Int. J. Advance Soft Compu. Appl, Vol. 6, No. 3, November 2014
ISSN 2074-8523

Integration Of The Dendritic Cell Algorithm With K-Means Clustering

Mohamad Farhan Mohamad Mohsin, Azuraliza Abu Bakar, and Abdul Razak Hamdan

Data Mining and Optimization Research Group, Centre for Artificial Intelligence Technology, Faculty of Science & Information Technology, Universiti Kebangsaan Malaysia, Selangor, Malaysia.
e-mail: farhan@uum.edu.my, aab@ftsm.ukm.my, and arh@ftsm.ukm.my

Abstract

The dendritic cell algorithm is an effective technique to detect anomalies in time series applications. However, the algorithm is less effective when it mines a general classification dataset because the items are not organized in an orderly event-driven manner. Ideally, for they need to be arranged in sequence by sorting them according to decision class. However, it is not practicable to apply this step because the decision classes for real datasets is unknown. Therefore, an integrated model that combines the dendritic cell algorithm and the k-means algorithm is proposed as an alternative to the existing sorting function based on decision class. The proposed model is evaluated by applying it to eight universal classification datasets and assessing its performance according to four evaluation metrics: detection rate, specificity, false detection rate, and accuracy. The results show that the proposed clustered dendritic cell algorithm is more effective than the non-clustered version. When applied to a benchmark dataset, the clustered dendritic cell algorithm demonstrates significant improvement in performance on the unordered version of the dataset and generates a comparable result to that of its competitor. For the other seven datasets, the proposed algorithm generates better specificity, false detection rate, and accuracy. The findings indicate that item–centroid distance within a cluster can be adopted to transform an unordered dataset into a sequential dataset, thus fulfilling the dendritic cell algorithm requirement for ordered data.

Keywords: *Artificial immune system, clustering, dendritic cell algorithm, k-means*

1 Introduction

The dendritic cell algorithm (DCA) is a biologically-inspired algorithm that belongs to the artificial immune system (AIS) category of heuristics. This essentially means that it is modelled based on the concept of danger theory, which posits that the human immune system is triggered when a dendritic cell recognizes a danger signal released by an unexpected cell death due to pathogenic infection. In the same way as it is the responsibility of the dendritic cell to recognize an intruder (bacteria, virus, parasite) that enters the body, the DCA is modelled to detect anomalies in the computer context. The first DCA prototype was initially developed in the field of computer network security, where the dendritic cell was employed to act as an agent to detect suspicious network intruders [1]. Subsequently, Greensmith et al [2] implemented a fully functioning real-time network intrusion detection system. Since this successful implementation, the algorithm has been widely applied in various areas, mainly to time series anomaly detection-based problems including intrusion [3], fault [4], fraud [5], and outbreak [6] detection. The published results of these applications demonstrate that DCA performs well in terms of producing a high detection rate and lower false detection rate in comparison to other systems. Besides this good detection output, the DCA has distinct advantages over other data mining approaches in recognizing anomalies because it employs the dangerousness of an antigen, known as the multi-context antigen value (MCAV), rather than a pattern-matching approach. Moreover, the algorithm does not require an extensive training phase and can be implemented in real-time applications with very low CPU processing requirements.

The DCA is designed for use with time series applications; in other words it is suitable for a problem with a time-dependent component. This algorithm demonstrates good detection performance when each item in the dataset is organized in an orderly event-driven manner in relation to its neighbouring item. However, the DCA is less effective in mining static and unordered datasets because of the nature of its design, which is characterized by a crisp separation between normality (semi-mature) and abnormality (mature) [7]. The experimental results of Greensmith [8] show that DCA has impeded performance with a higher detection error rate when it mines an unordered dataset. The result of his experiment is shown in Table 1, from which it can be seen that DCA generates more detection errors (FN=320, FP=78) for the unordered Wisconsin Breast Cancer (WBC) than the ordered WBC dataset. The TN, FP, TP, FN headings in the table denote true negative, false positive, true positive, and false negative.

Table 1: Number of errors for ordered and unordered WBC [8]

	TN	FP	TP	FN
Ordered	240	0	403	57
Unordered	162	*78*	140	*320*

The sensitivity of the DCA to data order means that it is less suitable for application to a general classification dataset because this type of dataset is not organized in time order. To overcome this problem, the facility to read an unordered dataset in an event-driven manner is needed. One proposed solution is to sort the data according to decision class [8]. By sorting the data in this way, items of a similar class are arranged in a sequential manner and therefore any changes that occur to the sequence can be recognized by the DCA because it is sensitive to changes in context. However, this approach is not practical because the decision classes in most real-world datasets are unknown.

To date, research on methods to handle unordered data for the DCA has been limited, particularly in relation to the general classification dataset. One of the notable recent approaches is the Multiplying and Merging Algorithm (MMDCA) [9]. In the MMDCA, two steps are applied. First, the *n* instances of each antigen are multiplied several times during the antigen sampling stage. After that, the MCAV of each antigen is determined based on two options: (1) comparing each MCAV with the anomaly threshold and adopting the majority or (2) calculating the average of each of *n* MCAV and comparing it with the anomaly threshold. The authors tested their algorithm on benchmark data and found that when applied to the unordered WBC it achieved better classification accuracy than the original DCA. However, the effectiveness of the performance of the MMDCA for other general classification data is as yet unknown.

