The Journal of Supercomputing (2022) 78:7016–7037
https://doi.org/10.1007/s11227-021-04133-4

# NKA: a pathogen dose-based natural killer cell algorithm and its application to classification

**Dongmei Wang**[1] · **Yiwen Liang**[1] · **Xinmin Yang**[2]

Accepted: 20 September 2021 / Published online: 4 November 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Inspired by the self/non-self discrimination theory, Forrest et al. proposed a distance-based negative selection algorithm (NSA). However, in the detector generation phase, NSA creates specific antibodies for each antigen based on the mechanism of specific antigen–antibody binding reactions, which lead to an excessive and redundant number of detectors. Moreover, during the detection phase, NSA needs to calculate the matching degree of the antigen with all the detectors, which has a high computational cost and low detection efficiency. This letter presents a novel natural killer cell algorithm (NKA) inspired by the mechanism of NK cells constructing phenotype detectors based on pathogen dose, which is a non-specific immune mechanism. NKA defines dose and phenotype detector, optimizes the detectors based on the memory evolution mechanism of phenotype, then establishes k-d tree for the optimized phenotype, and pathogens only need to match the dose with the nearest phenotype detector. Experimental results reveal that NKA can not only achieve the best performance through fewer detectors but also has a higher efficiency in the training and detection phase, compared with three versions of the distance-based NSA algorithms. Meanwhile, NKA generates comparable results compared with six popular machine learning algorithms.

**Keywords** Innate immune memory · NK cell algorithm · Artificial immune system · Classification · Negative selection algorithm

✉ Yiwen Liang
ywliang@whu.edu.cn

Dongmei Wang
dmwang@whu.edu.cn

[1] School of Computer Science, Wuhan University, Wuhan 430072, China

[2] School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

## 1 Introduction

AIS offers the potential for effective and efficient classification in many fields with interesting mechanisms and intelligence including self-learning, self-adaption, self-regulatory, distributed with self/non-self detection capabilities [2, 16, 21]. Among the various mechanisms in the immune system that are explored for AIS, the negative selection algorithm (NSA) is one of the most used and discussed models [5, 19, 20].

Negative selection algorithm is a classification algorithm generated by simulating the process of T cells to establish adaptive immune memory. It was first proposed by Forrest in 1994 [11]. The negative selection theory believes that the immune system can distinguish between self and non-self. The immune system does not respond to self, but produces an immune response to non-self. The negative selection algorithm inspired by this mechanism is divided into two phases—training phase and testing phase. The foothold of the training phase is that only mature T cells can build a detector based on adaptive immune memory, and the standard for T cell maturation is that it cannot recognize self-samples; the idea of the test phase is when the test data sample can be matched by the mature T cell detector, it is abnormal, otherwise it is normal.

NSA has proven to be effective in classifications including anomaly detection, fault diagnosis, computer security, financial domain, medical domain [7, 9, 33, 37, 41]. However, the original NSA randomly generates candidate detectors that produce numerous redundant detectors, and hardly cover the entire antibody space. In addition, the randomly generated candidate detectors must compare with all the self-sets; therefore, the inefficient generation of detectors seriously affects the application of NSA.

To overcome these defects, many people have made various improvements to the algorithm to improve the efficiency and expand the application of the algorithm, such as data representation, candidate detector generation and hole repair.

In the first decade of researching NSA, researchers primarily used binary or string to represent data, used r-contiguous bits, r-chunks, landscape-affinity matching, and used Hamming distance to match the similarity [17]. This type of data representation limited the application of NSA, mainly because it commanded high computational costs for generating efficient detectors. In 2002, Gonzalez proposed real NSA (RNSA) using distance matching among real-values instead of string matching [18]. RNSA rapidly became popular and started to be used in many areas, and it has been observed that it performs better than binary NS (BNSA) [19]. RNSA still has limitations, such as the difficulty of choosing appropriate matching rules and determining how to handle the dimensionality of the data [20, 26, 27, 42]. In fact, the Euclidean distance is still an important matching rule for RNSA at present.

In most early variations of the NSA, candidate detectors were generated randomly, which resulted in many detectors having duplicate coverage and poor coverage of the non-self space of detectors. Therefore, researchers tried many heuristics, such as using pseudo-randomness and adaptive techniques based on evolutionary computation to generate candidate detectors. In 2014, Idris et al.

[22] generated detectors based on local selective differential evolution in the random detector generation phase of the NSA; their subsequent work [23] introduced particle swarm optimization (PSO) into NSA, optimizing the generation of random detectors to achieve maximum coverage in a non-self space. In 2015, Xiao et al. [43] proposed an immune optimization-based real-valued NSA that introduced an immune optimization mechanism and generated candidate detectors from far to near in a self-centered layer. Chen et al. [28] generated an adaptive detector radius based on the self-radius distribution, randomly generating a few negative detectors at places far and near the self-set, which improved the efficiency of detector generation in 2017. Zhu et al. [47] proposed a Voronoi graph-based candidate detector generation method to improve the efficiency of detector generation. In 2018, Zhang et al. [46] used clonal selection algorithm (CSA) to generate initial immune cells within a specified range and executed the clonal selection algorithm at the same time. Chikh et al. [7] used k-means clustering and fruit fly optimization algorithm to optimize randomly generated detectors during the training phase in 2019. Yang et al. [44] proposed an antigen density clustering-based NSA to select non-self clusters and reduce the randomness of the detector generation. In 2021, He et al. [20] improved the negative selection algorithm by specifying candidate detector based on hierarchy division. Although most of the above-mentioned detector generation methods allow candidate detectors to be generated directionally rather than randomly, the problem of redundant detectors covering the antibody space still exists. In other words, these generated detectors are deficient in covering non-self regions in the most efficient way. This inadequacy results in the hole problem between the detector and the self-space due to uncovered non-self space, and these holes indicate classification errors.

In order to repair these holes and maximize detector coverage and efficiency, many researchers have focused on optimizing detector distribution and detector size determination. Meanwhile, multiple ensemble algorithms are combined into NSA for better optimizations; the notable methods are Genetic Algorithm (GA) and Fuzzy techniques in this era. In 2006, Gao [12] proposed GA-based NSA to optimize the detectors to occupy the maximal coverage of the non-self space. The limitation of this method is the high computational cost. In 2007, Bakicki [4] proposed a better GA-based NSA that introduced a rank based system with multi-objective GA, but it was application specific. Wang [40] used GA-based optimization approach combined with fuzzy logic, but this method also has limited applicability. Gao used clonal optimization [15] and particle optimization [13] in succession to determine the detector size. But both method suffered by higher computational cost. In 2008, Gao [14] proposed a Neural Network based approach to make NSA adaptive. Luo et al. [31] proposed a state graph supported NSA. In 2009, Zeng et al. [45] used an innovative technique to adjust the self-radius and evolve the non-self covering detectors to build an appropriate profile of the system. In 2011, Fakhari et al. [8] introduced a self-adaptive NSA to make detector size and position dynamic. Zhang et al. [46] generated big-radius and small-radius candidate detectors to make them gradually cover boundary space between selves and non-selves and reduce the number of holes. To sum up, the limitations of these methods are higher computational cost or limited applicability.

This paper proposes a new natural killer cell algorithm (NKA) based on pathogen dose inspired by the establishment process of the phenotype detectors of biological natural killer.

The main contributions of the present study are as follows:

1. By abstracting the biological mechanism of natural killer cells (NKs), we construct a novel artificial immune system algorithm - natural killer cell algorithm (NKA) for classification. The establishment of the phenotype detector of natural killer cells is not specific to a certain pathogen, but non-specific to the dose of all pathogens with certain characteristics in a certain dangerous zone. And, NKA establishes different NK cell phenotypes based on the characteristic dose of pathogens;
2. The evolution of phenotype detectors stage achieves the purpose of optimizing the detector through the cloning, tolerance, and extinction operations of the phenotype detector, and then establishes a k-d tree for the optimized phenotype detectors;
3. The specified phenotype detector only needs to be compared with core pathogen in the nearest dangerous zone instead of comparing it all the self-antigens in NSA, which improves the detector-generating efficiency of the NKA.

