Genetic Sequence Alignment Computing for Ensuring Cyber Security of the IoT Systems

Haejin Cho, Sangwon Lim, Maxim Kalinin, Vasiliy Krundyshev, Viacheslav Belenko, and Valery Chernenko

Abstract Functional integrity and sustainability of cyber systems, coherency, and connectivity of the flexible infrastructures based on the Internet of Things (IoT) concept arise in the cyber security as a stability manifestation. There is a set of polymorphic attacks that have mutations (local differences and time gaps) in the sequences of the operational acts forming an IoT-specific class of the intrusions such as a forced power consumption and a forced topology change. Our research proposes the nature-inspired technologies, namely a genetic sequence alignment computing and a sequence similarity calculation, instead of massive equity checking of multiple packets and signatures traditionally used for the system protection. The sequence alignment computing is used by nature to compare and copy the DNA and nucleotide chains keeping the stability of the bioinformatic structures, which are rather similar to sequences of the security events. The methods based on the Needleman-Wunsch, Needleman-Wunsch with the position weight matrix (PWM), Smith-Waterman, and Mauve techniques have been implemented and experimentally studied with a BoT-IoT dataset on the IoT Raspberry Pi4 platform. The experiments have shown the best effectiveness obtained with the Needleman-Wunsch algorithm with the PWM: accuracy—0.98%, precision—0.99%, and recall—0.98% achieved at detection of the polymorphic intrusions.

Keywords Alignment algorithm · Intrusion detection · Needleman-wunsch · Mauve · Pattern recognition · Polymorphism · Raspberry · Security · Sequence · Smith-waterman · DNA · IoT · IIoT · VANET
1 Introduction

Today, the Internet of Things (IoT) network concept has become an integral part of the modern information infrastructure of society. The IoT is a single network that connects the smart objects of the real world and virtual objects around us. The number and variability of such objects (smart devices) continues to grow, affecting various aspects of human life, from vehicles to mobile electronics and healthcare [26]. In other words, the IoT is not just a set of different devices and sensors connected by wired and wireless communications and the global cyber space of the Internet, but it is a closer integration of the real and virtual worlds, in which communication takes place between people and devices.

The sustainability of IoT is the availability of the system to perform its functions in the context of external influence [6]. Considering that the subsystems and components of the IoT have many horizontal connections, thus forming a heterogeneous cyber infrastructure with a large number of entities, most of which lack built-in protection mechanisms; the problem of network security in such networks is especially actual. Main attack vectors are aimed for overtaking the control by the IoT device and capturing the confidential data, as well as the purposeful node deactivation [1]. Not only objects of the virtual world—data, files, personal information—are under threat, but also physical objects of the real world. According to a report by Kaspersky Laboratory, in the first half of 2019, more than 100 million attacks were detected on the smart IoT devices, which are seven times more than in the H12018 [10]. Network infrastructure and wireless data transmission channels are the most vulnerable objects of the modern infocommunication environment [27]. Attackers using wireless links can remotely invade a target subnet or device (group of devices), intercept traffic, launch denial of service attacks (including distributed attacks), and take control of the IoT devices to create megabotnets.

In 2020, the world is facing with the COVID-2019 coronavirus pandemic, and life has almost completely moved to the network, people around the world work, study, shop and have fun online. This could not affect the goals of the cybercriminals, medical organizations, educational services, delivery, etc., faced massive DDoS attacks. A large network of hospitals in France fell victim to one of these attacks, when attackers tried to disable the infrastructure of medical institutions. As a result of the attack, hospital staff working remotely was unable to use work programs and corporate e-mail for some time. In Germany, on the first day of distance learning, a distance learning platform was attacked. The service through which teachers exchange teaching materials, homework, and tests with students was not available. In the Netherlands, the food delivery service Thuisbezorgd was unable to process orders as a result of a DDoS attack and had to refund customers. In the first quarter of 2020, there was a significant increase in the number and quality of DDoS attacks. In particular, the popularity of smart attacks that exploit vulnerabilities in the network infrastructure is noted. SYN flood is the leader among the types of attacks (92.6%); the share of attacks via ICMP continued to grow and exceeded the share of UDP and TCP floods (Kaspersky [11]).
To counter cyber attacks at the IoT, as in traditional networks, various intrusion detection systems (IDS) are applied (e.g., [17]). There are two main approaches on which most the IoT IDSs are based: anomaly detection and misuse detection. The advantages of the former are a global view on the network, the ability to detect new attacks, and the use of fewer rules (compared to signature methods); their disadvantages include a high level of errors of the first and second kind and low adaptation for changing conditions (new profiles, dynamic anomalies). The advantages of an IDS based on search for abuse are reliability, speed, low false positives for known intrusions. Its disadvantages are the strict dependence on the relevance of the signature base and the impossibility of detecting new attacks.

