A Revised Dendritic Cell Algorithm Using K-Means Clustering

Noe Elisa and Longzhi Yang
Department of Computer and Information Sciences,
Northumbria University,
Newcastle upon Tyne, UK
{noe.nnko, longzhi.yang}@northumbria.ac.uk

Yanpeng Qu
Information Science,
and Technology College,
Dalian Maritime University, China
yanpengqu@dlmu.edu.cn

Fei Chao
Cognitive Science Department,
Xiamen University,
China
fchao@xmu.edu.cn

***Abstract*—The most daunting and challenging task in intrusion detection is to distinguishing between normal and malicious traffics effectively. In order to complete such a task, the biological danger theory has appeared to be one of the most appealing immunological models which has been converted to a computer science algorithm, named as Dendritic Cell Algorithm (DCA). To perform a binary classification, the DCA goes through four phases, preprocessing, detection, context assessment and classification. In particular, the context assessment phase is performed by comparing the signal concentration values between mature (i.e., abnormality) and semi-mature (i.e., normality) contexts. The conventional DCA requires a crisp separation between semi-mature and mature cumulative context values. This can be hard if the difference between the two contexts is marginal, which negatively affects the classification accuracy. In addition, it is technically difficult to quantify the actual meaning of semi-mature and mature in the DCA. This paper proposes an approach that integrates the K-Means clustering algorithm to the DCA to map the DCA cumulative semi-mature and mature context values into semi-mature (normal) and mature (anomaly) clusters in order to improve the classification accuracy. The KDD99 data set was utilized in this work for system validation and evaluation, and the experimental results revealed an improvement in the classification accuracy by the proposed approach.**

***Index Terms*—Dendritic cell algorithm, Danger theory, K-Means clustering, Artificial immune systems**

I. Introduction

Network security has become one of the fundamental and essential requirements for information communication systems and their applications in smart cities and e-government [1] etc, especially due to the plethora of new cyber-attack techniques. Recently, many private and public organizations have lost their assets and finances as a result of network attacks, such as the denial of service [2]. The security measure of computer networks can be designed based on the biological immune systems (BISes), as it is straightforward to associate cyber-attacks with foreign molecules (pathogens) and the computer network with the mammalian body. Over hundreds of centuries, BISes have appeared to be strong and robust in protecting the human body against foreign molecules such as virus and bacteria by developing itself with great characters of adaptability, lightweight and autonomy. Since 1990s, a number of AIS algorithms were developed by mimicking the characteristics of BISes, which have been utilised to build intelligent intrusion detection systems that can protect computer networks against cyber-attacks.

According to the biological danger theory (DT) [3], the BISes concern the alarming signals that might cause damage to tissues and cells rather than foreigners. The human dendritic cells (DCs) collect, process and reveal these alarming signals to the immune system for response [4]. These signals are in the form of antigens, each of which is a foreign molecule that can activate immune response. Inspired by the DT theory and the antigens presentation behaviour of DCs [5], an artificial immune system, named as Dendritic Cell Algorithm (DCA), has been developed. The DCA has been applied to detect denial of service attacks in computer networks and smart grid with promising results [6], [7]. To detect anomaly, the DCA processes each network connection (represented as an antigen) in four processing phases, including preprocessing and initialization, detection, context assessment, and classification.

DCA context assessment phase is performed by comparing the signal concentration values between mature (abnormality) and semi-mature (normality) contexts. If the semi-mature concentration is greater than mature one, the antigen data will be mapped to the semi-mature class (normal); otherwise, the antigen data will be assigned to the mature class (anomalous). The most controversial question about DCA context assessment is the existence of a crisp separation between normality (semi-mature) and abnormality (mature) cumulative output values [8]. If the difference between the mature and semi-mature contexts is neglectable, the context of the antigen data collected in a DC will be hard to be separated. Any tiny noise could change and affect the context assessment decision which may pose a negative effect on the classification accuracy [8], which is particularly the case when the classes of data instances change over time. Also, it is very difficult to quantify the actual meaning of semi-mature and mature in the DCA compared to its counterpart in the conventional BISes.

This paper proposes a modified DCA algorithm which maps the DCs cumulative semi-mature and mature context values into semi-mature (normal) and mature (anomaly) clusters in order to address the aforementioned challenge. This is achieved by applying the K-Means clustering algorithm to the

This study was sponsored by the Commonwealth Scholarship Commission in the UK (CSC) and Northumbria University at Newcastle, UK.

DCA algorithm. Of course, any other clustering algorithms such as the mean-shift clustering [9] and the EM clustering [10] may also be applied here, but the K-Means is chosen because the number of clusters are known and it works well with large data sets and requires less computational effort compared to other clustering techniques [11]. Briefly, the DCA first preprocesses and initialises the signals for each antigen (data item), then computes cumulative semi-mature and mature context values and passes them to the K-Means clustering algorithm. From this, the K-Means algorithm assesses the context and clusters the data items to their classes.

