Anomaly Detection in the Internet of Things (IoT) by Using Artificial Immune System

Noe Elisa[1], Longzhi Yang[1], Fei Chao[2], Nitin Naik[3],

[1] Department of Computer and Information Sciences, Northumbria University,
Newcastle upon Tyne, NE1 8ST, UK
{noe.elisa, longzhi.yang}@northumbria.ac.uk
[2] Department of AI, Xiamen University, Xiamen, China, fchao@xmu.edu.cn
[3] School of Informatics and Digital Engineering, Aston University
Birmingham, B4 7ET, UK, n.naik1@aston.ac.uk

Abstract. Internet of Things (IoT) have demonstrated significant impact on all aspects of human daily lives due to their pervasive applications in areas such as telehealth, home appliances, surveillance, and wearable devices. The number of IoT devices and sensors connected to the Internet across the world is expected to reach over 50 billion by the end of 2020. However, the connection of such rapidly increasing number of IoT devices to the Internet leads to concerns in cyber-attacks such as malware, worms, denial of service attack (DoS) and distributed DoS attack (DDoS). To prevent these attacks from compromising the performance of IoT devices, various approaches for detecting and mitigating cyber security threats have been developed. This paper reports an IoT attack and anomaly detection approach using the dendritic cell algorithm (DCA). In particular, DCA is an artificial immune system (AIS), which is developed from the inspiration of the working principles and characteristic behaviours of the human immune system, specifically for the purpose of detecting anomalies in computer networks. The performance of the DCA on detecting IoT attacks is evaluated using publicly available IoT datasets involving five attacks including DoS, DDoS, Reconnaissance, Keylogging, and Data exfiltration. The experimental results show that, the DCA achieved better detection performance compared to some of the commonly used classifiers, such as the decision trees, random forests, support vector machines, artificial neural networks and naïve Bayes, but with reasonably high computational efficiency.

Keywords: IoT, artificial immune systems, dendritic cell algorithm, anomaly detection, cyber-attacks.

1 Introduction

The demand and deployment of IoT automated networks have been increasing significantly in the past years [1, 2]. IoT sensors and actuators are deployed

This work has been supported by the Commonwealth Scholarship Commission (CSC-TZCS-2017-717), the Royal Academy of Engineering (IAPP1\100077), and Mr. Aminu Abulmalik who contributed to data processing and part of the experimentation under the Royal Academy of Engineering project.

in various places, such as in industry, health monitoring systems, battlefield, weather, transportation system etc, for monitoring, reporting, and activating different events for timely and informed decision making [2]. IoT devices can misbehave due to cyber-attacks or even due to breakdown of the system itself. As IoT networks expand, attacks and anomalies in IoT networks increase significantly [3]. Typical cyber attacks to business websites, e-Government [4, 5], and internet devices [6], such as malware, keyloggers, network scan, spying, DoS, DDoS, Ramsonware, are more and more commonly appeared in IoT networks, which can cause serious damages to IoT services and applications.

Many machine learning approaches ranging from supervised, unsupervised to semi-supervised algorithms have been well exploited to develop IoT intrusion detection models with promising performance generated [1, 7]. One of the methods that was developed for the purpose of anomaly detection in computer network is AIS algorithms. This study develops an intrusion detection and mitigation approach for IoT networks by using AIS. One of the robust, effective and recently proposed AIS algorithm for anomaly detection is the DCA algorithm [8], and this algorithm has been used in this study to develop the anomaly detection approach for IoT networks. The development of AIS is mainly inspired by the human immune system.

The DCA is a mathematical representation of the danger theory of human immune system (HIS), which state that the HIS is concerned with things that might cause damage to the human body and things that might not [9]. Thus, in HIS, the recognition of invaders is performed by natural dendritic cells (DCs) [10]. In HIS, any harmful substance is said to be associated with three signals namely pathogenic associated molecular pattern (PAMP), danger signals (DS) and safe signals (SS); and hence, the DCs sample these signals and present them to HIS for a specialised immune response such as elimination or tolerance. In fact, the DCA has been applied to detect attacks in computer networks with promising performances [11–13].