The work of an IDS usually consists of three stages: (1) monitoring, (2) sequence extraction and misuse detection by pattern matching, (3) signaling of attack to provide the attack response. Matching with attack patterns is currently the main method for IDS when detecting malicious activity [13]; however, it has two significant drawbacks. Firstly, the IoT-specific attacks, such as forced power consumption, forced topology change, etc., have a split sequence of operations, time intervals in the chain of actions, and some attacks may also have a non-linear sequence of actions. Secondly, there is a class of polymorphic attacks that have mutations (namely local differences, omissions, delays) in the sequences of actions during the implementation of the attack. These polymorphic attacks adapt to a wide range of conditions, operating systems, and circumstances and try to avoid scanning by defenses to infect endpoints [8]. They are called the polymorphic attacks because they have different signatures that make them difficult to identify using a single signature. For example, in order to disguise ongoing network attacks, attackers can use polymorphic shellcode to create unique attack patterns. This method usually involves encoding the payload, with the decoder placed at the front of the payload before sending it. All known the pattern matching methods do not detect such variations of a single sample.

Our research proposes new, the nature inspired, technologies, namely a genetic sequence alignment computing and a sequence similarity calculation, instead of massive equity checking of multiple packets and signatures traditionally used for the IDS composing. The aim of our research is to assess the prospects of using the biological coded sequence alignment algorithms for solving the task of detection the specific network intrusions. The novelty of our results is the fact that it has been proposed to encode network packets (or the chain of activity acts) into the sequences of the nucleotide codes for their subsequent alignment and matching with patterns. The matching operation is based on the similarity checking, not equity checking, as it was usually applied for the intrusion detection. Therefore, our contribution to the cyber security is the proposed bio-inspired approach that, in contrast to traditional methods, makes it possible to detect new variations of the attacks in the IoT networks, to reduce the signature database, and to increase the detection effectiveness on the low-resource devices.

The following material is organized in the next manner: Sect. 2 reviews successful cases of applying modern nature-inspired methods in solving cyber security issues; Sect. 3 discusses methods for aligning sequences: global, local, and additive approaches; Sect. 4 presents the proposed coding and alignment technique; Sect. 5
presents our results of the experiments on detecting the DDoS attacks using a BoT-IoT dataset on the portative IoT Raspberry Pi4 platform; Sect. 6 presents a discussion of the results of applying the bio-inspired methods; Sect. 7 concludes our contribution.

2 The Related Works

Recently, IDS researchers and developers have been experimenting with mechanisms successfully used in other fields, such as artificial neural networks and genetic algorithms [19].

Existing solutions and proposals based on neural networks are described in [14]; however, these solutions have the following drawbacks: neural networks tend to overfit or diverge, and the behavior of the network may not always be unambiguously predictable, which increases the likelihood of false positives or missing events. Decisions on the application of bio-inspired methods for detecting network intrusions are much less [18]. Several successful examples are known:

1. IDS based on DNA using ASP module [23].

   The main goal of this project is to optimize the most resource-intensive part of network traffic analysis—matching nucleotide sequences to calculate the offset between them. In this work, a low-level Content Addressable Memory-based (CAM) module called ASP was developed. Associative String Processor (ASP) is capable of matching fixed-length patterns at a rate of one character per processor clock. Thanks to this module, sequence matching occurs at high speed.

2. Combination of IDS with a genetic algorithm [2].

   In this project, authors present a genetic approach with an improved initial population and selection operator, to efficiently detect various types of network intrusions. Genetic approach is used to optimize the search of attack scenarios in audit files, thanks to its good balance exploration and exploitation. Experimental results show an increase in intrusion detection efficiency in a reasonable processing time and a reduce the false positive rate.

3. IDS based on the theory of evolution [9].

   In this project, authors also present an intrusion detection system, by applying genetic algorithm to efficiently detect various types of network intrusions. This approach uses evolution theory to information evolution in order to filter the traffic data and thus reduce the complexity. To implement and measure the performance of developed system is used the KDD99 benchmark dataset and obtained reasonable detection rate.