The remainder of this paper is organized as follows: Section II provides the underpinning theoretical background in artificial immune system, danger theory and dendritic cell algorithm. Section III presents the proposed approach, which integrates the K-Means clustering algorithm with the conventional DCA algorithm. Section IV reports the experimentation that was performed to validate and evaluate the proposed approach by result analysis and comparison. Section V draws the conclusion of the study and points out the future research directions.

II. Artificial Immune System

AIS is a special class of computational intelligence approaches inspired by the mammalian immune system, which was developed for the purpose of classification, anomaly detection, and optimisation. The first immunological model ever exploited for AIS algorithms is known as the self-nonself model [12]. Self-nonself supposes that, the mammalian immune system can distinguish between self-cell, which is tolerated, and non-self (foreign), which is attacker and not tolerable. This is achieved through a process known as negative selection. In BIS, a negative selection process [12] occurs when new born immature T cells go through a negative selection process in the thymus to eliminate the self-reactive T cells binding with self-proteins. Therefore, the mature T-cells can only bind to nonself antigens when released to the blood circle.

Typical AIS algorithms based on the self-nonself model include Negative selection, positive selection and clonal selection algorithms [13]. Negative selection algorithm collects a set of self strings that define the normal state of the monitored system, and then generates a set of detectors that only recognize nonself strings. This detector set is used to monitor the anomaly changes of the traffics in the system in order to classify them as being self or non-self. Positive selection algorithm is an alternative to negative selection in which the detectors for self strings are evolved rather than for non-self.

Self-nonself AIS algorithm have been criticised and found to have weaknesses such as scalability, requiring initial learning phase, and large number of false positives amongst other [13]. To overcome these limitations, a new family of AIS algorithms based on the DT was introduced by [14], which is reviewed in the following subsections.

A. Danger theory

According to [3], the recognition of a foreign molecule such as virus is based on environmental context (signals) rather than the simple self-nonself discrimination behaviour. It doesn't matter if the damage is due to pathogens or by own cell defect, the immune response reacts against what is causing damage. If a foreign molecule does not cause damage or the cells die normal programmed death, immune tolerance is initiated [3]. The immune system is divided into two parts namely, the innate immune (e.g., skin) and adaptive immune system (e.g., white blood cells). The innate immune system is in-born immunity system and an important subsystem of the overall immune system that comprises the cells (e.g., DCS) and mechanisms that defend the host from infection by other organisms. The adaptive immune system is a subsystem of the overall immune system that is composed of highly specialized, systemic cells and the processes that eliminate pathogens or prevent their growth.

DCs reside in the innate immune system with the responsibilities of sampling and processing antigens and presenting them on the cell surface of the T-cells in the adaptive immune system [4]. Nevertheless, DCs express costimulatory molecules (CSMs) on their cell surfaces which limits the amount of antigens they can sample while in the affected tissue. The four signals that DCs collect from their neighborhood are pathogenic associated molecular patterns (PAMPs), safe signals (SSs), danger signals (DSs), and inflammatory cytokines (CKs).

PAMPs are proteins produced by pathogenic molecules such as virus and bacteria which can be easily detected by DCs and activate immune response. The presence of a PAMP signal expressed by an antigen indicates an anomalous situation. DSs are produced as a result of abnormal cell death. The presence of DSs indicate an anomalous situation but with lower confidence than PAMP signals. SSs are produced as a result of programmed cell death. The presence of SSs indicate that, DCs where collected in their normal conditions. CKs indicate that, a great number of DCs were collected in the tissue under distress but not affected.

Additionally, DCs exist in three states namely "immature", "semi-mature" and "mature", which determine exactly the properties of the collected antigens or data items.

- **Immature DC (iDC)**- iDCs are found in tissues in their pure state. In their immature state, iDCs collect signals and antigens which can either be, PAMP, DS or SS. The relative proportions of these signals causes iDCs to differentiate to a mature state or to a semi-mature state.
- **Mature DC**- an iDC become a mature DC (mDC) when the iDC is exposed to a greater quantity of either PAMPs or DSs than SSs. Sufficient exposure to PAMPs and DSs can cause maturation, the DC ceases antigen sampling and migrates from the tissue to the lymph node for antigen presentation and immune response.
- **Semi-mature DCs**- an iDC differentiate to a semi-

mature DC (smDC) as a result of exposure to more SSs than PAMPs and DSs. Antigens collected in a smDC cause immune tolerance and no immune reaction is initialised in such situation.