In this paper, we propose integrating the DCA with a k-means clustering algorithm to handle the unordered dataset. The two algorithms are integrated during the data preparation phase, before the dataset is presented to a specific DCA signal normalization algorithm. The aim of this approach is to group similar items into similar clusters so that they can be sorted accordingly, and thus to replace the existing sorting method that uses the decision class as the sorting criterion. The proposed integration model is tested on a benchmark dataset, namely the WBC dataset. It is also tested on seven other universal classification datasets; six from the UCI Machine Learning Repository [10] and one the StatLib Archive [11]. The performance of the proposed modification to the DCA is compared with the standard DCA [8] and the MMDCA [9] on unordered data. The model is evaluated with respect to four criteria: detection rate, specificity, false detection rate, and accuracy.

The remainder of the paper is organized as follows: Section 2 outlines the concept of the DCA. Section 3 presents the proposed model and explains how the DCA is integrated with the k-means algorithm. Section 4 describes the experimental setup. Then Section 5 presents the main results and Section 6 contains a discussion of the results. The final section, Section 7, concludes this work.

2 The Dendritic Cell Algorithm

A DCA is an abstraction model of the AIS paradigm that draws on danger theory [1]. It utilizes the role of the dendritic cell (DC) as an agent that monitors the antigens' life cycle until their death. In a biological immune system, danger theory views all cells in the human body as antigens that have a similar possibility of being infected by harmful pathogens. At the beginning of the detection process, the DCs, which are born as immature cells, observe the progress of the body's cells. Termed as input, the DC collects the body cell protein paired with its three signals; pathogen associated molecular patterns (PAMP), danger (DS), and safe signal (SS). Based on the collected input throughout its life span, the DC will evolve from being immature into one of two maturation states; either semi-mature (apoptotic death) or mature (necrotic death). Reaching a mature state indicates that the cell has experienced more danger signals throughout its life span that have been caused by foreign antigens, wounds, etc. If this happen, it indicates that an antigen has been detected and a danger zone will be released. A semi-mature state indicates that apoptotic death has occurred and this is seen as part of normal cell function and is tolerized to the presented antigen.

Analogized from danger theory, the DCA is formalized as depicted in Fig. 1. In the DCA, each data item in the monitored system is viewed as an antigen such that they can similarly be infected by a harmful pathogen. As the DCA is a population-based algorithm, DCs perform multiple input signal and antigens sampling. The DCA collects three input signals (PAMP, DS, and SS) tailored to antigens, calculates the changes and then determines which antigen is causing the changes. Using the accumulative function in Equation 1, all input signals are transformed into three cumulative output signals: co-stimulatory molecules (CSM), mature, and semi-mature [8]:

$$OS_j(x) = \left(\sum_{i=0}^{i=3} W_{ij} * ISi_j(x)\right) / \left(\sum_{i=0}^{i=3} W_{ij} * ISi_j(x)\right) \tag{1}$$

where W is the weight matrix, IS is the input signal, OS is the output signal, i is the input signal categories, and j is the output signal categories. The weight equation as applied in Equation 1 is presented in Table 2. The W is user defined; however, the relative interaction between the input signals and output signal must remain constant.

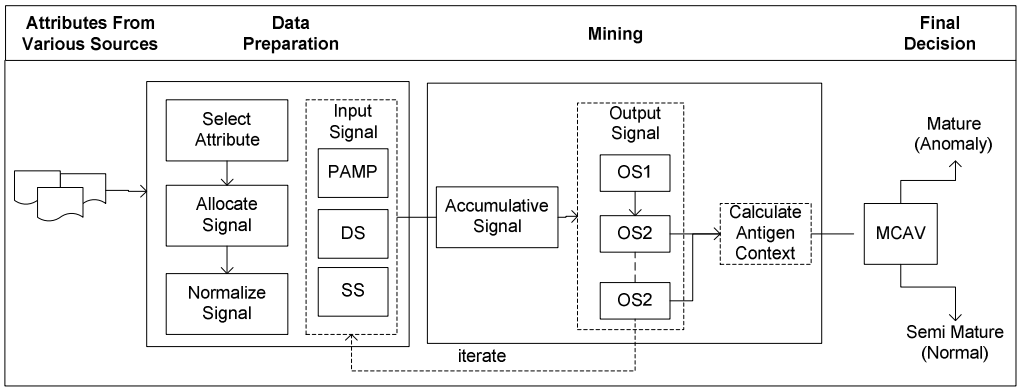

Fig 1: Dendritic cell algorithm.

