DeepDCA: Novel Network-Based Detection of IoT Attacks Using Artificial Immune System

Sahar Aldhaheri 1,*, Daniyal Alghazzawi 1, Li Cheng 2, Bander Alzahrani 1 and Abdullah Al-Barakati 1

1 Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; dgazzawi@kau.edu.sa (D.A.); baalzahrani@kau.edu.sa (B.A.); aaalbarakati@kau.edu.sa (A.A.-B.)
2 Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Sciences, Urumqi 830011, China; chengli@ms.xjb.ac.cn
* Correspondence: saldhaheri0002@stu.kau.edu.sa

Received: 2 February 2020; Accepted: 18 February 2020; Published: 11 March 2020

Abstract: Recently Internet of Things (IoT) attains tremendous popularity, although this promising technology leads to a variety of security obstacles. The conventional solutions do not suit the new dilemmas brought by the IoT ecosystem. Conversely, Artificial Immune Systems (AIS) is intelligent and adaptive systems mimic the human immune system which holds desirable properties for such a dynamic environment and provides an opportunity to improve IoT security. In this work, we develop a novel hybrid Deep Learning and Dendritic Cell Algorithm (DeepDCA) in the context of an Intrusion Detection System (IDS). The framework adopts Dendritic Cell Algorithm (DCA) and Self Normalizing Neural Network (SNN). The aim of this research is to classify IoT intrusion and minimize the false alarm generation. Also, automate and smooth the signal extraction phase which improves the classification performance. The proposed IDS selects the convenient set of features from the IoT-Bot dataset, performs signal categorization using the SNN then use the DCA for classification. The experimentation results show that DeepDCA performed well in detecting the IoT attacks with a high detection rate demonstrating over 98.73% accuracy and low false-positive rate. Also, we compared these results with State-of-the-art techniques, which showed that our model is capable of performing better classification tasks than SVM, NB, KNN, and MLP. We plan to carry out further experiments to verify the framework using a more challenging dataset and make further comparisons with other signal extraction approaches. Also, involve in real-time (online) attack detection.

Keywords: artificial intelligence; artificial immune system; cyber security; danger theory; deep learning; dendritic cell; internet of things; IoT; network security

1. Introduction

Recently in the academic and industrial circles, Internet of Things (IoT) have became an active area. According to Cisco, 500 billion devices will be connected by the year 2030 [1]. Although this technology is promising in many sectors, such as such as smart homes, health-care, intelligent transportation, power smart grid and numerous areas that not yet even conceived [2], it carries with it many security risks. Easy accessibility and tremendous propagation of IoT devices creates a fertile environment for cyber attacks. Most of these devices are small, inexpensive and have limited memory and computing capacity to run the current existing security software [3]. Additionally, the Original Equipment Manufacturers (OEMs) are using commercial embedded Real-Time Operating Systems (RTOS), such as FreeRTOS and OpenRTOS to minimize the cost [4] which makes these end devices vulnerable to be targeted. As stated by a report for Malwarebytes, the IoT attacks will continue at steady levels with increased sophistication [5]. For example, recent malware such as Mirai [6] and Ransomware of Things (RoT) [7,8]
proven that conventional security methods are ineffective and do not provide decentralized and strong security solutions. In addition to the urgent need for a new paradigm of security commensurate with the changes that have emerged with the IoT ecosystem. Where the security problems inherited from the traditional network alongside the Advanced Persistent Threat (APT). Therefore, to dissolve the obstacles security of IoT we must look at solutions from a comprehensive perspective and take into account new circumstances and requirements. Lately, multiple solutions have been applied to secure the IoT environment and guarantee security requirements like authentication, availability, integrity, confidentiality and privacy [9].

From another aspect, the Artificial Immune System (AIS) is a bionic intelligent system that mimics the biological immune system and its way to protect against foreign or dangerous invaders [10]. AIS has proven effective in protecting TCP/IP networks [11,12], Wireless Sensor Networks (WSN) [13,14], and Mobile Ad Hoc Network (MANET) [15,16]. This makes it more suitable for a dynamic and changeable environment such as IoT. Moreover, the human immune system properties make it a perfect approach to resolve IoT security dilemmas. Due to its ability to self-learning, adaptability robustness, resource optimization, dynamic structure, and lightweight [17] make it adapted to various applications such as computer security [18], intrusion detection [19,20], anomaly detection [21], data analysis [22,23], pattern recognition [24] and scheduling [25,26]. Over and above, AIS methods solve multi-objective optimization problems successfully [27,28], control engineering [29,30] and robotics [31].