B. Dendritic Cell Algorithm

The DCA acquires the knowledge of normal and anomalous data using statistical analysis through categorization of input features into PAMP, DS and SS [15], and thus the DCA is a supervised learning approach. For anomaly detection and classification purposes, the DCA creates a population of DCs to form a pool from which a number of DCs are selected to perform data item sampling regarding signals (PAMP, DS and SS). While in the pool, DCs are exposed to the current signal values and the corresponding data items from the data source. Each DC has an ability to sample multiple data items. During the classification stage, an aggregated sampling value from different DCs for a particular data item is computed which is used to classify an antigen as normal or anomalous.

The DCA algorithm is outlined in Algorithm 1. The DCA algorithm takes data items as inputs, and produces the classification results as system outputs. The DCA goes through the following four phases to perform a classification task:

1) Preprocessing and Initialization
   At this phase, features selection and signal categorization are performed by selecting the most relevant features (attributes) from the input training dataset and assigning each selected attribute to a signal categories of either PAMP, DS or SS.
   - **PAMP:** An attribute indicates clearly the presence of anomalous behaviour associated with a given data item. For instance an attribute reflecting the number of error messages generated per second by a failed network connection. Those attributes which show a signature of a certain abnormal behaviors are mapped as PAMP signal.
   - **DS:** An attribute indicates the presence of abnormal behaviour but with lower confidence than PAMP. Increase in DS value increase the anomalous confidence value associated with a given antigen, though represent normal behavior at a low signal strength. For instance, an attribute reflecting the number of transmitted network packets per second.
   - **SS:** Presence of SS associated with an attribute is an indicator of normal behavior associated with a given antigen. For instance, an attribute referring to the inverse rate of change of number of network packets per second.

   In some studies on DCA, expert knowledge on the problem domain were used to select the most significant features and map them into their appropriate signal categories [5]. Other techniques have also been employed for feature selection, such as rough fuzzy logic [16], information gain, correlation coefficient. Dimensionality reduction techniques such as the principal component analysis, may also be applied. At this stage, two empty pools of DCs are initialized for immature DCs (P_i) and migrated DCs (P_m).
2) Detection
   The DCA processes the input signals to obtain three cumulative signals for CSM, $smDC$, mDC. The CSM is used to limit the amount of data items that a DC can sample, whilst $smDC$ and mDC are used to determine the context (normal/anomalous) of the data items. the cumulative signal values are computed using the following equation:

$$C = \frac{(W_{PAMP} * C_{PAMP}) + (W_{SS} * C_{SS}) + (W_{DS} * C_{DS})}{W_{PAMP} + W_{SS} + W_{DS}}, \quad (1)$$

Algorithm 1 DCA

input: the dataset D, the DC pool size n, sampling ratio s, migration threshold θ, anomaly-threshold th
output: normal or anomalous for data items
/** Preprocessing & Initialization phase**/
Initialise immature DC pool P_i with n DC cells;
Initialise migration DC pool P_m with unlimited size;
signal categorisation;
/** Detection phase**/
for each d in D **do**
  calculate the concentrations of CSM, mDC and $smDC$
  for 1 to s **do**
    randomly select a DC from P_i
    associate d with DC
    if cumulative $CSM > \theta$ **then**
      migrate DC
      select new DC
    end if
  end for
end for
/*Context Assessment phase */
for each DC in P_m **do**
  if smDC $\leq$ mDC **then**
    DC-context=1;
  else
    DC-context=0;
  end if
end for
/* Classification phase */
for each d **do**
  if DC-context == 1 **then**
    mature++;
  end if
end for
for each d **do**
  Calculate_MCAV();
  if MCAV $> th$ **then**
    Anomalous;
  end if
end for

where C_{PAMP}, C_{SS} and C_{DS} are the input signal values of PAMP, SS and DS signals of each data item, respectively, and W_{PAMP}, W_{SS} and W_{DS} are signal concentration weights used for PAMP, SS and DS signals, respectively. These weights are either predefined or derived empirically from the data.

The three cumulative output values are summed up overtime for all data items sampled by a DC. Therefore, each sampling DC from the pool is assigned a migration threshold θ in order to limit the amount of data items a DC can take. If the CSM value of a DC exceeds the threshold θ, the DC will be removed from the pool and replaced by a new one; in the same time, the removed DC will be migrated to the migration pool (P_m) for data items presentation and classification.

3) Context Assessment

The cumulative values of $smDC$ and mDC contexts are used to perform context assessment in this phase. If data items collected by a DC have a greater mDC than $smDC$ values, it is assigned a binary value of 1 and 0 otherwise. This information is then used in the classification phase to compute the number of anomalous data items present in the data set. To alleviate the crisp separation that exists between mDc and $smDC$, [17] proposed a technique known as Fuzzy Classification Dendritic Cell Method (FCDCM) based on Gustafson-Kessel algorithm which clusters and defines the contexts regarding the cumulative mDC and $smDC$ values rather than simply comparing them. By using the Euclidian distance, each DC context is given a vector of membership measures with different strength for each cluster so that it can belong to all cluster simultaneously. When the clustering is complete, the fuzzy cluster is converted into crisp one by assigning the point with the highest value of the membership function to represent the cluster. The classification accuracy results indicates that, the Gustafson-Kessel algorithm is appropriate to smooth the crisp separation between mDC and $smDC$ although its performance heavily depends on the applied feature selection technique [18]. This approach has been further extended by applying fuzzy rough set [17] for automatic feature selection and signal categorization.