The DCA goes through four phases to perform anomaly detection including feature selection and signal categorisation, context detection of data samples, context assignment, and finally classification of data samples [8]. Briefly, feature selection is applied with the DCA to select the most informative attributes based on a training dataset. Then, the selected features are categorised into three signals of either PAMP, Danger Signal (DS) or Safe Singal (SS) depending on their characteristic behaviour abstracted from the natural immune signals. Then, the DCA initialises a number of artificial DCs which use a weighted function to determine the context of each data sample from the three signals. Since each DC process multiple data instances, during the context assignment phase, each DC assigns all its data samples to the context it has found either as normal or anomaly. During the classification phase, the final class of each data instance is determined from a consensus decision reached by multiples DCs which have sampled the data instance.

In this study, feature selection was performed by using the information gain method [14], then, mutual information method was used to categorise each feature to its appropriate signal category by maximising mutual information in normal and anomaly class label presented in the dataset. Particles swarm optimisation algorithm (PSO) [15] was used to generate and optimise the weights required by the DCs to process the signals of data samples during the context detection phase of the DCA. The PSO was selected due to its ability to achieve a good balance between exploitation and exploration of search space, implying faster performance compared to other optimisation methods, such as genetic algorithm, which has also been applied to optimise the DCA [12,16,17].

The performance of the proposed approach was evaluated by using publicly available BoT_IoT dataset which is comprised of five attacks including DoS, DDoS, Reconnaissance and Keylogging and Data_Exfiltration [7]. The performance of the DCA method was compared with five state-of-the-art machine learning classifiers including Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Artificial Neural Network (ANN) and Naïve Bayes (NB). The result shows that, the DCA method achieved better performance compared to SVM, ANN and NB; whilst producing the comparable performance to that of DT and RF.

The rest of this paper is structured as follows. Section 2 introduces the theoretical underpinnings of the IoT, biological DCs and the DCA algorithm. Section 3 presents the DCA-based approach used in this study. Section 4 details the experimentation process and finally, Section 5 concludes this study and suggests probable future works.

2 Background

This section presents the theoretical underpinnings of IoT, biological DCs and the DCA algorithm.

2.1 Internet of Things

IoT technologies comprise of automated devices that are networked together to share information via the Internet [18]. IoT is one of the recently fastest growing technologies, with the number of IoT devices and sensors connected to the Internet across the world expected to reach over 50 billion by the end of the year 2020 [18]. Three elements make up a typical IoT based infrastructure: identification/data capture, processing, and communication. Thus, devices collect data, process and send them to the server and return the results through the Internet. The frequently generated data from IoT devices is collected at a centralised server for storage and analytics [1]. There are several application domain in which IoT systems exist such as smart cities, logistics, healthcare, energy and home automation [1].

Like any other networked systems, IoT networks are facing many cyber security threats [1–3]. Devices, data and network are some of the ways in which attacks can be launched against IoT infrastructure. Attacks in IoT networks include malware, keyloggers, scanner, spyware, DoS, and DDoS [1]. A compromised IoT system can result into unavailability of network in its entirety or slow response due to overwhelmed request sent by attackers. The consequences of attacks in IoT can be devastating causing a substantial loss in finance and putting lives in danger such as in logistics and healthcare [1, 3].

Several techniques exist for detecting intrusion and attacks in IoT. Signature based detection compares network activities to a database of suspected attacks and whenever there is a match an alert is raised [3]. This approach clearly fails when a new attack pattern is discovered and also requires that the database is often updated with new attack patterns [17]. In contrast, anomaly based detection compares previous network activities with current activities and raises an alarm when a deviation is detected. Thus, anomaly based detection is able to detect novel patterns such as zero day attacks. Machine learning techniques such as SVM, ANN, DT, RF, NB and etc, have been exploited to develop models for anomaly detection for IoT [1]. However, these models suffer a number of limitations such as high rate of false positives, high running costs and scalability [3].

2.2 Biological DCs

In natural HIS, the DCs are available in the body tissues such as skin, nose and lung where they act as the first line of defense against foreign invaders [10]. DCs are responsible for capturing antigens such as bacteria and virus or anything identified as harmful and presenting them to the adaptive immune system for a specialised immune response. DCs express co-stimulatory molecules and cytokines on their own cell surface which limit the amount of time spent while gathering antigens in the tissue before they migrate to the adaptive immune system for presentation. In the adaptive immune system, T-cells are responsible for destroying the invaders presented by DCs, including any cells infected by a virus or bacteria [9]. DCs are sensitive to the concentrations of the following three signals in HIS [9]:

- **PAMP** are abnormal proteins produced by viruses or bacteria which can easily activate immune response.
- **DS** are released from the disrupted or stressed cells in the tissue which indicates an anomalous situation but with lower score than the PAMP.
- **SS** are produced by normal cell death process in the tissue, which is an indicator of normal cell behavior.