4. DNA-based IDS using the Smith-Waterman algorithm [5].

   This project presents a host-based intrusion detection system, which encodes each activity in the system into a DNA sequence (signature). Using the Smith-Waterman
signature matching algorithm, a reference signature is searched for. This approach is flexible from the point of view of matching two sequences, since in such a system, there are adjustable parameters, each of which can affect the magnitude of the offset and, consequently, the choice of the reference signature. Such parameters are, for example, the value of the penalty for the mismatch of two characters, for the gap, which is put in place of the missing characters in the sequence, etc.

The problem of finding correspondences and differences between sequences has been widely studied, since this allows one to determine the presence of many mutations. The main methods of bioinformatics can be divided into the following groups:

- methods based on RNA and DNA;
- protein algorithms;
- algorithms of genomics and proteomics.

All of the above algorithms work with long strings that have a limited alphabet. In biology, it is said that if two proteins or two nucleotide sequences have great similarity, then they are homologues. For bioinformatics, such homologous sequences are important, since they have a common origin, a similar 3D structure and, to one degree or another, a similar aminoacid sequence. Thus, their alignment is used to establish the correspondence of individual regions of the sequences [24].

### 3 Sequence Alignment Methods

Sequence alignment is a procedure for comparing two or more sequences by searching for rows of individual elements or characteristic combinations of sequence elements that are located in the sequences being aligned in the same order. Alignment algorithms allow you to search the databases of sequences of certain genes or patterns and find phylogenetic relationships using multiple alignments [3]. Thus, an important task of gene matching is to determine the degree of sequence similarity of genetic material. The general task of matching sequences is finding the largest common subsequence, that is, finding a subsequence that is a subsequence of several sequences [25]. A subsequence can be obtained by removing zero or more (optionally adjacent) characters. Variants are possible when the compared sequences have different lengths. Then, the smaller sequence is complemented by breaks that are of a conditional nature and do not carry a functional load. An important component is the quality assessment and comparison. The length of the longest common subsequence is an intuitive measure of the similarity between two sequences. To increase the capabilities as a comparison tool, you can use the scoring function to rank the different alignments so that biologically similar alignments are scored higher. The scoring function assigns positive scores to aligned characters that either match or are known to be like. Negative scores are assigned to both aligned characters that differ and characters that are aligned with spaces. Typically, the alignment score is the sum of the scores for each aligned character pair. A graphical representation of aligned
Fig. 1 Example of sequence alignment

sequences is a record containing sequences, where the aligned characters are written underneath each other. Thus, one column contains amino acids that are similar or have the same functional load. An example of a graphical representation of aligned sequences is shown on Fig. 1.

There are two basic matching methods: global and local alignment.

3.1 Global Alignment

Global alignment looks for the best alignment of two sequences from start to finish. And best of all in situations where the sequences are sufficiently similar and approximately the same length.

An efficient dynamic programming algorithm for optimal sequence alignment was first introduced by Needleman and Wunsch [16]. The task of this algorithm is to find the maximum occurrence of a sequence in another. Let there be given two lines $A = a_1a_2\ldots a_m$ and $B = b_1b_2\ldots b_n$ where $m \leq n$. The algorithm searches for the optimal alignment of the sequence data. Matrix $F$ is used to search for alignment. The element of matrices $F_{i,j}$ takes the value of the weight of optimal alignment of subsequences $a_1a_2\ldots a_i$ and $b_1b_2\ldots b_j$. The matrix $F$ is filled recursively according to the following formulas:

$$F_{i,0} = -i \ast d, \quad F_{0,j} = -j \ast d,$$

$$F_{i,j} = \max \begin{cases} F_{i-1,j-1} + s(a_i, b_j), \\ F_{i-1,j} - d, \\ F_{i,j-1} - d. \end{cases}$$

where $s(a_i, b_j)$ is the value of the substitution matrix for elements of biological sequences; $d$ is penalty for breaking the sequence. If $F_{i,j} = F_{i-1,j-1} + s(a_i, b_j)$, then the elements $a_i$ and $b_j$ of sequences $A$ and $B$, respectively, are compared with each other. If $F_{i,j} = F_{i-1,j} - d$, then the element $b_j$ is removed. If $F_{i,j} = F_{i,j-1} - d$, then the element $a_i$ is removed.