4) Classification

All the collected antigens are analysed by deriving the Mature Context Antigen Value (MCAV) for each data item. Thus the MCAV is used to assess the degree of anomaly of a given data item. Firstly, the anomaly threshold of MCAV is derived from the testing data set based on the total number of anomaly items and the total number of items in the data set. Then, the MCAV value is calculated by dividing the number of times an antigen presented in the mature context to the total number of presences. Antigens with greater MCAVs than the anomaly threshold are classified into the anomalous class whilst the others are classified as normal.

III. The proposed Approach

The proposed approach is outlined in Figure 1. The system firstly performs feature selection, and only the most relevant features are retained in the training data set. The selected features are categorised into three input signals namely PAMP, DS and SS followed by a normalization process. To be modeled as antigens, data items with the selected features are assigned IDs which are used for their identification by the DCs. Data item IDs and the input signals are taken by the DCA and GA. Then GA searches for the optimal set of weights to be used in Equation 1 [19], which are then relayed to the DCA to compute the CSM, mDC and $smDC$ context values.

To classify each of the input data item either as normal, expressed as a semi-mature DC context, or anomalous, expressed as a mature DC context, the DCA forwards the $smDC$ and mDC values to the K-Means algorithm for clustering. The K-Means algorithm is initialized with two cluster centroids for semi-mature and mature, each with two initial centroid values corresponding to $smDC$ and mDC cumulative context respectively. The K-Means computes the minimum Euclidean distance between the DC context values ($smDC$,mDC) and cluster's centroids, assigns the DCs to their closest clusters, and then updates the cluster centroids. This process is iterated for multiple times until the centroids settled. After the clustering is performed, all normal data items will be assigned to the semi-mature cluster while all anomalous data items will be assigned to the mature cluster. Each components of the proposed modified DCA is detailed in the following subsections.

A. Feature Selection

The information gain [20] is used in this work to perform feature selection. The information gain of each attribute is calculated and attributes with substantial lower information gains are discarded. The information gain of an attribute indicates the amount of information the attribute provides with respect to the classification [21]. In particular, the approaches presented in [20] is used in this work. Given a sample data set S, the information gain of an attribute A, denoted as $G(S, A)$ can be calculated as:

$$G(S, A) = E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} * E(S_v), \qquad (2)$$

where $Values(A)$ represent the set of possible values that attribute A may take, S_v is a subset of S for A each of which takes value v for attribute A (i.e., S_v=$\{d \in S | A(d)=v\}$), and $E(S)$ is the entropy which is defined as:.

$$E(S) = \sum_{i=1}^{i=2} -p_i * log_2 p_i, \qquad (3)$$

where p_i is the proportion of elements in class i in reference to the total number of elements in the data set S. Obviously,

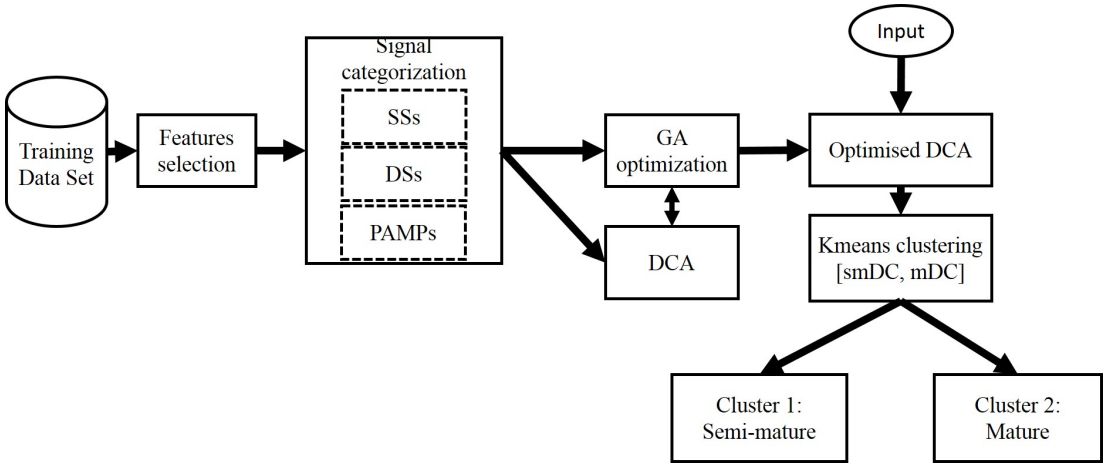

Figure 1: The modified DCA algorithm

there are only two classes as the DCA is a binary classification approach.