Table 2: The weights used in the cumulative function [8].

w_{ij}	PAMP (i=1)	DS (i=2)	SS (i=3)
CSM (j=1)	W_1	$W_1/2$	W_1
Semi-mature (j=2)	0	0	1
Mature (j=3)	$W_2 * (1.5)$	$W_2/2$	$W_2 * (-1.5)$

When monitoring begins, all DCs are in the immature state [12]. Throughout multiple antigen-signal sampling, DCs collect various experiences which may influence and change their maturity level. The maturation information is recorded via CSM ($OS1$), mature ($OS2$), and semi-mature ($OS3$) as output signals. As soon as a DC exceeds the maturation level, the sampling process being undertaken by that cell stops. This occurs when the DC has a $OS1$ value greater than the migration threshold and, as a result, the DC is migrated from the population for antigen presentation. After that, the output values $OS2$ and $OS3$ are compared in order to derive a context for the presented item. The antigen is termed as mature if $OS2>OS3$ or semi-mature if $OS2<OS3$. Then the migrated DC is replaced with a new cell to restart sampling and return to the population. This process is iterated several times.

When learning ends, antigens appear in different contexts. In the last step, the potential anomalous antigen is determined based on the collected context. Termed as the mature context antigen value (MCAV), the anomalous antigen is determined as [8]:

$$MCAV_i = \left(\sum AG_{mi}\right) / \left(\sum AG_m + \sum AG_{sm}\right) \quad (2)$$

where i refers to the antigen type, $\sum AG_{mi}$ refers to the total number of mature antigens of antigen type i, $\sum AG_m$ is the total number of mature antigens, and

$\sum AG_{sm}$ refers to the total number of semi-mature antigens. Those antigens with a $MCAV_i$ greater than the anomaly threshold are classified into the anomalous group while the others are classified into the normal category.

3 Integration of DCA with K-Means

In the proposed model, the k-means algorithm is integrated with the original DCA process as part of the data preparation phase; after the most relevant attributes have been selected and assigned into appropriate PAMP, DS, and SS. The reason for this integration is to utilize k-means as a sorting function to group similar data items into similar clusters and sort them according to their distance from the cluster centroid. The steps in Fig. 2 show how k-means is integrated into the data preparation process.

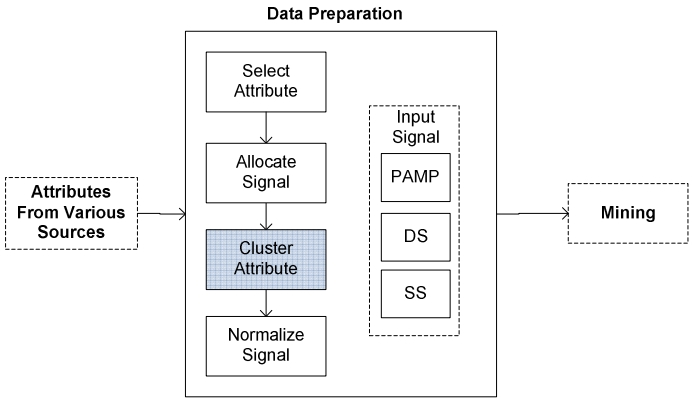

Fig. 2 Integration of k-means into DCA.

K-means clustering is a well-known data mining technique. It is a partition-based type of clustering [13], where a dataset is partitioned into *k* clusters. Categorized as an unsupervised learning approach, it relies on the concept that any items within a cluster inherit similar characteristics and that they are distinct from other items in other clusters. The algorithm finds the best clusters based on two general steps. First, data items are assigned to clusters based on the current centroids. After that, new centroids are updated based on the current assignment of data items to clusters until the process converges. Throughout this process, an item is placed in a particular cluster if it is closer to that cluster's centroid than any other centroid.

Based on this assumption, the clustering output is adopted for the use of DCA to transform the unordered dataset into an event-driven dataset by organizing the data items according to item–centroid distance. The clustering mechanism for the DCA follows the standard k-means activities except for an amendment to the final stage. In k-means, first the number of clusters *k* has to be determined. As anomaly

detection in the DCA is a two class classification, the default *k* number is set to two; representing the normal and abnormal group. When the clustering processes end, one of the clusters is selected and all the items are then sorted according to the distance between the data points and cluster's centroid. The k-means clustering for DCA is shown in Fig. 3.

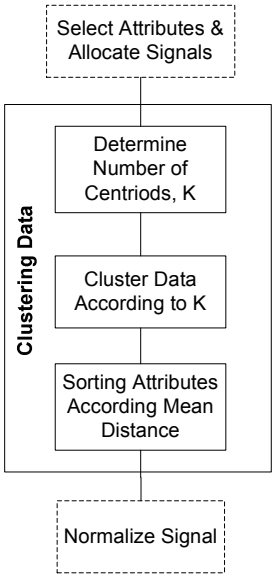

Fig. 3 Clustering mechanism based on k-means for DCA.