The complexity of this algorithm is $O(mn)$. But most often $m$ and $n$ are close, so we can assume that $O(n^2)$. 
3.2 Local Alignment

Local alignment looks for similar areas within two sequences, which are best used if the sequences are of different lengths or are not similar enough. It is often important to identify more subtle types of similarities. The two strings may not be uniform over their entire length; they may contain smaller substrings that are similar. Moreover, global alignment requires lines to be approximately the same length. To account for this possibility, Smith et al. [22] proposed a modification of the Needleman and Wunsch alignment algorithm to compute local alignment. Given two lines $A = a_1 a_2 \ldots a_m$ and $B = b_1 b_2 \ldots b_n$ where $m \leq n$. Task: find substrings $\alpha$ and $\beta$ of rows $A$ and $B$, respectively, for which the alignment estimate is maximum for all possible pairs of substrings from $A$ and $B$. To find the optimal alignment, the matrix $F$ is filled according to the following formulas:

$$F_{i,0} = 0, F_{0,j} = 0,$$

$$F_{i,j} = \max \begin{cases} 0, \\ F_{i-1,j-1} + s(a_i, b_j), \\ F_{i-1,j} - d, \\ F_{i,j-1} - d \end{cases}.$$

It can be noted that the comparison of sequence elements is performed in the same way as in the Needleman-Wunsch algorithm, but when $F_{i,j} = 0$, a new alignment begins; that is, when the alignment estimate is negative, a new alignment of the subsequence is sought. This algorithm has complexity $O(n^2)$. Local alignment allows you to match a substring of a larger input string to a smaller string. Typically, the global alignment algorithm is unable to detect such similarities because most characters from a longer string must be space aligned, resulting in a negative score. A graphical representation of global and local alignment is shown in Fig. 2 [5].

**Fig. 2** Example for global and local alignment
3.3 The Alignment with Additive Matrices

Typically, the alignment score is the sum of the scores of each aligned symbol pair. The task of optimal sequence alignment is to find the highest score alignment for a given score function and a pair of rows. There are three main aspects to consider when scoring:

- the value assigned to the matching characters;
- value assigned to non-matching characters;
- value assigned to spaces.

It should be noted that when aligning the DNA sequence, it is always penalized the same; that is, when constructing the alignment, no additional information is taken into account. However, the alignment of amino acids, which are not all equally different from each other, exploits their evolutionary conservatism.

The structural features of the compared symbols are taken into account as follows: substitution matrices (Fig. 3) or amino acid similarity matrices (Fig. 4) are used. Amino acids with similar biochemical properties, such as charge, polarity, and others, are more likely to be paired. In order to consider the unequal probability of substitutions, these matrices were developed, which contain estimates of the partial weights for any pair of replacement of an amino acid (or nucleotide) \(i\) with an amino acid (or nucleotide) \(j\). The larger the number at the intersection of the matching letters, the greater the quantitative indicator of similarity.

![Fig. 3 Example of a similarity matrix](image)

![Fig. 4 Example of positional weight matrix](image)
Similarity matrix data may also be applicable for sequence analysis in cybersecurity. Indeed, many events in the system can be different, but legal and interchangeable, when equalized, one cannot strongly penalize for non-coincidence. To take into account, the similarity of mismatched elements in this work proposed to construct a matrix based on the frequency of occurrence. In bioinformatics, there is also the concept of positional weight matrix (PWM)—a method that is used to search for motives in biological sequences [4]. The weights reflect the frequency of the sequence element at a particular position.

Thus, using the PWM, it is possible to vary the values of the estimates for different elements of the sequences in pairwise alignment, multiplying the base estimates by the weights, that is, by the values from the matrix. Then, at the testing stage, where the actual alignment of the sequences of network traffic packets takes place, PWM constructed at the training stage is used. The size of PWM is $n \times m$, where $n$ is the cardinality of the set of unique elements from the training sample, and $m$ is the length of the sequences. To build a matrix need:

1. for each unique element $i \in [0; n]$ calculate the number of occurrences at each position $j \in [0; m]$ in all original data sequences;
2. write the obtained value into the corresponding cell $(i, j)$ of the matrix;
3. convert the values of the matrix by dividing by the total number of sequences in the alignment.

At this stage, the elements of the resulting matrix show the probability of meeting a specific element at a specific position in the original alignment. To get weights, you need to normalize the values.

Taking into account the matrix, the assessment of the alignment quality at the testing stage is calculated as follows:

- If the elements match, then add 1.
- If the element was skipped, then add 0.
- If the elements are different, then add the corresponding value from the substitution matrix, located at the intersection of the current compared elements.