Generally, DCs exist in three states in HIS [9] depending on the concentrations of SS, PAMP or DS signals in the tissue as follows:

1. **Immature DCs (iDCs):** are found in tissues in their pure state where they still collect antigens (i.e.; normal proteins or anything foreign). The concentration of the signals of the collected antigens causes iDC to move to a full-mature or semi-mature state.
2. **Full-mature DCs (mDCs):** iDCs are transformed to mDCs when they are exposed to a greater quantity of either PAMP or DS than SS which causes immune reaction.
3. **Semi-mature DCs (smDCs):** iDCS are transformed to smDCs when they are exposed to more SS than PAMP and DS which causes immune tolerance.

2.3 Dendritic Cell Algorithm

DCA is a population based intrusion detection system where a population of artificial DCs is created to form a pool from which a number of DCs are selected to perform data sampling, context analysis and classification [8]. DCs in the pool are exposed to current signal values and the corresponding data items included in the data source [11]. Each DC has the ability to sample multiple data items. During the classification, an aggregated sampling value from different DCs for a particular data item is computed which is used to classify a data item as normal or anomalous.

Normally, feature selection process is first applied with the DCA. Then, the selected features are categorised into three input signals of either "PAMP", "DS" or "SS". Briefly, PAMP indicates an definite abnormality associated with a particular feature. DS represents an abnormality associated with a feature but with lower score than PAMP; and SS indicates a normality behaviour associated with a feature. There are two common signal categorisation techniques used with the DCA in the literature, including the manual approach by relying on the expert knowledge of the problem domain [8, 19], and the automatic methods such as PCA [13], fuzzy-rough set theory [20], GA shuffle mutation [21] or fuzzy inference systems [22, 23]. In addition, the input features can also be aggregated into the three signals using inference approaches [24].

After signal categorisation process, the DCA algorithm initialises a population of artificial DCs (often 100) in a pool [8]. Then, the three input signals are processed by a pre-selected number of DCs (often 10) in order to get three output signals namely co-stimulatory signal (csm), mature signal (mDC) and semi-mature signal ($smDC$) by applying:

$$Context[csm, smDC, mDC] = \sum_{d=1}^{m} \frac{\sum_{i,j=1,1}^{3}(c_j * w_i^j)}{\sum_{i,j=1,1}^{3} w_i^j}, \qquad (1)$$

where $c_j (j = 1, 2, 3)$, represent the PAMP, DS and SS signal values respectively; and $w_i^j (i, j = 1, 2, 3)$ represent the weights of csm, mDC and $smDC$ context, regarding PAMP, DS and SS, respectively. The weights are usually either predefined or derived empirically from the dataset [20]. Each DC is assigned a migration threshold in order to determine the lifespan that it spends while sampling data items and signals from the data source. Note that, DCs accumulate the values of csm, $mDCS$ and $smDC$ overtime for multiple data items they sample to obtain the cumulative values.

As soon as the cumulative csm value of a DC exceeds its assigned migration threshold, it ceases sampling and moves to the mature pool. Then, the DC compares the values of cumulative mDC and cumulative $smDC$ to determine the nature of the sampled data items. If cumulative $smDC$ is greater than cumulative mDC, the DC differentiate to semi-mature context, implying that the data items collected are under normal condition. Otherwise, the DCs goes to mature context, which implies that the data items sampled are potentially anomalous.

Note that, since one data item is usually sampled by multiple DCs, the final classification label of each data item is determined from the number of DCs that are fully matured, and it is represented by the mature context antigen value (MCAV). The MCAV is used to evaluate the degree of anomaly of each sampled data item by multiple DCs. The MCAV is determined by dividing the number of times a data sample is presented in mature context, by the total number of presentation by multiples DCs that have sampled it. The closer the MCAV is to 1.0, the higher the probability that the data item is anomalous. Thus, an anomaly threshold is computed from the training dataset by taking the percentage of anomalous data samples. Those data items whose MCAV are greater than the anomaly threshold are classified into the anomalous class, otherwise into the normal one.