During signal categorization, the selected features are analysed using their histograms with respect to the two class labels (normal and anomaly) presented in the input data set. The frequency of occurrence of the largest values present in each attribute from each class was used to decide its signal category. If the largest values of an attribute has a high frequency of occurrence in the normal class than that in the anomalous class, the attribute will be categorized to SS signal, and to PAMP and DS otherwise.

B. DCA Weight Optimisation by GA

The weights used in Equation 1 are optimised by using the general optimisation AI approach GA [19]. Briefly, GA uses techniques inspired by evolutionary biology such as selection, mutation, crossover, reproduction and elitism for search and optimization problems [22]. GA evolution starts with a population of randomly generated individuals, each being an array of nine random integer numbers. The nine integers are divided into three groups of CSM, $smDC$ and mDC with each group having three values corresponding to PAMP, DS and DS signals. Secondly, the DCA takes in data items and the three signal categories (SS, DS and PAMP) from the training data set. Thirdly, DCA uses a weighted sum function given by Equation 1 supported by the weights generated from the GA to determine the contexts ($smDC$ and mDC) of the data items sampled using its normal data items and signals processing mechanism. The GA fitness function in this work is defined as the classification accuracy. The GA iterates until the maximum number of generation is reached or the error is smaller than a pre-defined threshold. When the GA terminates, the individual with the optimal accuracy from the final generation are used as the optimal weights, which are taken by the DCA to compute $smDC$ and mDC context values before the clustering process by the K-Means algorithm.

C. Context Assessment by K-Means Algorithm

The K-Means is a clustering method aiming to group the data points into k clusters by finding a centroid position for each cluster that minimizes the distance from the data points to the cluster's centroid [11]. Given k as the number of clusters, K-Means clustering algorithm works as follows:

- Randomly select k data points to be the initial centroid;
- Assign each data point to the closest centroid;
- Update the centroids using the current cluster memberships by averaging the clustered points;
- If a convergence criterion is not met, repeat step 2 and 3 until convergence occurs.

The inputs of the K-Means algorithm are the number of clusters (i.e., k) and the selected features from the input data set. The centroids of the clusters are chosen by issuing a specific starting points or randomly selected by the algorithm. Iteratively, the algorithm assign each data point to one of the k clusters based on the minimum distance of that point to the centroids. In each iteration, the algorithm computes the average of all data points within a cluster and make that average position as a new centroid [11]. This process is repeated until it converges or if some other stopping condition is reached. This is performed by minimizing an objective function as defined below:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} ||x_i^{(j)} - \mu_j||^2, \tag{4}$$

Where $\left\| x_i^{(j)} - \mu_j \right\|^2$ is the Euclidean distance between data item $x_i^{(j)}$ and the cluster centroid μ_j.

In the context of DCA, DCs are selected randomly to sample data items and signals. As soon as a DC's lifespan exceeds the migration threshold, it ceases to sample any more data items, and the DC is migrated from the pool. The DC with its ID, cumulative $smDC$ and mDC context values is passed to the K-Means clustering algorithm. The overall process is shown in algorithm 2. The algorithm randomly initialises two cluster

centroids, one for semi-mature and the another for mature. The $smDC$ and mDC values calculated by the conventional DCA are used to initialize the initial centroid values in this work. Each centroid has two dimensions representing the $smDC$ and mDC. For each DC object, iteratively, the algorithm assigns it to one of the two clusters based on the distances of the ($smDC$ and mDC) context values to the centroids. Subsequently, the algorithm computes the average of all DC context values within a cluster and make that average position as the new centroid. The process of context assessment and centroid adjustment is repeated until the values of the centroids stabilize. Eventually, cumulative $smDC$ and mDC context values are sorted according to the minimum distance to the centroids.

The contexts of the DCs which are assigned to the mature cluster are set to 1, indicating that the sampled data items may be anomalous while the contexts of the other DCs in the semi-mature cluster are set to 0, showing that the sampled data items are likely to be normal. Once the clustering is performed for all the data items, the classification phase is performed in the same way as in the classical DCA. The derived values for the cell contexts (1/0) are used to determine the nature of the response by measuring the number of DCs that are in mature cluster which are represented by the MCAVs. To perform classification, the MCAV of each data instance is compared to the anomaly threshold (th), which is calculated by dividing the number of anomaly data instances presented in the training data set by the number of data instances in the data set. For each data item presented in the mature cluster, the MCAV is determined by dividing the number of times it is presented by different DCs in the mature cluster by the total number of sampling and presentation by different DCs. The data items with greater MCAVs than the pre-specified anomaly threshold are taken as potential anomalous.