## 4 The Proposed Bio-Inspired Method

The developed bio-inspired intrusion detection method based on the analysis of network traffic packets consists of two main stages: a training stage and a testing stage. Figure 5 shows the detailed diagram of our method, which uses the global alignment algorithm with a pre-built PWM.
4.1 Training Stage

The training stage is the training of the system, where a behavior profile (pattern) is created, with which the comparison will be made. A pattern is a set of sequences of labeled (normal or abnormal) training network traffic. The training pattern sequences are preliminarily presented in the same form as the test ones used in the analysis; that is, processing and converting are similar to the test mode described in Sects. 4.1.1 and 4.1.2. At this stage, a PWM is constructed based on the obtained pattern. The description of the construction of the matrix is presented below, in Sect. 4.1.3. The pattern and the constructed matrix are placed in the data storage of the information system. In addition, at this stage, a sign is determined; that is, a criterion (threshold) by which the input data is classified as normal or abnormal traffic. In this work, the threshold is determined on the basis of statistical data. For this purpose, the training data sequences are aligned with each other, and for example, the average value of the alignment quality assessment is taken.

4.1.1 Data Preprocessing

To carry out preprocessing of “raw” data, significant properties necessary for detecting an attack and reflecting the unique characteristics of an object are selected, and noise removal and normalization are performed. This work poses the problem of choosing meaningful fields in network packages, which are tuples formed into a set of a fixed size of the form:
sequence = packet1(field1, field2, ..., fieldn),
packet2(field1, field2, ..., fieldn)...

Each sequence is a matching unit for alignment. For a practical experiment, the original training set was transformed as follows:

- Selected dataset $X$ of normal traffic (labeled “normal”);
- From the obtained dataset, a dataset $X'$ is formed, which represents a set of structures with the following fields:

\[
data=(\text{part}[1], \quad \#\text{protocol_type}\text{part}[2], \quad \#\text{service}\text{part}[3], \\
\quad \#\text{flag}\text{part}[10] \#\text{num_failed_logins})
\]

- All possible types of packets were selected from the set $X'$, that is, combinations of fields, and thus, a matrix with types of packets $M$ was obtained, where each type of packet has its own serial number:

\[
(0, ("tcp," "telnet," "SF," "4"))
(1, ("tcp," "time," "RSTO," "0"))
(2, ("tcp," "ftp," "SF," "0"))
(3, ("tcp," "other," "S1," "0"))
(4, ("tcp," "smtp," "S1," "0"))
(5, ("tcp," "ftp," "RSTO," "0"))
(6, ("udp," "domain_u," "SF," "0"))
(7, ("tcp," "telnet," "S0," "0"))
(8, ("tcp," "finger," "RSTR," "0"))
(9, ("tcp," "ftp," "SF," "1"))
(10, ("tcp," "private," "REJ," "0"))
(11, ("tcp," "finger," "S3," "0"))
(12, ("tcp," "ftp_data," "S1," "0"))...

- From the dataset $X'$, a set of sequences of packets $S'$ was formed.
- In future, the resulting set will be used as a standard with which alignment should be performed when detecting attacks.

### 4.1.2 Encoding

To construct a PWM, the biopython was used. This module works with strings of characters $A, C, G, T$, which are sequences of amino acids. For this reason, it is necessary to convert the data obtained in the previous paragraph, for which the following encoding was used:

- From the matrix $M$, the ordinal number of each element, that is, each type of packet, is converted into a quaternion number system.
- Each digit of the obtained numbers is replaced by the corresponding amino acid symbol, that is, 0–to $A$, 1–to $C$, 2–to $G$, 3–to $T$. Thus, after converting the formed sequences of packages to a form suitable for use in Biopython, the following was obtained:
4.1.3 Building a PWM

At this stage, a PWM is built. The values of the PWM show the frequencies of occurrence of each symbol in each position.

"A": [0, 351, 1261, 62, 431, 928, 270, 433, 804, 392, 469, 674, 449, 491, 606, 504, 499, 589, 503, 497, 543, 564, 487, 550, 545, 496, 559, 551, 512, 529, 495, 520, 554, 507, 539, 526, 543, 561, 543, 504, 531, 559, 522, 513, 545, 532, 518, 549, 494, 517, 579, 511, 542, 543, 542, 533, 514, 528, 548, 507, 573, 521, 507, 511, 523, 546, 541, 500, 528, 539, 513, 506, 524]

The resulting values have been normalized to a range of −5 to 0:

"A": [-5.0, -3.6926605504587156, 5.0, -5.0, -3.8375165125495374, 5.0, -5.0, -4.294755877034358, 5.0, -5.0, -3.354114713216958, 4.9999999999999998, -5.0, -2.6504297994269344, 4.441340782122909, -3.643216080402012, -2.7737226277372273, 1.7622950819672134, -3.88111888118879, -2.908163265306123, -0.968508287292799, 1.25, -4.450549450549448, 0.097276254509143934, 0.15798473684210353, -2.5213675213675235, 0.794392523644888, 0.6372549019607865, ...]
Matrix values are in the range \([-5; 0]\), showing that the larger the number, the more likely it is to meet the given character at the given position. Consequently, during alignment, the higher the score, the larger the corresponding values in the matrix; that is, the sequence is more similar to the pattern. Thus, a PWM was obtained; the values of which are used as coefficients in evaluating the alignment of sequences.

### 4.2 Testing Stage

At the testing stage, the incoming test data is aligned with the earlier received:

- Sequences of normal and attacked traffic were obtained from the test suites, which were pre-processed.
- For each input test sequence, an alignment was carried out with the “reference,” and the score was calculated using a matrix. An example of equalizing normal and attacked traffic and evaluating it using a matrix:

```
CAGC-CT-C-C-CTGTGACGT-C-CGATCATGTG-AGT-ACTGACGTGAC-CTGAC-C-C-C-CG-TGTGACGTG-ACAGC-C-C-C-CA-
AG------------------------------------------
| ||| ||| | | || | | | | | | | | | |||| | || |||
| | |||| || | | | | | ||||| || C A -
CAC-ACACAC----AC--ACAC---ACA---CA--CACA-------CA--CACACACAC--
ACAC---ACACACAC-AC--A--CA--CA--CAC---ACA----CA--CACACACACACA-
CACACACACACACACACACACACACACACACACACACACACACACAC
"MyScore:", 100
```

For each comparison, a quantitative quality assessment is made, which is compared with the accepted threshold. The result goes to the response subsystem, which notifies the information system itself and counteracts the attack if necessary. The results are also fed into the database, which can be used to update it. The proposed bioinspired IDS architecture is shown in Fig. 6.

### 5 The Experimental Study and Results

The BoT-IoT dataset was used for training and testing [12]. It is based on the traffic of an emulated IoT network. This dataset contains 69 GB of traffic, which is divided into normal and infected portions. For the experiments, dataset records containing traffic with a DoS/DDoS attack and traffic without attack were selected.

The methods based on the global alignment: Needleman-Wunsch, Needleman-Wunsch with a PWM, and local alignment: Smith-Waterman and Mauve [7] techniques have been implemented and experimentally studied on the IoT Raspberry
Pi4 platform. The experimental setup had the following configuration: Raspberry Pi 4 Model B, Broadcom BCM2711 processor model, 4 cores, frequency 1500 MHz, LPDDR4 RAM 4096 MB, Raspbian operating system.

A comparison was made between two approaches to choosing a pattern: a white pattern (based on the normal traffic) and a black pattern (based on the abnormal traffic—with a DoS/DDoS attack).

The implementation was carried out using the following software tools and kits:

- Python3.6—an interpreter of the Python programming language;
- Biopython—a set of freely distributed tools for biological computing, written in Python by an international team of developers;
- NumPy—a software library written in Python, which provides a wide range of functions for working with various mathematical algorithms and processing of multidimensional datasets;
- scikit-learn—a software library written in Python that implements various machine learning algorithms.

Figures 7 and 8 show the ROC curves constructed in the course of experiments with white pattern and black pattern. Table 1 shows the accuracy, precision, recall for the algorithms under consideration when using the white pattern (WP) and black pattern (BP).

As it can be resulted out of the above graphics and the summary table, the best results have been obtained for the Needleman-Wunsch algorithm with the PWM with the black pattern. This can be explained by the fact that attack traffic is similar to each other, even despite some mutations. At the same time, pure traffic can differ quite strongly from each other, and the resulting averaged pattern may not take into account the features of individual secure packets, as a result of which the number of false positives will increase during the testing stage.
6 Discussion