The rest of the paper is structured as follows. Section 2 introduces the abstraction of NK cell biological model and the framework proposed in this article. In Sect. 3, the experimental settings were described. The results and discussion are showed in Sect. 4. And the paper is concluded in Sect. 5.

## 2 Proposed work

In this section, we propose a NKA based on natural killer cells. Therefore, we first introduce and abstract the biological mechanism of natural killer cell. Then, the details of NKA are presented.

### 2.1 The biological mechanism of natural killer cells

There are three states of NK cells: resting NK cells, effector NK cells and memory NK cells. The discovery of the state of memory NK cells has attracted attention due to a paper on innate immune memory published by Netea et al. in Science in 2016 [34]. Studies have found that the innate immune system is not static against pathogens. The innate immune system also has a memory mechanism. It is named ‘Innate Immune Memory’ (IIM) or ‘Trained Immunity’ (TI). Innate immune memory refers to the fact that after the innate immune cells have returned to an inactive state, their secondary response to non-specific stimuli will still change in some way. The immune response is more or less enhanced than the initial response, thus giving The ability of innate immune cells to adjust response according to environment and time. Some innate immune cells such as NK cells and macrophages have been confirmed

to have innate immune memory. The existence of innate immune memory proves that the innate immune system can change its functions adaptively and dynamically according to the pathogens or products encountered before, and recognize pathogens faster and stronger. We abstracted the innate immune memory establishment process of NK cells, as shown in Fig. 1.

The phenotype, which is used by NK cells to identify and study functional subsets, has served as an accessible means for profiling NK cells in different states. When NK cells are in a resting state, their surface phenotypes are mainly $KLRG1^{lo\ and\ hi}$, $CD62L^{lo\ and\ int}$, etc. When resting NK cells are stimulated by a certain dose of pathogens, they will transform into effector NK cells. The phenotypes on the surface of effector NK cells are mainly $KLRG1^{hi}$, $CD62L^{int\ and\ hi}$ Wait. When the pathogen dose further increases, the effector NK cells will differentiate into memory NK cells and produce the function of innate immune memory. The main phenotypes on the surface of memory NK cells are $KLRG1^{hi}$, $CD62L^{lo}$, etc.

When the pathogen dose is very small, NK cells are in a resting state, and no immune response is produced at this time; when the pathogen dose is low, a part of resting NK cells differentiate into effector NK cells, and the phenotype of the effector NK cell surface will vary according to the characteristics and dose of the pathogen will be produced, including different states of the same phenotype and different types of phenotypes; when the pathogen dose further increases, that is, when the dose of a pathogen with certain (or some) characteristic attributes reaches a certain threshold, part of the effect NK cells will differentiate into memory NK cells, and the generation of memory NK cell phenotypes is initiated according to the characteristics and dose of the pathogen at this time. Memory NK cells with the phenotype against this pathogen invasion have a longer survival time than other NK

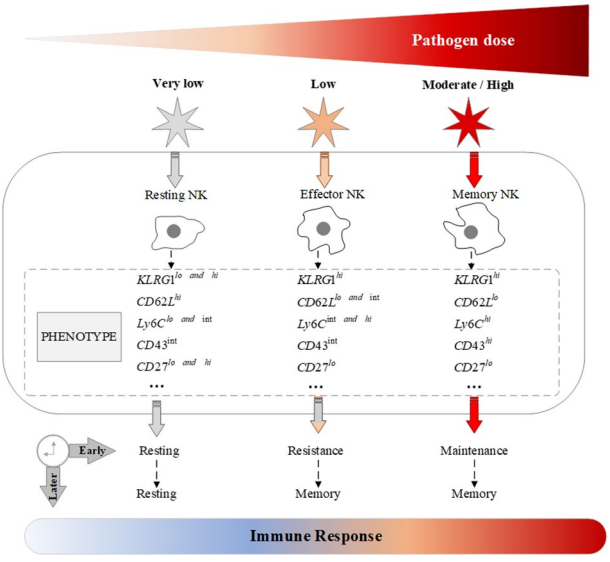



**Fig. 1** Innate immune memory establishment process of NK cells

cells, so when non-specific pathogens invade for a second time, only a lower dose of pathogens can activate memory NK with these phenotypes cells, the immune system will recognize pathogens with a faster and stronger immune response and protect the body.

The main function of the innate immune system is to recognize harmful pathogens and make corresponding immune responses; in addition, innate immune system is the basis for the development of adaptive immune system. After innate immune cells process and deal with pathogens, the characteristics of pathogens are transmitted to adaptive immune system through the process of antigen presentation. Then adaptive immune system produces specific antibodies to further enhance the ability to recognize and eliminate pathogens, and jointly protect the body with innate immune system. The innate immune system's natural abstraction, classification and recognition capabilities of pathogens provide a new idea for solving real-world classification problems. Therefore, this paper proposes a NK cell model based on innate immune memory and pathogen dose, then uses it for classification problems.

## 2.2 The framework of NKA

The proposed NKA algorithm inspired by natural killer cells consists of four stages: definition of the pathogen feature space, generation of phenotype detectors, evolution of phenotype detectors and detection.

### 2.2.1 Definition of the pathogen feature space

For real-valued data, the pathogen feature space is the real-valued vector of the $n$-dimensional space, denoted by $E = [0, 1]^n$, and $n$ is the dimension of the pathogen feature. The vector of features is represented as $V = (x_1, x_2, \ldots, x_n)$, and we use maximum and minimum method to normalize and scale each feature to [0, 1] interval.

### 2.2.2 Generation of phenotype detectors

Assuming that the sample set is $D = (x_1,\ x_2,\ ...,\ x_m)$, and $m$ is the number of samples in the training set. The phenotype detector is defined as:

(1) Dangerous zone: For $x_j \in D$, its dangerous zone is described as $\epsilon$-neighbors which contains the sub-sample set of the sample set $D$ whose distance from $x_j$ is not greater than $\epsilon$, that is, $N_\epsilon(x_j) = \{x_i \in D \mid distance(x_i, x_j) \leq \epsilon\}$. The number of samples in this dangerous zone is recorded as $\mid N_\epsilon(x_j) \mid$. We adopt Euclidean distance as the distance function.

(2) Core pathogen: For any sample $x_j \in D$, if the $N_\epsilon(x_j)$ corresponding to the dangerous zone contains at least $MinPts$ pathogens, that is, $\mid N_\epsilon(x_j) \mid \geq MinPts$, and the ratio of the number of abnormal pathogens to the total number of pathogens in the dangerous zone exceeds the set dose threshold $ratio$, that is, $count_{abnormals}/(count_{abnormals} + count_{normals}) \geq ratio$, then $x_j$ is the core pathogen, which is the phenotype detector we generate.

(3) Direct dose: If $x_i$ is located in the dangerous zone of $x_j$ and $x_j$ is the core pathogen, then the doses of $x_i$ and $x_j$ are directly reached, that is, if $x_j$ is an abnormal pathogen, then $x_i$ is also an abnormal pathogen. We will use this rule as a discriminator for the classification stage.

The process of phenotype detector generation is as follows: for each sample $x_j$ in training sample set $D$, determine its dangerous zone and the number of pathogens in it, then calculate the concentration occupied by the abnormal pathogens in that dangerous zone. If the concentration of abnormal pathogens is greater than the set threshold $\epsilon$, add $x_j$ to the phenotype detectors set $Detector_G$, otherwise discard it.

The pseudocode of the above steps is as shown in Algorithm 1.