Algorithm 2 Context assessment by K-Means

```
input : Data items with smDC, mDC values
output: anomalous or normal label for each data item
Arbitrarily initialize two cluster centroids;
repeat
  for each DC in pool P_m do
    Calculate the distance between the smDC and mDC
    values of this DC and the two centroids;
    (Re)assign the DC to the cluster with shorter Euclidean
    distance;
    if (cluster==mature) then
      DC-context=1;
    else
      DC-context=0;
    end if
    Update each centroid by the mean DC values;
  end for
until Convergence
```

IV. Experimentation

To validate and evaluate the proposed approach, the KDD99 [23] intrusion detection dataset was used.

A. Dataset Description

The KDD99 data set is an intrusion detection dataset which has been widely used for the evaluation of anomaly detection methods [23]. This data set has also been used to build a network intrusion detector as a predictive model with an ability of distinguishing between bad and good connections. There are four attack categories in the KDD99 data set:

- **DOS**: Denial of service attacks which intend to limit legitimate users from accessing the system e.g. syn flooding, teardrop and smurf.
- **Probes**: An attempt of gaining access to a computer and its files by exploiting the weak points available through surveillance and other probing techniques, e.g. port scanning.
- **U2R**: Unauthorized attempt to gain super user privileges by exploiting vulnerabilities that allow normal user to gain a root privileges, e.g. buffer overflow and rootkit attacks.
- **R2L**: Unauthorized access of a computer resources from a remote machine, e.g. password guessing and ftp_write attacks.

Notice that it is a common practice to use 10% of the KDD99 training data set [15], [17], [24]; this work also follows this practice. This data set consists of 494,021 instances among which 97,277 (19.69%) are normal, 391,458 (79.24%) DOS, 4,107 (0.83%) Probe, 1,126 (0.23%) R2L and 52 (0.01%) U2R connections. For computation efficiency, during DCA weights optimization by the GA, only part of the KDD99 training data set was utilised, which is 6.3% (311,029 instances) of the original training data set, and consists of 19.5% (60593 instances) of normal connections and 80.5% (250436 instances) of anomaly connections.

B. Experiment Setup

All experiments were performed on Intel Core i5 6200U 2.4GHz - 8GB RAM-HP running windows 8. The proposed system is implemented in Java using NetBeans IDE 8.2. Twelve attributes were selected based on the approach presented in Section III-A to generate the three input signals; and the detailed categorised features are:

- **PAMP**: serror_rate, srv_serror_rate, same_srv_rate, dst_host_serror_rate, dst_host_rerror_rate, rerror_rate, and srv_rerror_rate.
- **DS**: count and srv_count.
- **SS**: logged_in, srv_different_host_rate, and dst_host_count.

To differentiate data items during signal processing, data item IDs are created by combining three nominal attributes, which are protocol, service and flag. This help to trace a particular data item (i.e., antigen in biological term) in the

system since one antigen is processed by multiple DCs at the same time.

The inputs to the proposed system are data items and the three signal categories of PAMP, DS and SS used by DCs to derive the context values of $smDC$ and mDC, which are then applied to the K-Means clustering algorithm for context assessment before classification. To calculate the value of each signal category, all the selected attributes are normalized into a range of 0 to 1 utilizing the simple min-max (MM) normalisation technique [25]. Then the values of each signal category is calculated as the average of all attributes that jointly form this signal. Subsequently these values are combined with the corresponding data items and become input of the DCA.

In the experiments, the DCA weights were optimised using the GA, and the results are listed in Table I. A population of 100 DCs is used in the GA where 10 DCs are selected randomly to sample data items and signals. The DC migration threshold is set to 10. To perform classification of data items, an anomaly threshold is needed to evaluate the MCAVs. The number of anomalous class data items present in the 10% of KDD99 data set is 80% and therefore, the anomaly threshold is set to 0.8. Hence, if the MCAV value is greater than anomaly threshold (0.8) in the mature cluster, the antigen is classified as anomalous, otherwise normal.

To evaluate the performance of the proposed approach, the rate of True Positive (TPR), True Negative (TNR), False Negative (FNR) and False Positive (FPR) are calculated in addition to the overall classification accuracy (Acc). These measures are defined as follows: TPR=TP/(TP+FN), TNR=TN/(TN+FP), FPR=FP/(TP+FN), FNR=FN/(TN+FP); and the Acc is equal to the number of correctly classified data instances divided by the total number of instances classified.

Table I: The optimised DCA weights

	Weights	**Values**
CSM	W_{PAMP}	2
	W_{SS}	1
	W_{DS}	2
smDC	W_{PAMP}	2
	W_{SS}	3
	W_{DS}	6
mDC	W_{PAMP}	6
	W_{SS}	5
	W_{DS}	2

Three experiments were carried out in this study:

- The first experiment (classical DCA) was carried out by using 10% of KDD99 training dataset and the classical DCA with optimized weights generated by GA.
- The second experiment (DCA+K-Means) was performed by using the optimized weights plus the K-Means clustering on the same data set. DCA uses its weighted sum function and the optimized weights to compute $smDC$ and mDC context values, which are forwarded to the K-Means clustering algorithm. K-Means follows the procedures as described in Section III-C to assign the data items to a cluster with minimum distance to its centroid.
- The third experiment was conducted to compare the classification accuracies between the proposed approach and the DCA when considering different attacks presented in the KDD99 data set. The KDD99 data set consists of normal connections and four known attack types (Dos, Probe, U2R and U2L).