In this study, the items are sorted in ascending order starting with the shortest distance to the centroid, as shown in Table 2. Table 3(a) indicates a sample of the distance of each data point (#) from two clusters (C1, C2) while Table 3(b) shows the result after all data points have been sorted according to C1. Table 3(b), most of the records are arranged according to an accurate decision class (CLASS) except for two records; #519 and #52, which are inaccurately clustered into the wrong group. For the DCA, the accurateness of the cluster is vital to ensure the dataset can be sorted accurately into a time-based order. Fig. 4 depicts the integration of DCA and k-means to create the proposed clustered DCA.

Table 3: The distance between data item (#) and each cluster (C1, C2) of the WBC.

#	C1	C2	CLASS
1	225.48	5.75	2
2	36.53	216.66	2
3	157.24	29.26	2
4	169.22	5.48	2
5	101.45	72.17	1
6	63.31	414.89	1
7	197.51	2.53	2

#	C1	C2	CLASS
239	10.36	188.53	1
633	13.11	266.41	1
530	16.21	251.55	1
400	17.86	242.14	1
155	19.78	162.75	1
678	20.96	139.10	1
500	21.57	284.27	1

8	204.83	1.60	2
9	138.59	7.48	2
10	24.58	128.03	1
11	196.29	2.03	2
12	128.57	337.67	1
13	188.86	10.38	2
14	214.16	2.67	2
15	169.45	19.24	2
16	171.42	5.87	2
17	212.04	0.79	2
18	180.51	6.89	2
19	161.04	5.74	2
20	33.15	162.85	1

(a) Before sorting

685	23.01	281.70	1
263	24.07	291.06	1
268	24.21	250.37	1
90	24.28	286.61	1
10	24.58	128.03	1
160	25.04	212.16	1
519	25.19	155.63	2
52	25.56	266.69	2
571	26.75	166.58	1
234	26.86	209.83	1
286	27.18	265.91	1
465	30.02	256.24	1
395	30.05	239.22	1

(b) After sorting

Input: Raw Data; Number of Clusters, K
Output: Sorted Data
0 START
1 Get input from Attribute-Signal Mapping Algorithm
2 Initialize cluster centroids; $\mu_1, \mu_2, \ldots, \mu_n \in \mathbb{R}^n$
4 Repeat until convergence
5 Calculate distance between data point and each centroids , Ck
6 (re)assign each data point to the cluster to which the point is the most similar
7 Update the cluster means
8 Select one cluster
9 SORT data according distance between data point and centroids
10 END
11 Normalize input signal

Fig. 4 Proposed clustered DCA.

4 Experiment Setup

To test the validity of the proposed model, we apply it to eight universal classification datasets; seven from UCI Machine Learning Respiratory [10] and one from the StatLib Archive [11], as described in Table 4. The aim of the experiment is twofold: first, to evaluate the performance of the proposed integrated DCA with k-means on WBC (the benchmark dataset for DCA before and after integration with k-means) with that of the original (non-clustered) DCA and the MMDCA; second, to investigate the effect of different cluster numbers on the accurateness of the proposed clustered DCA by testing it on seven other datasets.

For the experiment, the datasets are prepared in three forms: ordered dataset, unordered dataset, and clustered dataset. The term 'ordered' is where the dataset is sorted by decision class, 'unordered' is where the data is randomly unsorted, and 'clustered' means the dataset is random but then sorted by k-means. All datasets need to have two decision classes: abnormal and normal. For the datasets with more than two decision classes, namely IRIS and WINE, the decision class attribute is restructured so that one of the decision classes is set as the abnormal group while the rest are labelled as the normal group.

Table 4: Description of the datasets.

Dataset (* Benchmark dataset)	Origin	Attributes #	Records #	Class #
Indian Pima Diabetic (DBC)	[10]	9	768	2
Wisconsin Breast Cancer (WBC)*		10	699	2
Iris (IRIS)		4	150	3
BUPA Liver Disorder (LDR)		7	345	2
Parkinson (PKN)		24	195	2
German Credit (GCD)		25	1000	2
Wine (WINE)		14	178	3
Biomedical (BIO)	[11]	6	209	2

The initial parameter setting is formalized as follows: in all experiments, a population of 100 cells is created and the total cycle cell update is set to 20. In every cycle, DCs are allowed to perform antigen sampling 10 times. The weight for the accumulative function is set to W1=1 and W2=2. The experiment is repeated 100 times and the average of each evaluation metric is recorded for analysis.

For attribute selection, the standard deviation of each attribute is retrieved and they are ranked according the highest standard deviation [8]. Based on the highest ranking, four attributes are chosen for mining, where the attributes at rank 1 and rank 2 are set as DS and PAMP, respectively, and the rest of the attributes are set as DS. After that, the attributes are normalized according to the DCA signal. The cumulative sum normalization technique is used to normalize the DCA signal, as depicted in Equation 3 and Equation 4 [6]:

$$C_i^+ = max\,[0, x_i - (\mu_0 + K) + C_{i-1}^+] \quad (3)$$

$$C_i^- = max\,[0, (\mu_0 - K) - x_i + C_{i-1}^-] \quad (4)$$

where $C_i^{+@-}$ is the cumulative sum value. If $C_i^{+@-}$ is greater than or equal to 0, the cumulative sum value is taken as the normalized value. The $C^+$is used to normalize the PAMP while the DS and SS are normalized with C^-.