3 Anomaly detection in IoT Networks Using DCA

The proposed system for detecting anomalies and attacks in IoT networks by using the DCA algorithm is illustrated in Figure 1. Firstly, feature selection process is applied to a training dataset to select the most informative features. Secondly, the selected features are categorised into three signals of either PAMP, DS or SS based on their definitions derived from the biological metaphor [8]. Then, the PSO algorithm is exploited to generate and optimise the weights used by the DCA in its context detection phase. The last two phases of the proposed system are exactly the same as that in the conventional DCA, and thus the rest of this section only focuses on feature selection, signal categorisation and context detection phases of the DCA.

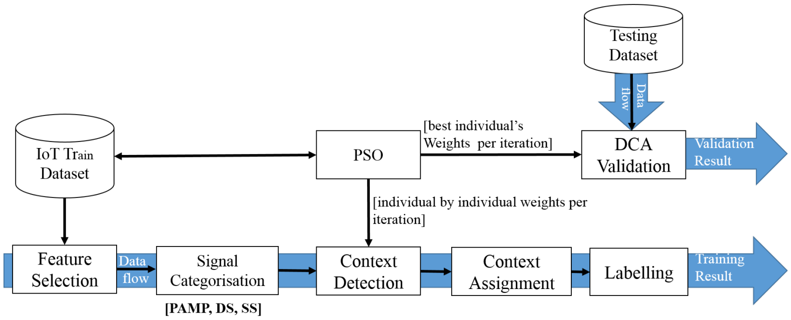

Fig. 1. The proposed DCA-based anomaly detection system in IoT networks

3.1 Feature Selection

The information gain method is firstly applied to select the most informative features from the IoT dataset due to its efficiency and effectiveness [14]. Features with high information gain are retained in the dataset as they have stronger influence in classifying the present data samples. Briefly, given a dataset S, the information gain of an attribute F can be evaluated by using [14]:

$$G(S,F) = H(S) - \sum_{v \in values(F)} \frac{|S_v|}{|S|} * H(S_v), \tag{2}$$

where $values(F)$ represents the whole set of potential values that attribute F may take, S_v is a subset of S each having value v for attribute F, G is the information gain, and H is the entropy. In particular, the entropy H is computed as:

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

where p_i is the probability of class i in the dataset S based on the values of attribute F. The higher the entropy is, the higher the information the corresponding attribute provides. A threshold is set so that only attributes with higher information gains than the threshold are retained in the dataset for signal categorisation.

3.2 Signal Categorisation

In order to find a relevant subset of features for either PAMP, DS or SS, signal categorisation is performed by maximising the feature-class mutual information in this study. The mutual information, $I(F;C)$ between two random features F and C is the amount of information that C gives about F. $I(F;C)$ is calculated using:

$$I(F;C) = \sum_{f \in values(F), c \in values(C)} p(f,c) * log(\frac{p(f,c)}{p(f)p(c)}), \tag{4}$$

where $p(f,c)$ is the joint probability of values of f and c being taken, $p(f)$ and $p(c)$ are the marginal probability of attributes values f and c being taken respectively.

Note that, If an attribute has a higher mutual information with the normal class and significant lower mutual information with the anomalous class, it is assigned to SS; if an attribute has higher mutual information with the anomalous class but significant lower mutual information with the normal class, it is assigned to PAMP; otherwise, the feature is assigned to DS.

3.3 Weights Generation and Optimisation Using PSO

PSO is a population based stochastic optimisation technique insipired by the behaviour of bird flocking [15,25]. Usually, PSO starts by initialising a population of random solutions (i.e. particles) and search for optimal solution by updating the velocity and position of each particle while iterating over a search space over a number of iterations. In PSO, diversification of solutions lies on the velocity of particles, direction of particles and the best regions in the search space.

Typically, the initialised particles in PSO algorithm move through search space by following the best particle in the current iteration. Thus, two best values are used to update the position of particles in every iteration. The first one is the fitness value that a particle has achieved so far, called personal best (*pbest*). The second is called the global best value (*gbest*), which is the fitness value of the best particle in the current iteration.