**Algorithm 1** The pseudocode for phenotype detector generation and selection.

**Input:** $D$: training sample set; $m$: the sample number of $D$; $n$: data dimension; $\epsilon$: the dangerous zone size; $MinPts$: minimum number of pathogens in the dangerous zone; $ratio$: dose threshold
**Output:** Phenotype detector set $Detector_G$;

```
Initialization: | N_ε(x_j) |= 0, j ∈ [0, m]; D' = D[class =' abnormals']
for each ∀x_j ∈ D' do
    count_abnormals, count_normals = 0
    for each ∀x_i ∈ D do
        if distance(x_i, x_j) <= ε then
            | N_ε(x_j) | += 1
            if x_i.Class ==' abnormal' then
                count_abnormals += 1
            else
                count_normals += 1
            end if
        end if
        if | N_ε(x_j) |>= MinPts and count_abnormals/(count_abnormals + count_normals) ≥ ratio then
            x_j is a core pathogen, and add x_j to Detector_G
        end if
    end for
end for
return Detector_G
```

### 2.2.3 Evolution of phenotype detectors

The phenotype detectors generated from above stage are not guaranteed to be optimal for classification. Therefore, in this section, we propose a phenotype evolution method to optimize these phenotype detectors.

(1) Parameters description

We first list and describe the parameters that will be used in the evolution of phenotype detectors.

$P_{init}$: the initial phenotype detector pool, that is, $Detector_G$ from Algorithm 1.
$Memory_p$: the set of memory NK cell phenotype detectors.
$max_{iter}$: max of iteration.
$P_{candidate}$: the candidate phenotype detectors of $Memory_p$.
$P_r$: the best 30% candidates from population $P_{candidate}$.
$Clone_t$: a temporary pool that clone the best phenotype detectors in $P_r$.
$Clone_m$: a temporary pool that mutate the best phenotype detectors in $P_r$.

(2) The process of phenotype evolution

Inspired by natural evolution mechanism of NK cells' phenotype, we abstract the diagram of the phenotype evolution method, as shown in Fig. 2, in which the corresponding iteration steps are explained as follows.

1) Initialization
   Initialize $P_{init}$, $Memory_p$ and $max_{iter}$. $P_{init}$ is generated from stage 2, that is, the core pathogens set. $Memory_p$ is initially set to NULL. $max_{iter}$ is set to 500 in this paper. Initialize $P_{candidate}$, which is initially chosen 70% randomly from $P_{init}$.
2) Termination condition judgment
   If the iterative stop criteria is satisfied, go to Step 10. The termination condition is that the algorithm reaches the maximum number of iterations $max_{iter}$ or the increment of the affinity of $Memory_p$ is less than a given threshold $\gamma$ which is set to 0.01 in the experiment.
3) Phenotype-pathogen affinity evaluation
   Evaluate the affinity of all the individuals in population $P_{candidate}$. The affinity here represents the coverage of the phenotype detector to pathogens, as shown in Eq. 1.

$$Aff = \frac{Num_{ab}}{Num_{total}}, \tag{1}$$

   where $Num_{ab}$ is the number of abnormal samples of the phenotype detector, $Num_{total}$ is the total sample number of the phenotype detector.
4) The best candidates selection
   Select the best candidates ($P_r$) from population $P_{candidate}$ in terms of their $Aff$.
5) Clone-like operating

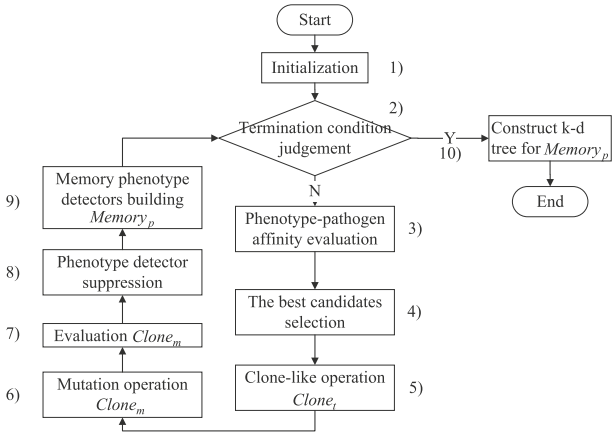



**Fig. 2** Diagram of the phenotype evolution method

Clone these best phenotype detectors as a temporary pool ($Clone_t$). The clone rate of each individual is equal to its affinity value, *Aff*.

6) Mutation operating

Generate a mutated phenotype detector pool ($Clone_m$). The mutation rate of each individual is inverse to its affinity, which is $1 - Aff$. Therefore, The better the affinity of the phenotype detector, the less likely it is to mutate.

7) Evaluation

Evaluate all the phenotype detectors in $Clone_m$.

8) Phenotype detectors suppression

Suppress and delete phenotype detectors similar to those in $Clone_t$, and update $Clone_m$. The similarity between two detectors is defined as the Euclidean distance between the centers of the core pathogens corresponding to each of the phenotypic detectors.

9) Memory phenotype detectors building

Re-select the phenotype detectors with better affinity from $Clone_m$ to build memory set $Memory_p$. Other improved phenotype detectors in $Clone_m$ can replace certain members with poor affinity in $P_{init}$ to maintain the phenotype diversity.

10) The k-d tree construction

In order to enable the pathogen $q$ to find its nearest neighbor phenotype detector more quickly, we construct the evolved phenotype detector in $Memory_p$ into the form of a k-d tree in a multi-dimensional space. The time complexity of the search is $O(log\ n + f)$, where $n$ is the total number of phenotype detectors, and $f$ is the number of leaf nodes.

The phenotype detector suppression in Step 9 is inspired by the building process of innate immune memory of NK cell. If the distance between the newly generated phenotype detector and the existing core pathogen in the candidate pool is less than $min_{dis}$, delete the phenotype detector to prevent too many phenotype detectors of the same type and maintain the diversity of phenotypes. $min_{dis}$ is set to 0.2 for Breast Cancer dataset, and 0.1 for KDDCUP99 dataset.

The pseudocode of the above steps is as shown in Algorithm 2.

**Algorithm 2** The pseudocode for phenotype detector generation and selection.

**Input:** $Detector_G$: Phenotype detector set
**Output:** a k-d tree for phenotype detector memory set $Memory_p$;
Initialization: $P_{init} = Detector_G$, $Memory_p = NULL$, $P_{candidate} = randomCopy(P_{init}, 70\%)$
**while** Termination condition judgement **do**
  **for** each $\forall p \in P_{candidate}$ **do**
    Calculate the number of abnormal samples $Num_{ab}$ and the total number of samples $Num_{total}$ covered by detector $p$
    $Aff.append(Num_{ab}/Num_{total})$
  **end for**
  Rank the $Aff$
  Select the best 30% candidates from $P_{candidate}$ according to the $Aff$ as $P_r$
  Clone $P_r$ as a temporary pool $Clone_t$
  Generate a mutated phenotype detector pool $Clone_m$
  **for** each $\forall m \in Clone_m$ **do**
    Calculate the number of abnormal samples $Num_{ab}$ and the total number of samples $Num_{total}$ covered by detector $m$
    $Aff_m.append(Num_{ab}/Num_{total})$
    **for** each $\forall t, q \in Clone_t$ **do**
      **if** $Euclidean(t, q) <= \epsilon$ **then**
        Remove $t$ from $Clone_t$
        Add $q$ to $Clone_m$
      **end if**
    **end for**
  **end for**
  Re-select the best 30% detectors from $Clone_m$ as $Memory_p$
  Replace poor detectors in $P_{init}$ with the improved detectors in $Clone_m$
  Construct a k-d tree for $Memory_p$
**end while**
**return** $Memory_p$, $k-d\ tree$

### 2.2.4 Detection

The detection stage can be described as whether a given pathogen $q$ can find the core pathogen in its $\epsilon$ neighborhood in the k-d tree. If it can be found, the pathogen $q$ is the same as the core pathogen, which is abnormal, otherwise is normal.