To evaluate the performance of the proposed model we examine the algorithms' results using four evaluation metrics: detection rate (DR), false detection rate (FDR), specificity (SPS), and accuracy (ACC): DR measures the accurateness of the model to detect an abnormal class as an abnormal class DR=TP/(TP+FN); SPS measures the ability of the model to detect a normal class as a normal class SPS=TN/(TN+FP); FDR measures the amount of false detections of an abnormal class as a normal class FDR= FP/(TN+FP); and ACC measures the accurateness of the model in classifying both classes correctly ACC= (TP+TN)/(TP+TN+FN+FP). For DR, SPS, and ACC, the highest value indicates the best result while for FDR the best result is the lowest value.

5 Results and Findings

This section aims to show that the proposed integration of the DCA and k-means can improve the DCA's detection capability for unordered datasets. First, an experiment is carried out on the benchmark dataset, the WBC dataset. The results of the performance of the proposed integration model are presented in Table 5, which shows a comparison of the full results for DCA when WBC is presented to it in different input presentations; ordered WBC, unordered WBC, and clustered WBC. The ordered and unordered WBC is presented to the original (non-clustered) DCA (DCA_OR) model and their result published in Table 5 are taken from Greensmith [8]. Meanwhile, the proposed integrated model (DCA_KM) mines the clustered WBC and the dataset is clustered into two groups (k=2) before it is presented to the DCA.

Table 5: Comparison between ordered, unordered, and clustered WBC presented in DCA.

	TN	FP	TP	FN
DCA_OR (Ordered WBC)	240	0	403	57
DCA_OR (Unordered WBC)	162	*78*	140	*320*
DCA_KM (Clustered WBC)	237	4	429	29

From Table 5, the DCA_OR generates higher false detection errors (FN) for unordered WBC than the ordered WBC such that 320 anomalous records are incorrectly detected as normal. This is similar to the result for the number of false negatives (FP=78), which is again higher for the unordered WBC than the ordered WBC. However, in the case of the proposed model, the DCA_KM, there is a significant improvement in the handling of the unordered dataset. In comparison to the DCA_OR for unordered WBC, the DCA_KM generates a lower number of FNs (4) and FPs (29) as well as maintains the number of TNs and TPs. Moreover,

the proposed DCA's result for clustered WBC is comparable to that of the DCA_OR for ordered WBC version. This shows that k-means in DCA_KM can sort WBC items as accurately as ordered WBC that pre-sorted its items accordingly to decision class.

Next, the proposed DCA_KM model is compared with MMDCA [9]. The comparison of the results is shown in Table 6, from which it can be seen that both models exhibit a comparable ability when dealing with the unordered WBC dataset. From Table 6, the DR of DCA_KM is slightly lower than that of MMDCA; however, the results for DR are not significantly different. In spite of that, the DCA_KM produces higher SPS, higher ACC, and lower FDR than MMDCA, which shows that the proposed model has the ability to classify normal records more accurately than MMDCA. Nevertheless, the results for SPS, FDR, and ACC are not significantly different.

Table 6: Results of DCA_KM and MMDCA for unordered WBC.

	DR	SPS	FDR	ACC
DCA_KM	0.936747	0.982404	0.017593	0.952488
MMDCA	0.97236	0.97893	0.02107	0.95129
Difference	(-0.035613)	0.003474	0.003477	0.001198

In the second part of the experiment we examine whether the number of clusters affects the detection performance of the proposed model. For this analysis, seven unordered datasets are presented to the DCA_KM and there are five different numbers of clusters, k. The results are shown in Table 7, which consists of four sub-tables, one for each evaluation metric (DR, SPS, FDR, and ACC). Each row represents the result for each dataset (BIO, GCD, IRIS, PKN, DBC, WINE, and LDR) while each column represents one of the five k assessed in the study (2, 3, 4, 5, and 10).

Table 7: The effect of number of clusters on the efficacy of the proposed model.

	Cluster, k					**Cluster, k**				
	2	**3**	**4**	**5**	**10**	**2**	**3**	**4**	**5**	**10**
BIO	0.855	0.781	0.775	0.747	0.716	0.721	0.661	0.702	0.712	0.712
GCD	0.332	0.306	0.313	0.320	0.316	0.806	0.805	0.803	0.789	0.804
IRIS	0.936	0.777	0.764	0.821	0.853	0.747	0.735	0.746	0.723	0.712
PKN	0.739	0.731	0.726	0.693	0.730	0.433	0.367	0.418	0.411	0.431
DBC	0.324	0.329	0.326	0.333	0.329	0.779	0.663	0.672	0.661	0.667
WINE	0.758	0.605	0.546	0.538	0.539	0.640	0.578	0.564	0.579	0.609
LDR	0.550	0.535	0.529	0.520	0.468	0.463	0.469	0.530	0.553	0.590
	DR					**SPS**				