After evaluating the best values, each particle update its velocity ($V_j(t+1)$) and position ($Z_j(t+1)$) by using:

$$V_j(t+1) = \omega V_j(t) + c_1 r_1[pbest - Z_j(t)] + c_2 r_2[gbest - Z_i(t)], \tag{5}$$

$$Z_j(t+1) = Z_j(t) + V_j(t+1), \tag{6}$$

where j is the index of a particle; ω is the inertia coefficient of the PSO; c_1 and c_2 are particle's acceleration coefficients ($0 \leq c_1, c_2 \leq 2$); r_1 and r_2 are random values ($0 \leq r_1$, $r_2 \leq 1$) which are regenerated every time the velocity of a particle is updated; $V_j(t)$ is the velocity of a particle at time t; $Z_j(t)$ is the position of a particle at time t; *pbest* is the particle's individual best position at time t; *gbest* is the swarm's best particle in the current iteration at time t.

Each of the PSO parameters used in this work is detailed in the following main steps.

1) Particle representation: In this work, a particle (P) within a swarm is a designated solution that comprises of all the parameters of DCA's Equations 1. Therefore, an individual is represented as $I = \{w^1_{smDC}, w^2_{smDC}, w^3_{smDC}, w^1_{mDC}, w^2_{mDC}, w^3_{mDC}, w^1_{csm}, w^2_{csm}, w^3_{csm}\}$, where 1,2,3 represent the three signals categories extracted from the selected features during the pre-processing step.

2) Initialisation of particle's parameters: The swarm $\mathbb{S} = \{P_1, P_2, ..., P_M\}$ is initialised with random numbers from a Gaussian distribution with a mean of 0 and a standard deviation of 5. Where, M is the total number of particles in the swarm, with 10-30 being widely used [15, 25].

3) Objective function: In this study, the objective function of the PSO is equal to the classification accuracy computed by the DCA for each particle.

4) Acceleration coefficients: The acceleration coefficients c_1 and c_2, and the random values r_1 and r_2, control the collaboration among particles and the best particle in the swarm, as well as stochastic influence on the overall velocity of a particle. Low values for c_1 and c_2 allow the particles to visit the search space far from the best regions before being pulled back towards the best solutions [15], while, r_1 and r_2 are randomly updated every time the velocity is calculated.

5) Inertia component: The ω helps to maintain the steady movement of particles in the same direction. Smaller value of ω accelerate convergence while larger value encourages exploration of the search space, and is usually set between 0.8 and 1.2 [15].

6) neighborhood size: The neighborhood size defines the degree of particles interaction within the swarm. The larger the neighborhood size the faster the convergence although the PSO will be more susceptible to local optimal solutions. The smaller the neighborhood size, the slower the convergence but the PSO will be more reliable to converge to global optimal solutions.

7) Iteration and termination: In this study, the PSO terminates when the classification performance of the DCA exceeds the pre-specified threshold of the maximum optimum accuracy or the pre-defined maximum number of iterations is attained.

4 Experimentation

This section details the experimentation process and validation of the results as well as the comparative evaluation of the DCA-based approach with five of the state-of-the-art machine learning classifiers. All experiments were implemented by using JAVA in NetBeans IDE 8.2. Then, the performance comparison were performed by using an HP workstation with Intel$^{®}$ Xeon^{TM} E5-16030 v4 CPU @3.70 GHz and 32GB RAM.

4.1 Benchmark Datasets

The dataset named BoT_IoT [7] from Cyber Range Lab of the University of New South Wales, Australia was used to evaluate the performance of the proposed approach. The BoT_IoT dataset was created by designing a realistic IoT network environment while incorporating legitimate and simulated IoT network traffic, along with five types of attacks including DDoS, DoS, Reconnaissance, Keylogging and Data exfiltration attacks (i.e., Information Theft) [7]. The dataset contains 7336090 samples and 17 features. Each data sample within the dataset is labelled as either normal or anomaly.

Dataset Pre-processing: The IG method and mutual information maximisation were used for feature selection and signal categorisation respectively. Each feature was normalised using the min-max normalisation.

DCs Initialisation and Sampling: The size of DCs population in the pool was initialised to 100, then, in each DCA cycle, 10 matured DCs were used to process the data samples and signals in the mature pool. The migration thresholds of DCs were initialised in a Gaussian distribution with a mean of 5.0 and standard deviation of 1.0. The anomaly threshold of each dataset was computed by taking the ratio of the total number of anomaly class's samples to the total number of samples present in the dataset.