## 2.3 Analysis of time and space complexity

**Theorem 1** *The time complexity of NKA is* $O(m^2)$, *, where m is the sample number of training set D.*

***Proof*** The first stage of the algorithm is to normalize the pathogens, with time complexity $T_1 = O(n \cdot m)$, where $n$ is the dimension of pathogen feature space, $m$ is the sample number of training set $D$.

In the second stage, we calculated the pathogen dose in the dangerous zone for each abnormal sample and built a corresponding phenotype detector based on the abnormal sample dose. Thus, the time complexity is $T_2 = O(|D'| \cdot m)$.

In the third stage, to improve the affinity of these phenotype detectors, the clone-like and mutation operating are introduced. And the time complexity is $T_3 = O(0.7|Detector_G| \cdot 0.3|Detector_G|) \approx O(|Detector_G|^2)$.

In the fourth stage, we use k-d tree for searching the core pathogen and classify the given pathogen, in which the time complexity is $T_4 = O(log|Memory_p| + f)$, where f is the number of leaf nodes.

In summary, the time complexity of the algorithm is the sum of $T_1, T_2, T_3, T_4$, as shown in Eq. 2.

$$O(n \cdot m + |D'| \cdot m + |Detector_G|^2 + log|Memory_p| + f) \approx O(m^2) \tag{2}$$

, where $log|Memory_p| + f \ll m^2$. Therefore, the time complexity of NKA is $O(m^2)$. □

**Theorem 2** *The space complexity of NKA is O(m), , where m is the sample number of training set D.*

***Proof*** The space complexity of NKA depends on the number of detectors, and is linearly proportional to the size of training set, therefore, the space complexity of NKA is $O(m)$.

We can see that the generation and evolution process of phenotype detectors are time-consuming. A possible explanation for this might be that the samples (pathogens) in the training set (population) is disordered or randomly distributed, thus we can only compare with the surrounding pathogens one by one. And this is a common problem with NSA and its variations. Further research should be undertaken to investigate how to improve the efficiency of the generation phase. But the time complexity of detection stage ($log|Memory_p| + f$) is significantly lower than NSA and its variations ($O(|D|^2)$, where $|D|$ is the detector size). The experiments in Sect. 4 will further show the phase execution time comparison of these methods on different datasets.

The space complexity of NKA is consistent with NSA and its variations, which are $O(m)$. □

## 3 Experimental settings

In this section we describe some experimental settings, including the benchmark datasets, the comparing algorithms, the main parameters set of NKA and the criteria for performance evaluation.

### 3.1 Benchmark data analysis

The data used in this paper are carried on two classification datasets from UCI machine learning repository, namely Breast Cancer dataset and KDDCUP99 dataset, which are most common, widely acceptable and recognized datasets [24]. First, the UCI Wisconsin Breast cancer data set is used to validate the NKA and is a well understood two-class data set. The UCI data consists of 700 items, classified by their corresponding real-valued attributes. These data can be used to give preliminary figures of NKA performance in terms of accuracy and precision. Second, anomaly detection is one of the important research directions of AIS, and the KDDCUP99 dataset is one of the most used and most representative intrusion detection datasets.

### 3.2 Comparing algorithms

Although many variations of NSA have been proposed, the most significant of which include constant sized real-valued negative selection (RNS) [25], variable size V-detector [27] and an adaptive NSA-PSO [23]. They have shown excellent classification performance and have been successfully used in many fields. And, in this research, we focus more on comparing with the mature improved variations of NSA.

Moreover, the original and variations of NSA have an extensive competition with binary classification (BC). Most common approach for solving binary classification problem is support vector machine (SVM [39]). These binary classification techniques can be classified in 4 types. SVM, Logistic Regression (LR [36]) considered as linear model based BC techniques. Another type is proximity based which includes Local K-Nearest Neighbors (KNN [6]), decision tree (DT [38]), Bayesian network (BN [32]). Combining several methods of BC are known as ensemble techniques include Random Forest (RF [35]), Adaptive Boosting (AdaBoost [10]), Gradient Boosting Decision Tree (GBDT [30]). With the improvement of neural networks, BC problems were solved using different neural network models such as back-propagation neural network (BPNN [29]), Convolutional neural network (CNN [1]), Generative Adversarial Networks (GAN [3]). Therefore, we choose six of these algorithms, namely LR, DT, RF, SVM, BPNN and KNN, for further comparison.

### 3.3 Parameters set of NKA

There are three important parameters to be set in the second stage ('Generation of phenotype detectors') of NKA—$\epsilon$, *MinPts*, *ratio*.

A relatively easy parameter to set for NKA is the *minPts* parameter. It is intended for smoothing density estimates and for many datasets it can be set at the default value $minPts = 4$ (for two-dimensional data). For datasets with a lot of noise, very large, high dimensionality or with many repetitions, increasing *minPts* may improve the results. In this paper, *minPts* is initially set to 4 for both two datasets.

The dangerous zone parameter $\epsilon$ is often difficult to set. $\epsilon$ should be chosen to be as small as possible. The value of $\epsilon$ also depends on the distance function. In this paper, we choose the $\epsilon$ parameter based on the distance to the $(2dim1)$ nearest neighbor, where *dim* is the dimensions of the dataset. And for Breast Cancer dataset, $\epsilon$ is set to 0.24, while $\epsilon$ is set to 0.11 for KDDCUP99 dataset.

The parameter *ratio* is aim to control the abnormal dose of the phenotype detector. Generally, it is accepted as a phenotype detector when the proportion of abnormal samples reaches 80%. We set the *ratio* to 0.8 in our experiments.

The parameters in evolution process have been described in Sect. 2.

### 3.4 Criteria for performance evaluation

To evaluate the accuracy and performance of the proposed model for classification and compare it with RNS (NSA using a fixed detector radius) [18], V-detector (NSA using a variable detector radius) [25] and NSA-PSO (NSA combining NSA and PSO) [23]. And, all methods use the expected coverage($c_0$) as the termination condition in experiments, and $c_0$ is set to 99.9% according to [42] to achieve a better detection rate. The experiments' evaluation measures include accuracy ($A$), recall rate ($R$), specificity ($S$), precision ($P$), false alarm rate ($FAR$), Matthews correlation coefficient ($MCC$), and f-measure ($F1$). Moreover, the number of mature detectors ($ND$), training time ($T_{train}$) and testing time ($T_{test}$) are often used. The calculations for these measures are as follows:

$$A = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}) \tag{3}$$

$$R = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{4}$$

$$S = \mathrm{TN}/(\mathrm{TN} + \mathrm{FP}) \tag{5}$$

$$P = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \tag{6}$$

$$\mathrm{FAR} = \mathrm{FN}/(\mathrm{FN} + \mathrm{TN}) \tag{7}$$

$$\begin{aligned}\mathrm{MCC} = &(\mathrm{TP} * \mathrm{TN} - \mathrm{FP} * \mathrm{FN})/((\mathrm{TP} + \mathrm{FP}) \\ &* (\mathrm{TP} + \mathrm{FN}) * (\mathrm{TN} + \mathrm{FP}) * (\mathrm{TN} + \mathrm{FN}))^{1/2}\end{aligned} \tag{8}$$

$$F_1 = \frac{2PR}{P + R} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{9}$$

## 4 Results and discussion

To validate our algorithm in real-world applications, this section presents the experimental results on WBC and KDDCUP99 datasets. Table 1 shows the description of datasets. All methods were evaluated by dividing the dataset using a stratified sample approach with 75% training set and 25% testing set. The experiments were

**Table 1** Description of two standard UCI datasets

| Datasets | Records | Dimensions | Normal number | Abnormal number | Training set | Test set |
|---|---|---|---|---|---|---|
| WBC | 699 | 9 | 458 | 241 | 556 | 143 |
| 10% KDDCUP99 | 5000 | 41 | 2770 | 2230 | 3750 | 1250 |

repeated 100 times and the average performances were taken. In all tables, the number in bold represents the best result.

