R. Hamsaveni, and G.Gunasekaran, "A Secured Pattern Matching Technique for Intrusion Detection System in Wireless Sensor Network", *International Journal of Computer Networks and Wireless Communications* 6(3), May - June 2016.

[11] N. L. Dang, D. Le, and V. T. Le, "A New Multiple-Pattern Matching Algorithm for the Network Intrusion Detection System", *International Journal of Engineering and Technology* 8(2), April 2016.

[12] T. M. Pattewar, and H. A. Sonawane, "Neural Network based Intrusion Detection using Bayesian with PCA and KPCA Feature Extraction", *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)*, Nov. 2015.

[13] F. Amiri, M. R. Yousefi, C. Lucas, A. Shakery, and N. Yazdni, "Mutual Information-based Feature Selection for Intrusion Detection Systems", *Journal of Network and Computer Applications* 34: 1184–1199, 2011.

[14] S. Saha, "Network Intrusion Detection System Using String Matching", *Internet* 1-46, 2010.

[15] S. S. Sheik, S. K. Aggarwal, A. Poddar, N. Balakrishnan and K. Sekar,"A Fast Pattern Matching Algorithm", *Bioinformatics Centre and Supercomputer Education and Research Centre*, Indian Institute of Science, India, June 18, 2003.

[16] S. Nimisha, and G. Deepak, "String Matching Algorithms and their Applicability in Various Applications", *International Journal of Soft Computing and Engineering* I(6), January 2012.

[17] D. Gusfield, "Chapter 2 - Exact Matching: Classical Comparison-Based Methods", *Algorithms on Strings, Trees, and Sequences Book*: 16-17, 1999.

[18] R. Y. Tsarev, A. S. Chernigovskiy, E. A. Tsareva, V. V. Brezitskaya, A. Y. Nikiforov, and N. A. Smirnov, 2016. Combined string searching algorithm based on Knuth-Morris-Pratt and Boyer-Moore algorithms. *XIX International Scientific Conference Reshetnev Readings, Materials Science and Engineering* 122.

[19] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," *Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defence Applications*, 2009