PSO Parameters: The parameter values used for the PSO include 250 number of iterations, 20 individuals in a swarm, $c_1 = c_2 = 2.0$, $0 \leq r_1$, $r_2 \leq 1$, initial velocity $V_0 = 0$, maximum velocity $V_{max} = 10$ and $\omega = 0.95$, as these values have been found to be most suitable for the PSO [15].

Measurement Metrics The performance of the DCA-based approach was measured in terms of accuracy and detection rate of each attack type present in the dataset.

Accuracy and detection rate are defined as follows:

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+TN+FP+FN}, \\ Detection\_Rate &= \frac{TP}{TP+FN} \end{aligned} \tag{7}$$

where TP, FP, TN, and FN refer respectively to true positive, false positive, true negative and false negative, respectively.

4.2 Results and Analysis

The testing results on accuracy and detection rate for each attack type are summarised in Table 1 regarding the proposed DCA-based approach and four widely used classifiers including SVM, ANN, DT, RF and NB. The best performance amongst these approaches on overall accuracy and each attack type is highlighted in bold.

Table 1. Performance results and comparison

Approach	Overall Accuracy (%)	Attacks detection rate(%)			
		DoS	DDoS	Reconn	Info Theft
DCA	97.96	**99.38**	**98.24**	97.73	73.74
SVM	72.65	90.49	37.12	92.84	25.32
DT	95.92	96.94	95.95	93.94	89.67
RF	**97.97**	97.95	96.98	**100.00**	97.47
ANN	89.15	96.67	85.42	84.24	0.0
NB	66.10	92.38	95.18	2.68	**98.73**

From the results displayed in Table 1, it can be noticed that, the DCA-based approach has produced higher detection rates on DoS and DDoS attacks compared to the state-of-the-art classifiers. The classification accuracy of the DCA is almost the same as the best performing accuracy of RF, thus, it has best overall detection performances compared to other classifiers. For instance, RF has better detection rate on reconnaissance attack but less effective on DoS and DDoS and information theft when compared to the DCA. Same remark is noticed for NB classifier where it has produced better detection rate on information theft attack but it has poor detection performance on other attacks and the overall classification accuracy. Ultimately, it can be concluded that, the DCA algorithm is effective on detecting anomalies and attacks in IoT networks with better detection performances compared to some best state-of-the-art machine learning classifiers.

Furthermore, the performance comparison in terms of running time for the DCA and the compared classifiers is illustrated in Figure 2. Although DCA takes much more time to train in comparison to RF and DT, taking its detection performances in account, DCA is more appropriate and reasonable for mitigating anomalies in IoT networks. Generally, the main goal is to make sure that, when the DCA is deployed to detect attacks in IoT networks, it is capable of producing higher and satisfactory detection results in comparison to the referenced commonly used classifiers.

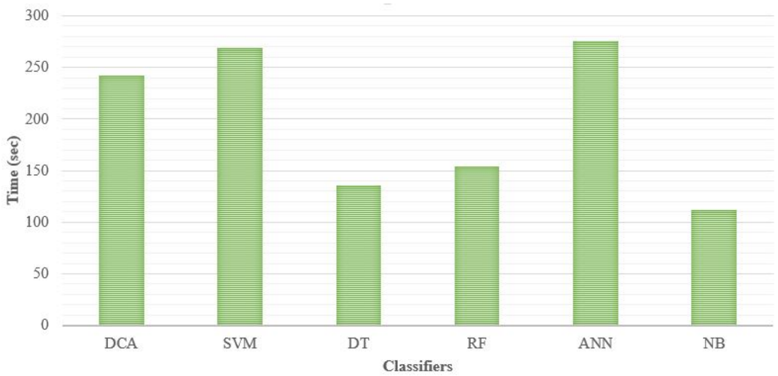

Fig. 2. The running time (sec) for classifiers

5 Conclusions

This work presented an approach for detecting anomalies and attacks in IoT networks by using the DCA algorithm. The performance of the proposed approach was evaluated by using publicly available IoT datasets which include five attacks DoS, DDoS, Reconnaissance and Keylogging and Data_Exfiltration. The DCA-based approach achieved better performance compared to some of the commonly used classifiers such as SVM, DT, RF, ANN and NB. Although promising, future work can improve its performance. It is interesting to further explore how the DCA-based approach can be integrated with privacy-preserving technologies such as the blockchain technology and cryptography to develop a privacy-preserving system for IoT networks in order to reinforce information privacy and security.