### 4.1 WBC dataset

The WBC dataset contains 699 records of nine attributes. Among them, 458 instances are benign (normal), and 241 instances are malignant (abnormal). In this work, we stochastically select 75% (556 instances) of them for training the phenotype detectors. And, we use Principal Component Analysis(PCA) with Min Max Normalization to reduce dimensions of attributes for visualization. To avoid the effect of randomly selected training sample data on the experimental results, each experiment is repeated 100 times independently.

Table 2 shows the experimental results for RNS, V-detector, NSA-PSO, and the proposed model NKA on WBC dataset. And, Fig. 3 shows the detector distribution of RNS, V-detector, NSA-PSO, and the proposed model NKA.

From Table 2 we can see that, RNS performs unsatisfying results in WBC dataset because RNS randomly generates candidate detectors and fixes the radius of the detectors. Therefore, it is difficult for the detectors to cover the whole antibody space and more detectors need to be generated to achieve the desired coverage, so the accuracy is as low as 73.42%, the recall rate (or detection rate) is as low as 59.09%, the false alarm rate is as high as 40.45%, and the detector number is 765. NSA-PSO uses PSO to improve the random detector generation of RNS. After PSO optimization, the performance has been significantly improved, with a 23.82% increase in accuracy, a 61.18% increase in recall rate and an 81.68% decrease in false alarm rate. Meanwhile, the number of detectors has also decreased. But due to the introduction of the PSO optimization algorithm, there is a significant increase in training time and testing time compared with RNS. V-detector uses the variable radius of detectors, which significantly improves the accuracy which is 24.78% and 0.77% higher than that of typical RNS and NSA-PSO but with the most detectors number of 8374. While V-detector generates the most detectors, and trained and tested in the longest time. It is worth noting that V-detector achieved the lowest FAR of 0.0, the best recall rate of 1.0, which means the positive samples predicted is totally correct. NKA generates phenotype detectors by pathogen dose of dangerous zone, and pathogens only need to be compared with the nearest neighbor phenotype detector via k-d tree. Thus, no matter the classification performance or the generation and detection efficiency of

**Table 2** Experimental results for RNS, V-detector, NSA-PSO, and NKA on Breast Cancer dataset($c_0$ =99.9%)

| | $A$ | $R$ | $S$ | $P$ | FAR | MCC | $F_1$ | ND | $T_{train}$(s) | $T_{test}$(s) |
|---|---|---|---|---|---|---|---|---|---|---|
| RNS | 73.42 | 59.09 | **96.36** | 96.30 | 40.45 | 55.65 | 73.38 | 765 | 17.61 | 18.34 |
| NSA-PSO | 90.91 | 95.24 | 84.75 | 89.89 | 7.41 | 81.22 | 90.84 | 563 | 35.43 | 36.14 |
| V-detector | 91.61 | **100.0** | 77.36 | 88.24 | **0.0** | 82.62 | 91.33 | 8374 | 57.48 | 59.42 |
| NKA | **96.50** | 97.85 | 94.00 | **96.81** | 4.08 | **92.29** | **96.50** | **7** | **0.39** | **0.04** |

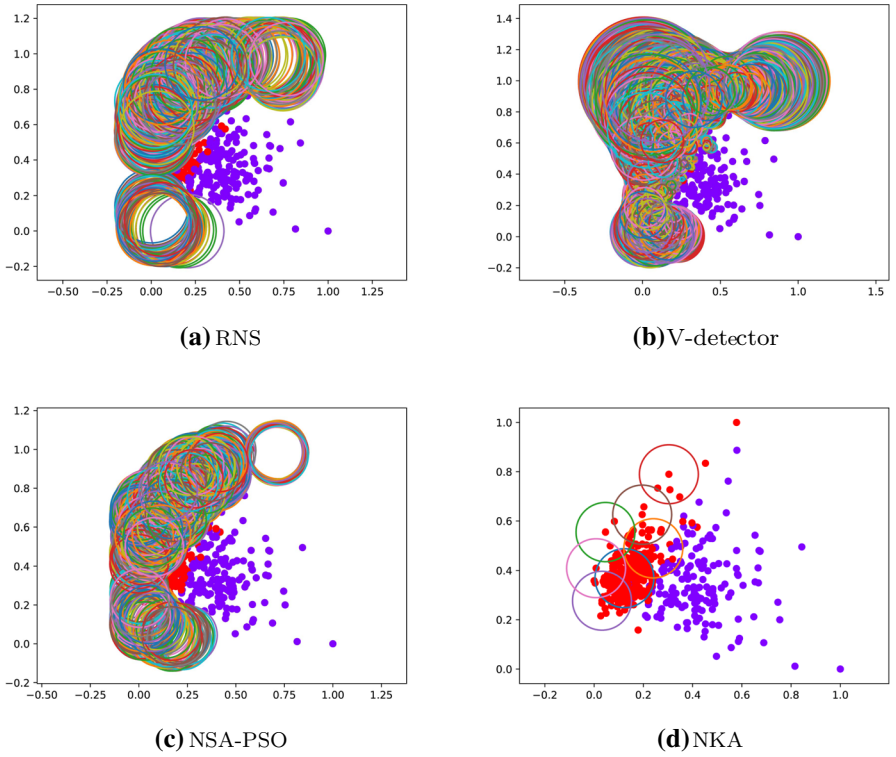



**Fig. 3** Detector distribution of RNS, V-detector, NSA-PSO, and NKA on Breast Cancer training dataset

phenotype detectors are obviously better than other three algorithms except the recall rate and specificity value of V-detector. Compared to V-detector, NKA only generated about 0.16% of the detectors, trained in 1.77% the training time and tested in 1.72% the testing time. From Fig. 3, we can see more clearly the number and distribution of detectors generated by four algorithms. NKA generated the fewest detectors with a better coverage of abnormal samples, while V-detector generated an excessive number of redundant detectors. The RNS and NSA-PSO do not cover abnormal antigens well because of their fixed detector radius, so although their number of detectors is high, the results are less satisfactory. Overall, on the Breast Cancer Wisconsin dataset, our proposed NKA achieved better performance than RNS, V-Detector and NSA-PSO, especially in the detector number, training and testing complexity.

In addition, Table 3 shows the results of six popular machine learning algorithms. From the chart, it can be seen that NKA obtains the best performance on the accuracy of 96.50%, which is 5.79% higher than that of the lowest KNN. NKA also shows competitive results with the other six algorithms in four metrics: precision, specificity, recall rate, and $F_1$ score. From a joint analysis of five metrics value (the average value of five metrics value), NKA achieves the best result,

**Table 3** Experimental results of popular machine learning algorithms on Breast Cancer dataset

| | $A$ | $P$ | $R$ | $S$ | $F_1$ | $T_{train}$(s) | $T_{test}$(s) |
|---|---|---|---|---|---|---|---|
| LR | 95.91 | **97.39** | 96.55 | **94.55** | 96.97 | **0.01** | 0.003 |
| DT | 92.39 | 93.27 | 95.68 | 85.45 | 94.46 | 0.02 | **0.002** |
| RF | 93.56 | 94.11 | 96.55 | 87.27 | 95.31 | 0.17 | 0.01 |
| SVM | 95.32 | 97.36 | 95.68 | 94.54 | 96.52 | 0.05 | 0.004 |
| BPNN | 96.49 | 96.61 | **98.27** | 92.72 | **97.43** | 0.32 | 0.003 |
| KNN | 91.22 | 96.33 | 90.51 | 92.72 | 93.33 | 0.03 | 0.008 |
| **NKA** | **96.50** | 96.81 | 97.85 | 94.00 | 96.50 | 0.39 | 0.04 |