The results led by experiments one and two are presented in Table II while the results of the third experiment is listed in Table III.

Table II: Experiment 1 & 2 Classification Results

	TPR	**FPR**	**TNR**	**FNR**	**Acc**
Classical DCA (%)	98.52	7.71	92.29	1.48	97.29
DCA+KMeans(%)	98.68	4.72	95.28	1.32	98.01

Table III: Classification results for different attack types

Acc	**Normal**	**DoS**	**Probe**	**U2R**	**R2L**
Classical DCA(%)	89.12	99.42	68.56	29.08	5.72
DCA+KMeans(%)	94.23	99.57	87.54	28.50	13.42

C. Results Analysis and Comparison

The experiments show that the application of the K-Means clustering algorithm to the DCA context assessment can improve classification accuracy, as demonstrated in Tables II and III. The proposed approach has produced higher accuracy (98.01%) compared to the classical DCA (97.29%). This indicates an improved percentage of successful classified data items by 0.71%. Additionally, the TPR, FPR, TNR and FNR also indicate notable improvements. For instance the TNR is increased from 92.29% to 95.28%, expressing a positive improvement of 2.99% whilst the FPR is decreased by 2.99%. Comparing the performance using different attack types, the proposed approach outperforms classical DCA as shown in Table III except for U2R attack type where the classification accuracy is slightly decreased from 29.08% to 28.50%.

We have also compared the result of the proposed approach with a recently proposed version of the DCA known as Fuzzy Classification Dendritic Cell Method which uses the Gustafson-Kessel clustering algorithm (FCDCMGK) presented in [17] and other five well-known classifiers which are J48 Decision Tree (JDT), Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF) and Artificial Neural Network (MLP). The experiments for JDT, NB, SVM, RF and MLP classifiers were conducted by using the Weka software [26] with the parameter values of the algorithms set to the default.

The comparison was performed in terms of the overall accuracies on the 10% of KDD99 training data sets and the results are presented in Figure 2. From this figure, it is clear

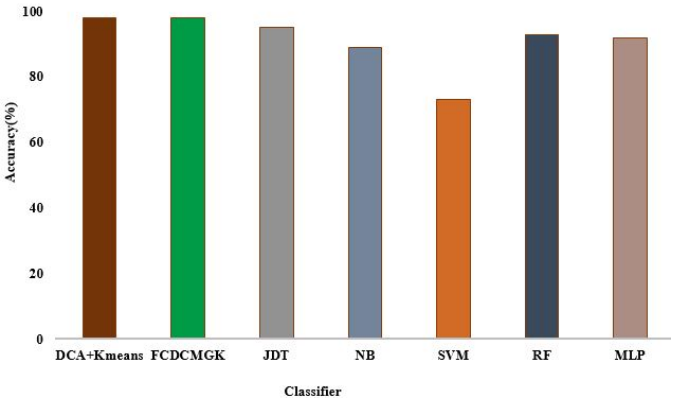

Figure 2: Comparison of the Classifiers in Terms of Classification Accuracy

that the classification performance of the proposed approach is comparable to the FCDCMGK approach, and outperforms JDT, NB, SVM and MLP, in terms of overall classification result.

V. Conclusions

This paper proposed an alternative approach for the DCA context assessment process using the K-Means clustering algorithm. More precisely, the K-Means algorithm is applied in this work to cluster the cumulative $smDC$ and mDC context values into two clusters, namely semi-mature and mature, to address the challenge of hard context-based separation. The experimental results demonstrate the efficiency of the proposed approach in terms of TPR, TNR, FPR and FNR, in addition to the overall accuracy. Compared to other classifiers, the overall accuracy result of the proposed approach is comparable to the recently proposed DCA version based on Gustafson-Kessel clustering technique (FCDCMGK) [17] and are generally competitive in reference to other commonly used classifiers such as JDT, NB, SVM, RF and MLP. Although promising on the static KDD99 dataset, the proposed approach needs to be evaluated using real time data set to evaluate its performance especially when traffic behaviours change rapidly overtime. In addition, the feature categorisation phase has been further developed using fuzzy interpolation [27]–[29] to address the potential non-linearity problem. It would be interesting to integrate such approaches to explore the full potential of DCA.

References

[1] Longzhi Yang, Noe Elisa, and Neil Eliot. *Privacy and Security Aspects of E-Government in Smart Cities*. Elsevier Press, 2018.