References

1. Bruno Bogaz Zarpelão, Rodrigo Sanches Miani, Cláudio Toshio Kawakani, and Sean Carlisto de Alvarenga. A survey of intrusion detection in internet of things.

Journal of Network and Computer Applications, 84:25–37, 2017.

2. Partha Pratim Ray. A survey on internet of things architectures. *Journal of King Saud University-Computer and Information Sciences*, 30(3):291–319, 2018.
3. Alma Oracevic, Selma Dilek, and Suat Ozdemir. Security in internet of things: A survey. In *2017 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6. IEEE, 2017.
4. Noe Elisa, Longzhi Yang, Fei Chao, and Yi Cao. A framework of blockchain-based secure and privacy-preserving e-government system. *Wireless Networks*, pages 1–11, 2018.
5. Longzhi Yang, Noe Elisa, and Neil Eliot. Privacy and security aspects of e-government in smart cities. In *Smart Cities Cybersecurity and Privacy*, pages 89–102. Elsevier, 2019.
6. N. Naik, P. Jenkins, B. Kerby, J. Sloane, and L. Yang. Fuzzy logic aided intelligent threat detection in cisco adaptive security appliance 5500 series firewalls. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8, 2018.
7. Nickolaos Koroniotis, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, 100:779–796, 2019.
8. Julie Greensmith, Uwe Aickelin, and Steve Cayzer. Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. In *International Conference on Artificial Immune Systems*, pages 153–167. Springer, 2005.
9. Polly Matzinger. Essay 1: the danger model in its historical context. *Scandinavian journal of immunology*, 54(1-2):4–9, 2001.
10. Jacques Banchereau and Ralph M Steinman. Dendritic cells and the control of immunity. *Nature*, 392(6673):245, 1998.
11. Zeineb Chelly and Zied Elouedi. A survey of the dendritic cell algorithm. *Knowledge and Information Systems*, 48(3):505–535, 2016.
12. Noe Elisa, Longzhi Yang, Xin Fu, and Nitin Naik. Dendritic cell algorithm enhancement using fuzzy inference system for network intrusion detection. In *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6. IEEE, 2019.
13. Feng Gu. *Theoretical and empirical extensions of the dendritic cell algorithm.* PhD thesis, University of Nottingham, 2011.
14. Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2016.
15. Andries P Engelbrecht and Christopher Wesley Cleghorn. Recent advances in particle swarm optimization analysis and understanding. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pages 747–774, 2020.
16. N. Elisa, L. Yang, F. Chao, and N. Naik. A comparative study of genetic algorithm and particle swarm optimisation for dendritic cell algorithm. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2020.
17. Noe Elisa, Longzhi Yang, and Nitin Naik. Dendritic cell algorithm with optimised parameters using genetic algorithm. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2018.
18. Sandip Ray, Yier Jin, and Arijit Raychowdhury. The changing computing paradigm with internet of things: A tutorial introduction. *IEEE Design & Test*, 33(2):76–96, 2016.
19. Noe Elisa, Longzhi Yang, Yanpeng Qu, and Fei Chao. A revised dendritic cell algorithm using k-means clustering. In *2018 IEEE 20th International Conference*

on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pages 1547–1554. IEEE, 2018.

20. 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.
21. Noe Elisa, Longzhi Yang, and Fei Chao. Signal categorisation for dendritic cell algorithm using ga with partial shuffle mutation. In *UK Workshop on Computational Intelligence*, pages 529–540. Springer, 2019.
22. Longzhi Yang, Fei Chao, and Qiang Shen. Generalised adaptive fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, 25(4):839–853, 2017.
23. Longzhi Yang and Qiang Shen. Closed form fuzzy interpolation. *Fuzzy Sets and Systems*, 225:1–22, 2013.
24. Noe Elisa, Jie Li, Zheming Zuo, and Longzhi Yang. Dendritic cell algorithm with fuzzy inference system for input signal generation. In *UK workshop on computational intelligence*, pages 203–214. Springer, 2018.
25. James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.