Analysis of the existing related works devoted to the use of various genetic, nature-inspired, and artificial intelligence algorithms confirms the promising of the proposed bioinformatics approach. For example, in their work, the authors propose the use of chaotic variants of grasshopper optimization for solving protein folding optimization
### Table 1  Quality of traffic classification

| Alignment algorithm                  | Accuracy | Precision | Recall |
|--------------------------------------|----------|-----------|--------|
| Needleman–Wunsch with PWM (WP)       | 0.96     | 0.95      | 0.97   |
| Needleman–Wunsch (WP)                | 0.82     | 0.80      | 0.87   |
| Smith-Waterman (WP)                  | 0.85     | 0.85      | 0.88   |
| Mauve (WP)                           | 0.96     | 0.94      | 0.96   |
| Needleman–Wunsch with PWM (BP)       | 0.98     | 0.99      | 0.98   |
| Needleman–Wunsch (BP)                | 0.89     | 0.93      | 0.87   |
| Smith-Waterman (BP)                  | 0.86     | 0.88      | 0.85   |
| Mauve (BP)                           | 0.96     | 0.97      | 0.93   |

problem [21]. The genetic algorithm, differential evolution, particle swarm optimization, the flower pollination algorithm, and the plant propagation algorithms are widely used for different solutions [15]. A gene expression programming technique can be applied for CBR-based traffic and file transfer applications in the networks [20]. Our paper extends the wide range of the nature-inspired research studies.

Our work deals with the area of cyber security and intrusion detection, namely the classification of the network packets. As the result of the experiments, it has been found that the encoding of network packets into nucleotide chains occurs in a short time, and the use of parallelization allows avoiding time delays in the IDS, which is especially important for the highly mobile and low-resource IoT networks.

Much attention should be paid to the selection and calculation of the pattern and the similarity threshold. Composing several patterns (each type of attack has its own pattern) can improve the accuracy of the traffic classification, but this leads to a decrease in the speed of the IDS. The choice of the similarity threshold is a question about the number of the false positive and false negative. Depending on the type, task, and criticality of the IoT system, the threshold can be either overestimated or underestimated. In case of matching with the black pattern, an overestimation of the threshold will lead to the risk of missing the attack packet. An underestimate of the threshold will lead to the fact that normal packets will be dropped out of the IDS and will not reach the recipient.

### 7 Conclusion

The presented work discusses the task of detecting the variant types of the network intrusions in the IoT-based systems. The research study reviews the successful cases of applying the modern nature-inspired methods for solving the cyber security tasks. To detect the network intrusions in the IoT, it was proposed to apply the genetic sequence alignment calculations.
The developed methods based on the Needleman-Wunsch, Needleman-Wunsch with a PWM, Smith-Waterman, and Mauve techniques have been implemented and experimentally studied with the BoT-IoT dataset on the IoT Raspberry Pi4 platform. The best results for the classifying of the network traffic have been obtained with the global Needleman-Wunsch alignment accomplished with the PWM and the black pattern.

Our approach is adapted for the low-performance IoT devices, which compare favorably with existing analogues that use massive signature databases. The limitation of our technique is that the accuracy of the attack detection decreases with long chain length. But the real-life use case does not meet this limitation, because the network packet’s length is limited.

The advanced nature-inspired data processing technology is likely to play a leading role in sustainability of a newly volatile environment of the IoT, thereby decreasing the risk of undermining the secure, stable, and useful application of cyber space to facilitate the economic growth and individual lives.

Acknowledgments The research is financially supported by LG Electronics Inc.

References

1. E. Aleksandrova, Methods of group authentication for low-resource vehicle and flying self-organizing networks. Automat. Control Comput. Sci. 51(8), 947–958 (2017)
2. S. Benaicha, L. Saoudi, S. Guermache, O. Lounis, Intrusion Detection System Using Genetic Algorithm. in 2014 Science and Information Conference (London, 2014), pp. 564–568
3. H. Carrillo, D. Lipman, The multiple sequence alignment problem in biology. SIAM J. Appl. Mathe. 48(5), 1073–1082 (1988)
4. X. Chen et al., Learning position weight matrices from sequence and expression data. Comput. Syst. Bioinfo. 6, 249–260 (2007)
5. S. Coull, J. Branch, B. Szymański, E. Breimer, Intrusion Detection: A Bioinformatics Approach. in 19th Annual Computer Security Applications Conference (Las Vegas, NV, USA, 2003), pp. 24–33
6. A. Dakhnovich, D. Moskvin, D. Zegzhda, An approach for providing industrial control system sustainability in the age of digital transformation. IOP Conf. Series Mater. Sci. Eng. 497, 1–10 (2019)
7. A. Darling et al., Mauve: multiple alignment of conserved genomic sequence with rearrangements. Genome Res. 14, 1394–1403 (2004)
8. P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, W. Lee, Polymorphic Blending Attacks. in USENIX (2006), pp. 241–256.
9. M. Hoque, M. Mukit, M. Bikas, An implementation of intrusion detection system using genetic algorithm. Int. J. Netw. Sec. Appl. 4(2), 109–120 (2012)
10. Kaspersky Lab Report, IoT under fire: Kaspersky detects more than 100 million attacks on smart devices in H1 2019 (2019) URL: https://www.kaspersky.com/about/press-releases/2019_iot-under-fire-kaspersky-detects-more-than-100-million-attacks-on-smart-devices-in-h1-2019
11. Kaspersky Lab, DDoS attacks in Q1 2020. URL: https://securelist.ru/ddos-attacks-in-q1-2020/95949/
12. N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset (2018) https://arxiv.org/abs/1811.00701
13. D. Lavrova, M. Poltavtseva, A. Shtyrkina, *Security Analysis of Cyber-Physical Systems Network Infrastructure*, in 2018 IEEE Industrial Cyber-Physical Systems (ICPS) (St. Petersburg, 2008), pp. 818–823.