	Cluster, k					Cluster, k				
	2	**3**	**4**	**5**	**10**	**2**	**3**	**4**	**5**	**10**
BIO	0.279	0.339	0.298	0.288	0.248	0.721	0.661	0.702	0.712	0.712
GCD	0.194	0.191	0.197	0.211	0.194	0.806	0.805	0.803	0.789	0.804
IRIS	0.253	0.265	0.254	0.277	0.289	0.747	0.735	0.746	0.723	0.712
PKN	0.567	0.633	0.582	0.589	0.571	0.433	0.367	0.418	0.411	0.431
DBC	0.221	0.337	0.328	0.339	0.333	0.779	0.663	0.672	0.661	0.667
WINE	0.360	0.422	0.436	0.421	0.391	0.640	0.578	0.564	0.579	0.609
LDR	0.537	0.531	0.470	0.447	0.410	0.463	0.469	0.530	0.553	0.590
	FDR					**ACC**				

From Table 7, DCA generally generates good detection results when k is set to 2. However, the value of each evaluation metric (DR, SPS, ACC) slightly drops (increases for FDR) when the number of clusters increases; in other words, performance declines. The best k are summarized in Table 8. It is evident from Table 8 that for most datasets k=2 is the best because it contributes to a higher detection result. The one exception is the LDR dataset, where the proposed model requires a higher number of clusters to detect anomalous data well.

Table 8: The best number of clusters to use with the proposed method.

	BIO	**GCD**	**IRIS**	**PKN**	**DBC**	**WINE**	**LDR**
DR	2	2	2	2	4	2	2
SPS	2	2	2	2	2	2	10
FDR	2	2	2	2	2	2	10
ACC	2	2	2	2	2	3	10
Best Cluster	2	2	2	2	2	2	10

The above results are further analysed by comparing the performance of the proposed integrated DCA_KM and DCA_OR on seven unordered datasets. To represent the DCA_KM, we choose the best model with the best k cluster, as summarized in Table 8. The full results of our additional analysis are illustrated in Table 9, which indicates the comparative result DCA_OR and DCA_KM and the two rows above the last row summarize (1) the average values of each performance metric and (2) the results for all datasets in term of wins, ties, and losses (indicated by W/T/L) at the 5% level ($p<0.05$). The W/T/L measurement is considered in addition to the average measurement because the average criteria would be susceptible to outliers. The p value (p_val) represents the Wilcoxon test result, where the value of the DCA_KM must be less than 0.05 to make it statistically significant compared to the DCA_OR. The last row totals the significant (+) and not significant (-) datasets.

Table 9: Comparative results of DCA_KM and DCA_OR for seven datasets

	DR				SPS			
	DCA_ OR	**DCA_ KM**	**Diff**	**p_val**	**DCA_ OR**	**DCA_ KM**	**Diff**	**p_val**
BIO	0.4753	0.855	0.379^{W}	0.000^{+}	0.685	0.721	0.036^{W}	0.001^{+}
GCD	0.3774	0.332	-0.045^{L}	0.000^{+}	0.722	0.806	0.084^{W}	0.000^{+}
IRIS	0.6582	0.936	0.278^{W}	0.000^{+}	0.721	0.747	0.026^{W}	0.022^{+}
PKN	0.7406	0.739	-0.002^{L}	0.010^{+}	0.398	0.433	0.035^{W}	0.000^{+}
DBC	0.0197	0.324	0.304^{W}	0.000^{+}	0.04	0.779	0.739^{W}	0.000^{+}
WINE	0.3542	0.758	0.404^{W}	0.000^{+}	0.724	0.64	-0.083^{L}	0.000^{+}
LDR	0.4734	0.468	-0.005^{L}	0.202^{-}	0.553	0.59	0.036^{W}	0.464^{-}
AVG. W/T/L	0.443	0.630	4/0/3		0.549	0.674	6/0/1	
+/-				6/1				6/1

	FDR				ACC			
	DCA_ OR	**DCA_ KM**	**Diff**	**p_val**	**DCA_ OR**	**DCA_ KM**	**Diff**	**p_val**
BIO	0.3151	0.279	0.036^{W}	0.001^{+}	0.61	0.769	0.159^{W}	0.000^{+}
GCD	0.2785	0.194	0.084^{W}	0.000^{+}	0.618	0.664	0.045^{W}	0.000^{+}
IRIS	0.2791	0.253	0.026^{W}	0.022^{+}	0.7	0.786	0.086^{W}	0.001^{+}
PKN	0.6018	0.567	0.035^{W}	0.000^{+}	0.483	0.507	0.024^{W}	0.000^{+}
DBC	0.9605	0.221	0.739^{W}	0.000^{+}	0.033	0.62	0.587^{W}	0.000^{+}
WINE	0.2762	0.36	-0.083^{L}	0.000^{+}	0.624	0.672	0.048^{W}	0.000^{+}
LDR	0.4467	0.41	0.036^{W}	0.464^{-}	0.507	0.519	0.012^{W}	0.029^{+}
AVG. W/T/L	0.451	0.326	6/0/1		0.511	0.648	7/0/0	
+/-				6/1				7/0