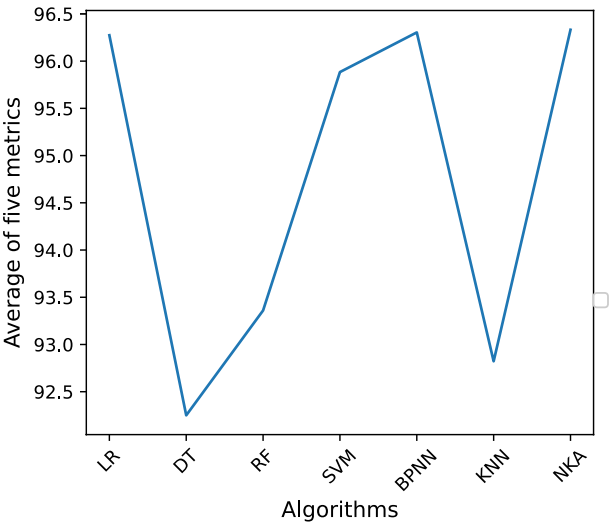



**Fig. 4** Average value of {A, P, R, S, $F_1$} of algorithms on Breast Cancer dataset. And NKA achieves the best score

as shown in Fig. 4. Nevertheless, compared with these more maturely developed machine learning algorithms, NKA does not have an advantage in time complexity due to its more complex detectors generation and evolution process. In fact, how to reduce the time complexity of AIS algorithms remains an open challenge to AIS practitioners. In the future, AIS algorithms can be benefited by the recent improvement in the distributed big data system.

### 4.2 KDDCUP99 dataset

KDDCUP99 dataset is the most common and widely used anomaly detection dataset. The complete dataset consists of more than 5 million records; each of them represents a TCP/IP connection composed of 41 features. This paper uses KDDCUP99 Mini (10% of original KDDCUP99) as the experimental dataset which contains 5000 records.

**Table 4** Experimental results for RNS, V-detector, NSA-PSO, and NKA on KDDCUP99 dataset ($c_0$ =99.9%)

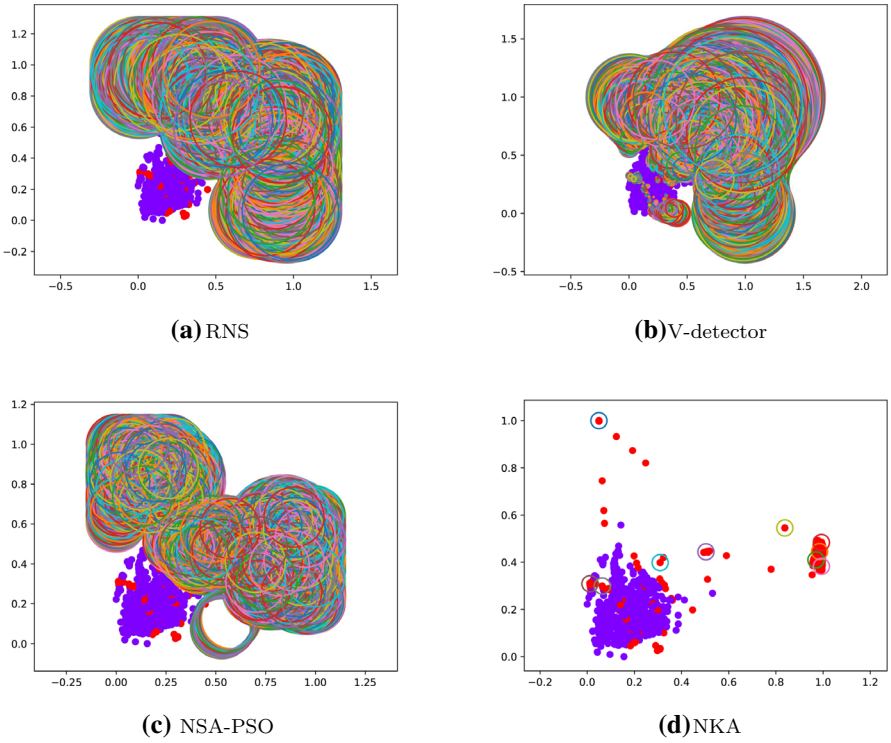
| | $A$ | $R$ | $S$ | $P$ | FAR | MCC | $F_1$ | ND | $T_{train}$(s) | $T_{test}$(s) |
|---|---|---|---|---|---|---|---|---|---|---|
| RNS | 92.88 | 95.42 | 90.76 | 89.59 | 4.03 | 85.87 | 92.89 | 17,176 | 1095.00 | 1108.04 |
| NSA-PSO | 96.96 | 95.34 | **99.00** | 98.36 | 4.27 | 93.87 | 96.95 | 12,892 | 995.40 | 546.20 |
| V-detector | 96.48 | **97.56** | 95.68 | 94.36 | **1.86** | 92.87 | 96.49 | 147,737 | 4611.78 | 5211.37 |
| NKA | **97.80** | 96.80 | 98.65 | **98.38** | 4.96 | **94.33** | **97.58** | **10** | **31.71** | **0.64** |

In the Table 4, we present the experimental results for RNS, V-detector, NSA-PSO, and the proposed model NKA on KDDCUP99 dataset. The V-detector was consistent in its performance, where RNS and NSA-PSO did better on this dataset. One of the main reasons for this is that bio-heuristic algorithms require more data for evolution and optimization. The experimental results in Table 4 show that our proposed NKA achieves a better performance than that of other three distance-based methods of AIS. It is worth noting that, NKA only generated 10 detectors and achieved the best accuracy, sensitivity, precision, et al. With the increase of data volume and dimensionality, the training and testing time of these four algorithms has increased significantly. What stands out in the table is the shortest training time and testing time of NKA. Compared to V-detector, NKA only generated about 0.08‰ of the detectors, trained in 0.34% the training time and tested in 0.12‰ the testing time. Moreover, we can see that the detector number of NKA did not increase with the increase in the dimensionality of the data set, and this feature can be further explored. And, Fig. 5 shows the detector distribution of RNS, V-detector, NSA-PSO, and NKA. NKA only generates detectors in dangerous zone with high dose of abnormal samples, hence the number of detectors is minimum.

The results with six classic machine learning algorithms are shown in Table 5. From this data, we can see that NKA resulted in the highest value of accuracy, recall rate and $F_1$. It is worth noting that SVM generates the best precision of 100% and sensitivity of 100%, but the recall rate has dropped a lot, to 89.68%. Therefore, from the view of the average value of accuracy, precision, recall rate, specificity and $F_1$, it can be concluded that NKA is a relatively more suitable classifier for this dataset in terms of the evaluation metrics. However, the time complexity of NKA is much higher than these machine learning algorithms on high-dimensional dataset. And this is one of the shortcomings of immune-heuristic algorithms and one of the directions of our improvement afterward.