14. E. Malyshev, D. Moskvin, D. Zegzhda, Application of an artificial neural network for detection of attacks in VANETs. Autom. Control Comput. Sci. *53*(8), 889–894 (2019)

15. M. Mellal, A. Salhi, Parallel-Series System Optimization by Weighting Sum Methods and Nature-Inspired Computing, in *Applied Nature-Inspired Computing: Algorithms and Case Studies*. ed. by N. Dey, A. Ashour, S. Bhattacharyya (Springer Tracts in Nature-Inspired Computing, Springer, Singapore, 2020), pp. 231–251

16. S. Needleman, C. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. *48*(3), 443–453 (1970)

17. E. Pavlenko, D. Zegzhda, M. Poltavtseva, *Ensuring the Sustainability of Cyberphysical Systems Based on Dynamic Reconfiguration*. in 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS) (Taipei, Taiwan, 2019), pp. 785–789

18. A. Procopiou, N. Komninos, *Bio/Nature-inspired Algorithms in A.I. for Malicious Activity Detection*, in Nature-Inspired Cyber Security and Resiliency: Fundamentals, techniques and applications, ed. by E.-S. El-Alfy, M. Elroweissy, E.W. Fulp, W. Mazurczyk (2019)

19. A. Shenfield, D. Day, A. Ayesh, Intelligent intrusion detection systems using artificial neural networks. ICT Express *4*(2), 95–99 (2018)

20. J. Sangeetha et al., *Analyzing and Predicting the QoS of Traffic in WiMAX Network Using Gene Expression Programming*. in Advancements in Applied Metaheuristic Computing, ed. by N. Dey (IGI Global, 2018), pp. 34–70

21. A. Saxena, R. Kumar, Chaotic Variants of Grasshopper Optimization Algorithm and Their Application to Protein Structure Prediction, in *Applied Nature-Inspired Computing: Algorithms and Case Studies*. ed. by N. Dey, A. Ashour, S. Bhattacharyya (Springer Tracts in Nature-Inspired Computing, Springer, Singapore, 2020), pp. 151–175

22. T. Smith et al., Identification of common molecular subsequences. J. Mol. Biol. *147*(1), 195–197 (1981)

23. Z. Trabelsi, R. Hamdy, *Implementation of a Dna-Based Anomaly Identification System Utilizing Associative String Processor (ASP)*. in ACS/IEEE International Conference on Computer Systems and Applications—AICCSA (Hamamet, 2010) pp. 1–8

24. L. Volkert, D. Stoffer, *A Comparison Of Sequence Alignment Algorithms for Measuring Secondary Structure Similarity*. in 2004 Symposium on Computational Intelligence in Bioinformatics and Computational Biology (La Jolla, CA, USA, 2004), pp. 182–189

25. Y. Wang, H. Wu, Y. Cai, A benchmark study of sequence alignment methods for protein clustering. BMC Bioinfo. *19*, 529 (2018)

26. D. Zegzhda, D. Lavrova, E. Pavlenko, Management of a dynamic infrastructure of complex systems under conditions of directed cyber attacks. J. Comput. Syst. Sci. Int. *59*(3), 358–370 (2020)

27. P. Zegzhda, M. Poltavtseva, A. Pechenkin et al., A use case analysis of heterogeneous semistructured objects in information security problems. Autom. Control Comput. Sci. *52*(8), 918–930 (2018)