From Table 9, the overall results indicate that there is a clear improvement in performance after the integration of DCA and k-means. In the case of the BIO dataset, the DR, SPS, FDR, and ACC in DCA_KM demonstrate a significant improvement in comparison with the DCA_OR. This result is also similar to other datasets IRIS, DBN, and WINE, where the DCA_KM generates a significant improvement for all evaluation metrics. Moreover, the DCA_KM seems to have better ability in discriminating both normal and abnormal items after it has been integrated with k-means. For example, in the case of the DBC dataset, the original DCA fails to classify normal items well because a high FDR is recorded, but the result improves significantly after integration with k-means. The results for SPS and ACC in the case of the DCA_KM also demonstrate a similar pattern. The

W/T/L analysis is in line with this result because it can be seen that the DCA_KM has the highest winning score over the unordered DCA in all measurements. In terms of significance, the DCA_KM exhibits a significant difference in most datasets when p_val <0.05, mainly in terms of ACC.

The results generated for every evaluation metric are different in that if the algorithm has a good score for DR and ACC it performs less well in terms of SPS and FDR. This causes difficulty in determining whether the proposed model has performed well overall in the case of certain datasets. For a model to be a good detection model, it must have the ability to generate a balanced result in terms of DR, FDR, and SPS when detecting anomalies [14]. Therefore, to simplify the evaluation of the effectiveness of the proposed model, a preference matrix approach is implemented to calculate the accumulative score of each of the performance metrics [15]. The score is given based on the priority of each metric; highest (DR, SPS, ACC) or lowest priority (FDR). A score of 1 is given for the best mining result and a score of 2 for the worst result according to the priority metric. The ∑ Score DCA_OR column and ∑ Score DCA_KM column of the preference matrix represent the total scores of all priorities. The lowest score indicates the best model.

Table 10 contains the preference matrix for BIO, GCD, IRIS, PKN, WINE, and LDR. From the table, the proposed DCA_KM has the lowest accumulative score for all datasets except GCD and IRIS. This indicates the applicability of the proposed model that integrates k-means with DCA. In the case of the IRIS dataset, both DCA_KM and DCA_OR have as similar ability such that DCA_KM has better DR and ACC while DCA_OR has good SPS and FDR.

Table 10: The preference matrix.

	DCA_OR				**∑ Score DCA_OR**	**DCA_KM**				**∑ Score DCA_KM**	**Best Model (DCA..)**
	DR	**SPS**	**FDR**	**ACC**		**DR**	**SPS**	**FDR**	**ACC**		
BIO	2	2	2	2	8	1	1	1	1	4	_KM
GCD	2	1	1	1	5	1	2	2	2	7	_OR
IRIS	2	1	1	2	6	1	2	2	1	6	=
PKN	2	2	2	2	8	1	1	1	1	4	_KM
DBC	2	3	3	1	9	1	1	1	1	4	_KM
WINE	2	2	2	2	8	1	1	1	1	4	_KM
LDR	2	2	2	2	8	1	1	1	1	4	_KM

6 Discussion

The aim of using k-means is to group items with similar characteristics into one cluster instead of resorting to sorting by decision class. In this process, each item in the cluster is sorted according to its distance from the cluster centre to make them have a time-dependent relation with each other. This sorting of the items allows the DCA to easily detect the change of item context. The results of the experiment on the benchmark dataset (WBC) show that the integration of DCA with k-means can transform unordered data into sequential data effectively because fewer false detection errors are generated than original DCA. The result of DCA_KM over unordered WBC is comparable with the DCA_OR, which was initially sorted according to decision class. Also, the results achieved by the proposed model are comparable to those of MMDCA; however, MMDCA still generates a more balanced result between DR and FDR. The main difference between the DCA_KM and the MMDCA is that the MMDCA handles the unordered data during the mining phase while the clustered DCA deals with it by k-means during data preparation, i.e. before mining starts.

Moreover, a similar result was achieved when the DCA_KM was applied to several general classification datasets. For example, in the case of certain datasets such BIO and IRIS, the performance of initial clustering leads to a significant improvement in the DCA detection result. Based on an evaluation of four separate evaluation metrics (DR, SPS, FDR, and ACC), the DCA_KM outperformed the DCA_OR in terms of SPS and FDR while ACC in certain datasets. However, if we consider all four evaluation metrics together, the DCA_KM performs significantly better than DCA_OR. The proposed clustered DCA will produces better detection accuracy when there are fewer clusters and the best number of clusters is two. A possible explanation for this finding is that the datasets used in this study were set to have two decision classes, normal and abnormal.

Furthermore, DCA performance depends on how good the clustering algorithm is at clustering the data into similar groups because it is only after that step that all the items can be arranged into the appropriate sequence. Therefore, any rules that are applied to achieve a good cluster must be considered carefully before clustering the data for the proposed DCA. The information in Table 11 shows the relationship between the state of the art k-means algorithm and DCA_KM based on classification accuracy. Apart from the GCD and IRIS datasets, all the results demonstrate that the DCA_KM requires a good cluster result in order to generate better DR, SPS, FDR, and ACC.