After extensive IoT attacks, we reviewed the proposed recommendations and solutions to avoid infection. The suggested solutions were generally summed up to change the default passwords for the IoT devices, disable some ports, and guide consumers and manufacturers to use more secure devices. Although these security practices are effective and provide the first line of defense, their application is limited to security management and human interaction. Another challenge in the case of known software vulnerabilities is the delay of download the patches. Under these conditions, intrusion detection techniques become more important. Thus, the motivation for this study is that the traditional detection approaches are not able to efficiently detect new variants of IoT attacks. Consequently, it is urgent to study intrusion detection approaches in depth. Wherefore, the immune-based detection methods consider as a priority option due to its desirable properties.

The artificial immune system has various algorithms detect different types of attacks. The second generation of these algorithms called Dendritic Cell Algorithm (DCA). Greensmith introduced this novel danger-based AIS to detect port-scan attacks over wired networks [32]. DCA is inspired by the capability of DCs to receive multiple antigens and signals, as well as reveal the context of each antigen. A novel DeepDCA is introduced in this study by sing an AIS inspired algorithm promises to address the challenges of IoT environment that make it vulnerable to attacks. The DeepDCA is verified and tested in this study to detect DoS, DDoS, Information gathering and theft. DeepDCA can be generalized to detect other types of attacks on IoT.

In this paper, we propose a novel Deep Learning and Dendritic Cell Algorithm based IDS framework; named DeepDCA. To identify IoT intrusion and minimize the false alarm generation. Our contributions can be summarized as follows:

- Design a novel IDS composite of Self Normalizing Neural Network (SNN) for signals categorization with Dendritic Cell Algorithm (DCA)
- Introduce the concept of Self Normalizing Neural Network in the DCA signal extraction phase to search for the convenient features, reduce the complexity and automate this phase while preserving excellent performance.
- Implement the proposed framework on IoT dataset and evaluate its performance based on a variety of IDS performance metrics.
- Use an IoT dataset instead of the out-dated KDD Cup 99 dataset.

The rest of this paper is organized as follows. Section 2 provides a brief background of recent large-scale attacks targeting IoT devices and existing IoT security Approaches. Section 3 describes the
dendritic cell algorithm generally and the basic concepts of Self Normalizing Neural Network. Related work for the AIS algorithms in the IoT ecosystem summarizes in Section 4. While Section 5 presents the proposed method based on DCA and SNN for signal selection and categorization. Followed by the evaluation results of the proposed IDS in Section 5. Finally, Section 6 concludes the paper.

2. Background

2.1. Intrusion Detection Systems

Many technologies designed to protect the internet from destruction, breaches and unauthorized access. However, there are many defense technologies designed to protect this environment. IDSs are one of the essential parts that aim to monitor, analyze the network traffic and detect attacks. According to Hernández-Pereira, Elena, et al. [33] Intrusion can be defined as “any set of actions that attempt to compromise the Confidentiality, Integrity, and Availability (CIA) of information resources.” Typical IDS have an analysis engine, sensors, and a reporting system. The sensors collect network, and host data then send it to the analysis engine. Hence, the analysis engine investigates the collected data and detect intrusions. If intrusion exists, the network administrator receives an alert from the reporting system [34].

Intrusion Detection Systems Architecture

The architecture of the IDSs functional modules divided into four types: [35] (see Figure 1)

- **Event-boxes (E blocks)**: these blocks monitor the target system by sensor elements and acquire the information events.
- **Database-boxes (D blocks)**: these blocks store information from E blocks.
- **Analysis-boxes (A blocks)**: these blocks analyze events and detect potential abnormal behavior.
- **Response-boxes (R blocks)**: these blocks execute in case an intrusion occurs.

![Figure 1. General architecture for IDS systems.](image)

2.2. IoT Security Overview

The dynamic characteristics of the Internet of things and the desire to make devices connected anywhere, anytime, and anyplace creates critical challenges about privacy and security. Researchers from the HP lab addressed that almost 70% of IoT devices are vulnerable to be targeted, which means 25 vulnerabilities per device [36]. The vulnerabilities around privacy, lack of encryption standards, authentication/authorization. Additionally, the threats and security problems inherited from the traditional network [37]. As shown in Figure 2, the attacks classified on the IoT three layers: perception, network, and application-layer [38].
2.3. Existing IoT Security Approaches

2.3.1. Non Artificial Intelligence-Based Security Method

Many of non-artificial intelligence-based security methods have been implemented to secure the IoT environment. A good example of these methods: identity-based encryption, watchdog, reputation and trust mechanisms, Complex Event Processing (CEP) and lightweight cryptography. Table 1 summarize the presented techniques.