### 4.3 Discussion

The results obtained from the proposed NKA model is compared with the three variations of NSA model and other machine learning algorithms in this research. The differences in the performances between the proposed NKA and the NSA variations are very significant. The best accuracy of dose-based NKA is 96.5% on WBC dataset and 97.80% on KDDCUP99 dataset, while for RNS, NSA-PSO and NKA, they are (73.42%, 92.88%), (90.91%, 96.96%) and (91.61%, 96.48%). The optimization



**(a)** RNS **(b)** V-detector **(c)** NSA-PSO **(d)** NKA

**Fig. 5** Detector distribution of RNS, V-detector, NSA-PSO, and the proposed model NKA on KDD-CUP99 dataset

**Table 5** Experimental results of popular machine learning algorithms on KDDCUP99 dataset

| | $A$ | $P$ | $R$ | $S$ | $F_1$ | $T_{train}$(s) | $T_{test}$(s) |
|---|---|---|---|---|---|---|---|
| LR | 95.73 | 99.68 | 90.99 | 99.75 | 95.14 | **0.04** | 0.02 |
| DT | 96.67 | 97.05 | 95.64 | 97.54 | 96.34 | **0.04** | **0.01** |
| RF | 97.53 | 99.24 | 95.35 | 99.38 | 97.26 | 0.46 | 0.03 |
| SVM | 95.27 | **100.00** | 89.68 | **100.00** | 94.56 | 0.14 | 0.02 |
| BPNN | 95.73 | 99.68 | 90.99 | 99.75 | 95.14 | 0.59 | **0.01** |
| KNN | 96.93 | 99.54 | 93.75 | 99.63 | 96.56 | 0.06 | 0.03 |
| **NKA** | **97.80** | 98.38 | **96.80** | 98.65 | **97.58** | 31.71 | 0.64 |

of the time complexity of NKA in the training and testing phases is also evident, with decreases of 99.78%, 99.89%, and 99.93% compared to the three variations of NSA on the WBC dataset; and 99.94%, 99.88%, and 99.99% on the KDDCUP99 dataset, respectively. In general, the proposed NKA model outperforms these three NSA variations. Moreover, compared with six state-of-the-art machine learning algorithms, NKA also performs comparable results. The NKA outperforms LR, DT, RF, SVM, BPNN and KNN, with the highest accuracy of 96.50% and 97.80% on the

two datasets, respectively. These findings show that the accuracy of the NKA is better than the existing models using state-of-the-art machine learning tools.

If we now turn to the mechanism of NKA, the advantages of the NKA in theory are concluded as follows:

(1) The establishment of the NK cell model is non-specific to pathogens and has a fast generation rate. The non-specific pathogen matching mechanism can increase the recognition probability of homologous variant pathogens and improve the coverage and detection rate of the detector to pathogens;
(2) The classic NSA will delete all the detectors that can detect self-samples, which resulting in the failure to establish effective detectors in areas where self and non-self overlap, and the detection accuracy is extremely low in these areas. Figure 6 is a partial enlargement of Fig. 3a, which shows more clearly the detector distribution of classic NSA in the overlapping area, and we can see that NSA has hardly built any effective detectors in these borderline areas. In fact, a detector should be created to the right of the yellow circle in the lower left corner. Although there is a purple sample mixed in with the red samples, this purple sample should be ignored to achieve a better detection rate. In contrast, the dose-based NKA only deletes the detectors whose abnormal sample dose does not reach the threshold. Thus, NKA can generate effective detectors in the overlapping area ignoring the few noisy samples, as shown in Figs. 3d and 5d, or we can say that NKA has the ability to handle noise data, which refers to a point

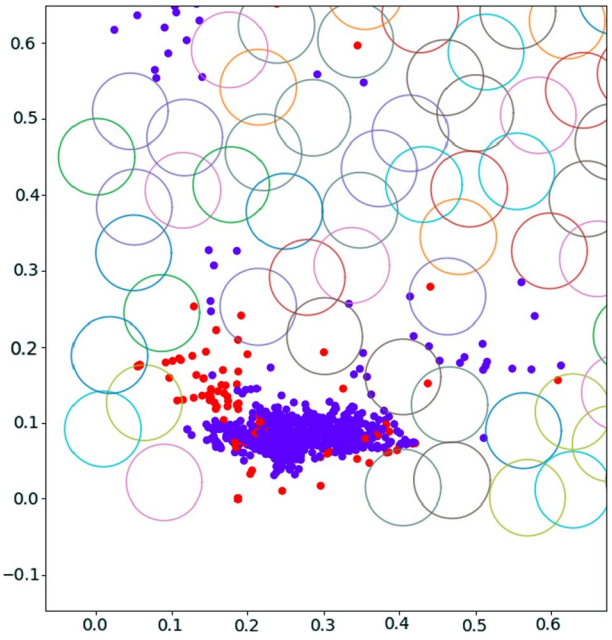



**Fig. 6** Detector distribution of classic NSA in the overlapping area of self and non-self samples, which is a partial enlargement of Fig. 3a

that is neither the core pathogen nor the dose boundary of core pathogen. This may be one of the reasons that NKA's detection rate is better than NSA-based algorithms.

(3) What's more, the introduction of k-d tree greatly improves the detection efficiency of NKA.

However, the shortcoming of NKA is also apparent, i.e., the time complexity is still high compared to these advanced machine learning methods and needs to be improved, especially on high-dimensional datasets. One of the key reasons is that the phenotype evolution process is relatively time-consuming. But compared with NSA variations, which are also in the AIS, the time complexity of the proposed algorithm is slightly lower.

## 5 Conclusions

Aiming at the problem that distance-based NSA cannot establish an effective detector in normal and abnormal mixed areas, a natural killer algorithm (NKA) based on pathogen dose is proposed in this paper. A theoretical analysis and experimental results show that NKA has a better accuracy, detector generation quality, training and testing complexity compared to three versions of distance-based NSA. Yet, the time complexity of NKA is much higher than the state-of-the-art machine learning algorithms. Moreover, this method is not applicable to very large data now, and when there are too many dimensions, the computational cost can be high. In the future, we will utilize a distributed system to be better time-efficient. Meanwhile, our future work will mainly focus on more in-depth theoretical analysis, improvement of its time complexity and applying NKA to more different situations to explore more possibilities of this method, such as intrusion detection, anomaly detection, fault diagnosis.

**Acknowledgements** National Natural Science Foundation of China (No. 61877045).

## References

1. Albawi S, Mohammed TA, Al-Zawi S (2018) Understanding of a convolutional neural network. In: Proceedings of 2017 international conference on engineering and technology, ICET 2017, vol 2018, pp 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186
2. Aldhaheri S, Alghazzawi D, Cheng L, Barnawi A, Alzahrani BA (2020) Artificial immune systems approaches to secure the internet of things: a systematic review of the literature and recommendations for future research. SCI2 J Netw Comput Appl 157:102537
3. Alqahtani H, Kavakli-Thorne M, Kumar G (2021) Applications of generative adversarial networks (GANs): an updated review. Arch Comput Methods Eng 28(2):525–552. https://doi.org/10.1007/s11831-019-09388-y
4. Balachandran S, Dasgupta D, Nino F, Garrett D (2007) A framework for evolving multi-shaped detectors in negative selection. In: 2007 IEEE symposium on foundations of computational intelligence, pp 401–408. https://doi.org/10.1109/FOCI.2007.371503
5. Chen J, Wang X, Su M, Lin X (2021) A fast detector generation algorithm for negative selection. Appl Intell 51(7):4525–4547. https://doi.org/10.1007/s10489-020-02001-x