[2] Louis-François Pau. Business and social evaluation of denial of service attacks of communications networks in view of scaling economic counter-measures. In *2010 IEEE/IFIP Network Operations and Management Symposium Workshops*, pages 126–133. IEEE, 2010.

[3] Polly Matzinger. The danger model: a renewed sense of self. *Science*, 296(5566):301–305, 2002.

[4] Jacques Banchereau and Ralph M Steinman. Dendritic cells and the control of immunity. *Nature*, 392(6673):245–252, 1998.

[5] Julie Greensmith, Uwe Aickelin, and Steve Cayzer. Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. In *ICARIS*, volume 3627, pages 153–167. Springer, 2005.

[6] Obinna Igbe, Ihab Darwish, and Tarek Saadawi. Deterministic dendritic cell algorithm application to smart grid cyber-attack detection. In *Cyber Security and Cloud Computing (CSCloud), 2017 IEEE 4th International Conference on*, pages 199–204. IEEE, 2017.

[7] Yousof Al-Hammadi, Uwe Aickelin, and Julie Greensmith. Dca for bot detection. In *IEEE Congress on Evolutionary Computation*, pages 1807–1816. IEEE, 2008.

[8] Zeineb Chelly and Zied Elouedi. Fdcm: A fuzzy dendritic cell method. In *ICARIS*, volume 2010, pages 102–115. Springer, 2010.

[9] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995.

[10] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[11] Shruti Kapil and Meenu Chawla. Performance evaluation of k-means clustering algorithm with various distance metrics. In *Power Electronics, Intelligent Control and Energy Systems (ICPEICES), IEEE International Conference on*, pages 1–4. IEEE, 2016.

[12] Frank M Burnet. Immunological recognition of self. *Science*, 133(3449):307–311, 1961.

[13] Thomas Stibor. *On the appropriateness of negative selection for anomaly detection and network intrusion detection*. PhD thesis, Technische Universität, 2006.

[14] Uwe Aickelin, Peter Bentley, Steve Cayzer, Jungwon Kim, and Julie McLeod. Danger theory: The link between ais and ids? *Artificial immune systems*, pages 147–155, 2003.

[15] Feng Gu, Julie Greensmith, and Uwe Aickelin. Further exploration of the dendritic cell algorithm: Antigen multiplier and time windows. *Artificial immune systems*, pages 142–153, 2008.

[16] Zeineb Chelly and Zied Elouedi. A fuzzy-rough data pre-processing approach for the dendritic cell classifier. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 109–120. Springer, 2013.

[17] Zeineb Chelly and Zied Elouedi. Hybridization schemes of the fuzzy dendritic cell immune binary classifier based on different fuzzy clustering techniques. *New Generation Computing*, 33(1):1–31, 2015.

[18] Zeineb Chelly and Zied Elouedi. A survey of the dendritic cell algorithm. *Knowledge and Information Systems*, 48(3):505–535, 2016.

[19] Noe Elisa, Longzhi Yang, and Nitin Naik. Dendritic cell algorithm with optimised parameters using genetic algorithm. In *2018 IEEE Congress on Evolutionary Computation (IEEE CEC 2018)*. IEEE, 2018.

[20] Ryszard S Michalski, Jaime G Carbonell, and Tom M Mitchell. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[21] H Günes Kayacik, A Nur Zincir-Heywood, and Malcolm I Heywood. Selecting features for intrusion detection: A feature relevance analysis on kdd 99 intrusion detection datasets. In *Proceedings of the third annual conference on privacy, security and trust*, 2005.

[22] John H Holland. Adaptation in natural and artificial systems. an introductory analysis with application to biology, control, and artificial intelligence. *Ann Arbor, MI: University of Michigan Press*, pages 439–444, 1975.

[23] KDD Cup 1999 Data. "http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html/. Accessed: 2018-01-11.

[24] Obinna Igbe, Oluwaseyi Ajayi, and Tarek Saadawi. Denial of service attack detection using dendritic cell algorithm.

[25] Zheming Zuo, Jie Li, Philip Anderson, Longzhi Yang, and Nitin Naik. Grooming detection using fuzzy-rough feature selection and text classification. In *Fuzzy Systems (FUZZ-IEEE), 2018 IEEE International Conference on*. IEEE, 2018.

[26] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[27] Longzhi Yang and Qiang Shen. Closed form fuzzy interpolation. *Fuzzy Sets and Systems*, 225:1 – 22, 2013. Theme: Fuzzy Systems.

[28] Longzhi Yang, Fei Chao, and Qiang Shen. Generalized adaptive fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, 25(4):839–853, Aug 2017.

[29] Noe Elisa, Jie Li, Zheming Zuo, and Longzhi Yang. Dendritic cell algorithm with fuzzy inference system for input signal generation. In *Advances in Intelligent Systems and Computing*, Cham, 2018. Springer International Publishing.