Table 11: The relationship between the accurateness of k-means and the DCA_KM.

	K-means	DCA_KM			
	ACC	DR	SPS	FDR	ACC
BIO	0.959	0.855	0.721	0.279	0.769
GCD	0.708	0.332	0.806	0.194	0.664
IRIS	0.325	0.936	0.747	0.253	0.786
PKN	0.687	0.739	0.433	0.567	0.507
DBC	0.687	0.324	0.779	0.221	0.62
WINE	0.660	0.758	0.64	0.36	0.672
LDR	0.567	0.468	0.59	0.41	0.519

7 Conclusion

In this study, the k-means clustering algorithm is integrated with the DCA with the aim of creating a DCA-based model that can mine unordered datasets as well as it can mine ordered datasets. In the proposed model, k-means is integrated into a standard DCA as a part of data preparation phase before the dataset is normalized into appropriate DCA signals. In the experiments performed on eight universal datasets, the proposed DCA with k-means was found to be more effective than original DCA. In the case of the benchmark dataset (WBC), the DCA_KM demonstrated a significant improvement in comparison to DCA_OR and achieved a comparable result to that of MMDCA. In the case of the seven other datasets tested, based on an evaluation of four separate performance metrics, the DCA_KM outperformed the DCA_OR in terms of SPS and FDR while ACC in certain datasets. Also, considering all the performance metrics together, the proposed model achieved the highest accumulative score for five of the seven datasets. The findings indicate that the k-means output can be utilized to transform an unordered dataset into sequential data by sorting the items according item–centroid distance, thus fulfilling the DCA requirement that data be organized in an orderly event-driven manner. To further evaluate the effectiveness of the proposed approach, further analysis will be conducted on different types of clustering algorithm such c-means clustering, hierarchical clustering, and density-based clustering. Furthermore, the proposed clustered DCA will also be tested by applying it to time series datasets.

References

[1] J. Greensmith, U. Aickelin, and S. Cayzer, "Introducing Dendritic Cells as a Novel Immune Inspired Algorithm for Anomaly Detection " in *4tn International Conference in Artificial Immune Systems (ICARIS)*, 2005, pp. 153-167.

[2] J. Greensmith, J. Twycross, and U. Aickelin, "Dendritic Cells for Anomaly Detection," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, 2006, pp. 664-671.

[3] C.-M. Ou, "Host-based intrusion detection systems adapted from agent-based artificial immune systems," *Neurocomputing,* vol. 88, pp. 78-86, 2012.

[4] B. Ran, J. Timmis, and A. Tyrrell, "The Diagnostic Dendritic Cell Algorithm for robotic systems," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, 2010, pp. 1-8.

[5] R. Huang, H. Taufik, and A. K. Nagar, "Artificial Dendritic Cells Algorithm for Online Break-in Fraud Detection," presented at the Second International Conference on Developement in eSystems Engineering, 2009.

[6] M. F. Mohamad Mohsin, A. R. Hamdan, and A. Abu Bakar, "The Preliminary Design of Outbreak Detection Model Based on Inspired Immune System," in *Third World Congress on Information and Communication Technologies (WICT 2013)*, Hanoi, Vietnam, 2013.

[7] Z. Chelly and Z. Elouedi, "FDCM: A Fuzzy Dendritic Cell Method," in *Artificial Immune Systems*. vol. 6209, E. Hart, C. McEwan, J. Timmis, and A. Hone, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 102-115.

[8] J. Greensmith, "The Dendritic Cell Algorithm," PhD, University of Nottingham, 2007.

[9] Y. Song and C. Qijuan, "Dendritic Cell Algorithm for Anomaly Detection in Unordered Data Set," in *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*, 2012, pp. 249-252.

[10] P. M. Murphy. (1997, 2 January 2013). *UCI repositories of machine learning and domain theories"* Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[11] StatLib. (2005, 3 Febuary 2014). *"Statlib — datasets archive"* Available: http://lib.stat.cmu/datasets

[12] J. Greensmith, U. Aickelin, and G. Tedesco, "Information fusion for anomaly detection with the dendritic cell algorithm," *Information Fusion,* vol. 11, pp. 21-34, 2010.

[13] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations,," in *5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, 1967, pp. 281-297.

[14] M. F. Mohamad Mohsin, A. R. Hamdan, and A. Abu Bakar, "The Effect of Normalization for Real Value Negative Selection Algorithm," in *Soft Computing Applications and Intelligent Systems*. vol. 378, S. Noah, A.

Abdullah, H. Arshad, A. Abu Bakar, Z. Othman, S. Sahran, N. Omar, and Z. Othman, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 194-205.

[15] L. Al Shalabi and Z. Shaaban, "Normalization as a Preprocessing Engine for Data Mining and the Approach of Preference Matrix," in *International Conference on Dependability of Computer Systems, 2006. DepCos-RELCOMEX '06.* , 2006, pp. 207-214.