6. Chen Y, Hu X, Fan W, Shen L, Zhang Z, Liu X, Du J, Li H, Chen Y, Li H (2020) Fast density peak clustering for large scale data based on kNN. Knowl Based Syst 187:104824. https://doi.org/10.1016/j.knosys.2019.06.032
7. Chikh R, Chikhi S (2019) Clustered negative selection algorithm and fruit fly optimization for email spam detection. J Ambient Intell Humaniz Comput 10(1):143–152. https://doi.org/10.1007/s12652-017-0621-2
8. Fakhari SNS, Moghadam AME (2011) NSSAC: negative selection-based self adaptive classifier. In: INISTA 2011—2011 international symposium on innovations in intelligent systems and applications, pp 29–33. https://doi.org/10.1109/INISTA.2011.5946064
9. Farzadnia E, Shirazi H, Nowroozi A (2021) A novel sophisticated hybrid method for intrusion detection using the artificial immune system. J Inf Secur Appl 58(February):102721. https://doi.org/10.1016/j.jisa.2020.102721
10. Feng DC, Liu ZT, Wang XD, Chen Y, Chang JQ, Wei DF, Jiang ZM (2020) Machine learning-based compressive strength prediction for concrete: an adaptive boosting approach. Constr Build Mater 230:117000. https://doi.org/10.1016/j.conbuildmat.2019.117000
11. Forrest S, Perelson AS, Allen L, Cherukuri R (1994) Self-nonself discrimination in a computer. In: IEEE computer society symposium on research in security and privacy, pp 202–212. https://doi.org/10.1109/RISP.1994.296580. arXiv:1212.5701
12. Gao XZ, Ovaska SJ, Wang X (2006) Genetic algorithms-based detector generation in negative selection algorithm. In: 2006 IEEE mountain workshop on adaptive and learning systems, SMCals 2006, pp 133–137. https://doi.org/10.1109/SMCALS.2006.250704
13. Gao XZ, Ovaska SJ, Wang X (2007) Particle swarm optimization of detectors in negative selection algorithm. In: Conference proceedings—IEEE international conference on systems, man and cybernetics, pp 1236–1242. https://doi.org/10.1109/ICSMC.2007.4413731
14. Gao XZ, Ovaska SJ, Wang X, Chow MY (2008) A neural networks-based negative selection algorithm in fault diagnosis. Neural Comput Appl 17(1):91–98. https://doi.org/10.1007/s00521-007-0092-z
15. Gao XZ, Ovaska SJ, Wang X, Chow MY (2009) Clonal optimization-based negative selection algorithm with applications in motor fault detection. Neural Comput Appl 18(7):719–729. https://doi.org/10.1007/s00521-009-0276-9
16. Goel L (2020) An extensive review of computational intelligence-based optimization algorithms: trends and applications. Soft Comput
17. Gonazalez F, Dasgupta D, Gomez J (2003) The effect of binary matching rules in negative selection. In: Genetic and evolutionary computation conference, pp 195–206. Springer, Berlin. https://doi.org/10.1007/3-540-45105-6_90
18. González FA, Dasgupta D (2003) Anomaly detection using real-valued negative selection. Genet Program Evolvable Mach 4(4):383–403. https://doi.org/10.1023/A:1026195112518
19. Gupta KD, Dasgupta D (2021) Negative selection algorithm research and applications in the last decade: a review. arXiv:2105.06109
20. He J, Chen W, Li T, Li B, Zhu Y, Huang M (2021) HD-NSA: a real-valued negative selection algorithm based on hierarchy division. Appl Soft Comput 112:107726. https://doi.org/10.1016/j.asoc.2021.107726
21. Hofmeyr SA, Forrest S (2000) Architecture for an artificial immune system. Evol Comput
22. Idris I, Selamat A, Omatu S (2014) Hybrid email spam detection model with negative selection algorithm and differential evolution. Eng Appl Artif Intell 28:97–110. https://doi.org/10.1016/j.engappai.2013.12.001
23. Idris I, Selamat A, Thanh Nguyen N, Omatu S, Krejcar O, Kuca K, Penhaker M (2015) A combined negative selection algorithm-particle swarm optimization for an email spam detection system. Eng Appl Artif Intell 39:33–44. https://doi.org/10.1016/j.engappai.2014.11.001
24. Irvine U (2021) UCI machine learning repository. [EB/OL]. https://archive.ics.uci.edu/ml/datasets.php. Accessed March 30
25. Ji Z, Dasgupta D (2004) Real-valued negative selection algorithm with variable-sized detectors. In: GECCO, p 55. http://eprints.uanl.mx/5481/1/1020149995.PDF
26. Ji Z, Dasgupta D (2006) Applicability issues of the real-valued negative selection algorithms. In: GECCO 2006—genetic and evolutionary computation conference, vol 1, pp 111–118. https://doi.org/10.1145/1143997.1144017
27. Ji Z, Dasgupta D (2009) V-detector: an efficient negative selection algorithm with "probably adequate" detector coverage. Inf Sci 179(10):1390–1406. https://doi.org/10.1016/j.ins.2008.12.015

28. Jinyin C, Mengmeng S, Haibin Z (2017) A novel radius adaptive hybrid detector generation algorithm. Optik 142:621–643. https://doi.org/10.1016/j.ijleo.2017.06.034
29. Karsoliya S (2012) Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture. Int J Eng Trends Technol 3(6):714–717
30. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY (2017) LightGBM: a highly efficient gradient boosting decision tree. In: Advances in neural information processing systems 2017-December (Nips), pp 3147–3155
31. Luo W, Wang X, Wang X (2007) A novel fast negative selection algorithm enhanced by state graphs. In: International conference on artificial immune systems. Springer, Berlin, , pp 168–181. https://doi.org/10.1007/978-3-540-73922-7_15
32. Marcot BG, Penman TD (2019) Advances in Bayesian network modelling: integration of modelling technologies. Environ Model Softw 111(2018):386–393. https://doi.org/10.1016/j.envsoft.2018.09.016
33. Mohapatra S, Khilar PM (2020) Fault diagnosis in wireless sensor network using negative selection algorithm and support vector machine. Comput Intell 36(3):1374–1393. https://doi.org/10.1111/coin.12380
34. Netea MG, Joosten LA, Latz E, Mills KH, Natoli G, Stunnenberg HG, O'Neill LA, Xavier RJ (2016) Trained immunity: a program of innate immune memory in health and disease. Science 352(6284):427. https://doi.org/10.1126/science.aaf1098
35. Probst P, Wright MN, Boulesteix AL (2019) Hyperparameters and tuning strategies for random forest. Wiley Interdiscip Rev Data Min Knowl Discov 9(3):1–15. https://doi.org/10.1002/widm.1301
36. Ranganathan P, Pramesh C, Aggarwal R (2017) Common pitfalls in statistical analysis: measures of agreement. Perspect Clin Res 8(4):187–191. https://doi.org/10.4103/picr.PICR_123_17
37. Rashid N, Iqbal J, Mahmood F, Abid A, Khan US, Tiwana MI (2018) Artificial immune system-negative selection classification algorithm (NSCA) for four class electroencephalogram (EEG) signals. Front Hum Neurosci. https://doi.org/10.3389/fnhum.2018.00439
38. Song YY, Lu Y (2015) Decision tree methods: applications for classification and prediction. Shanghai Arch Psychiatry 27(2):130–135. https://doi.org/10.11919/j.issn.1002-0829.215044
39. Tsang IW, Kwok JT, Cheung PM (2005) Core vector machines: fast SVM training on very large data sets. J Mach Learn Res 6:363–392
40. Wang H, Wang KJ, Yu LJ, Li XL (2007) Immune negative selection algorithm with an adjustable threshold based on fuzzy logic. Harbin Gongcheng Daxue Xuebao/J Harbin Eng Univ 28(11):1222–1227
41. Wang W, Ren L, Chen L, Ding Y (2019) Intrusion detection and security calculation in industrial cloud storage based on an improved dynamic immune algorithm. Inf Sci 501:543–557. https://doi.org/10.1016/j.ins.2018.06.072
42. Wen C, Tao L (2017) Parameter analysis of negative selection algorithm. Inf Sci 420:218–234. https://doi.org/10.1016/j.ins.2017.08.062
43. Xiao X, Li T, Zhang R (2015) An immune optimization based real-valued negative selection algorithm. Appl Intell 42(2):289–302. https://doi.org/10.1007/s10489-014-0599-9
44. Yang C, Jia L, Chen BQ, Wen HY (2020) Negative selection algorithm based on antigen density clustering. IEEE Access 8:44967–44975. https://doi.org/10.1109/ACCESS.2020.2976875
45. Zeng J, Liu X, Li T, Liu C, Peng L, Sun F (2009) A self-adaptive negative selection algorithm used for anomaly detection. Prog Nat Sci 19(2):261–266. https://doi.org/10.1016/j.pnsc.2008.06.008
46. Zhang R, Xiao X (2018) A clone selection based real-valued negative selection algorithm. Complexity. https://doi.org/10.1155/2018/2520940
47. Zhu F, Chen W, Yang H, Li T, Yang T, Zhang F (2017) A quick negative selection algorithm for one-class classification in big data era. Math Probl Eng. https://doi.org/10.1155/2017/3956415


**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.