| Ref | Year | Techniques | Metrics | IoT Layer | Simulator | Category |
|-----|------|------------|---------|-----------|-----------|----------|
| [39] | 2019 | Generic honeypot framework by utilizing VPN connections | Detection rate, low cost and maintenance effort | Applications Layer | Simulation Data | Honeypot |
| [40] | 2016 | Identity-based encryption | Efficiency (overhead cost associated with computation and communication) | Applications Layer | Simulation Data | Lightweight cryptography (LWC) |
| [41] | 2015 | Watchdog, reputation and trust mechanisms | Detection rate, false positives and false negatives. | Network Layer | Simulator Cooja | Intrusion detection system |
| [42] | 2014 | External entity as the ISP or Security as a service (SECaas or SaaS) provider, to install access control rules in the network | ...... | Network Layer | Captured the network activity using the Wireshark packet analyzer. | Emerging household appliances |
| [43] | 2014 | Complex event processing (CEP) | CPU utilization, memory consumption and processing time | Applications Layer | Esper an engine for CEP and event series analysis. | Real-time intrusion |
| [44] | 2013 | Dynamic variable cipher security certificate “one time one cipher” | Repetition rate | Sensor Layer | Simulation Data | Lightweight cryptography (LWC) |
| [45] | 2013 | Rule-based detection | Detection rate, false positives | Network Layer | Demo | Intrusion detection system |
| [46] | 2013 | End-to-End message security such as IPsec and DTLS | Detection rate, True positives rate energy and memory consumption | Network Layer | Contiki’s network simulator Cooja | Real-time intrusion |
2.3.2. Artificial Intelligence-Based Security Method

Artificial intelligence has attracted attention in recent years, especially in the field of IoT security. The following Table 2 summarizes the leading technologies that have been introduced to protect the IoT ecosystem.

Table 2. Artificial Intelligence-based Security Methods (FS = Features Selection).

| No | Year | Techniques | Dataset | Metrics | IoT Layer | FS | Category |
|----|------|------------|---------|---------|-----------|----|----------|
| [47] | 2019 | C5 and One Class Support Vector Machine classifier | Bot-IoT dataset | Detection rate and False positive | Network Layer | Yes | Intrusion Detection System |
| [48] | 2019 | Multilayer Perceptron (MLP) | ADFA-LD and ADFA-WD | Accuracy, Recall, and F1 | Network Layer | No | Intrusion Detection System |
| [49] | 2018 | Recurrent Neural Network (RNN) + Convolutional Neural Network (CNN) | RedIRIS | Accuracy, Detection rates, and False Positive Rate | Network Layer | No | A network traffic classifier (NTC) |
| [50] | 2018 | Deep-learning (DAE + Deep Feed Forward Neural Network (DFNN)) | NSL-KDD and UNSW-NB15 | Accuracy, Precision, Recall and F1 | Network Layer | Yes | Intrusion Detection System |
| [51] | 2017 | Deep Neural Network + Grid Search Strategy | UNSW-NB15, CIDDS-001, and GPRS. | Accuracy, Precision, Recall and F1 | Transport Layer | No | Anomaly Detection |
| [52] | 2017 | Convolutional Neural Networks (CNN) | IoTPOT | Detection rates, Accuracy | Network Layer | No | Light-weight Detection |
| [53] | 2017 | Recurrent Neural Network (RNN) | ISCX Detection and the Dataset CTU-13 | Sensitivity, Specificity, Confusion Matrix, Accuracy and F1 Score | Applications Layer | No | Botnet Detection |
| [54] | 2016 | Stacked Auto Encoder (SAE) Deep learning | KDD99 | Detection rates, and False Positive Rate | Transport Layer | Yes | Intrusion Detection System |
| [55] | 2016 | artificial neural network (ANN) | Collect the data in Testbed | Detection rates, and False Positive Rate | Network Layer | No | Anomaly Detection |
| [56] | 2016 | Artificial Neural Network (ANN) | Simulated network. | Accuracy, and False Alarm Rate. | Applications Layer | No | Threat Analysis |
| [57] | 2016 | Machine Learning + Security as a Service | ...... | Accuracy, and False Alarm Rate | Transport Layer | No | Integrated Intrusion Detection System |
| [58] | 2013 | Hormone-based Service Detection Algorithm (HSDA) | Simulated network | Stability of the network, Saving energy and Detection rates | Network Layer | No | Anomaly Detection |
3. Dendritic Cell Algorithm

Algorithm Overview

In 1994, Pollu Matzinger introduced the danger theory and described the immune mechanism through the danger signals activation when damage exists [59,60]. It also states that in the absence of tissue-related danger signals, the innate immune mechanism will be suppressed [61]. This process derived from the cell death process (apoptosis and necrosis). The dendritic cell algorithm (DCA), presented by Green-Smith et al. which considered as de facto danger theory algorithm. DCA goes through four phases as detailed below [62]:

- **Phase 1. Pre-processing and initialization**: this phase includes two main steps: feature reduction and signal categorization. First, feature reduction which selects the most important attributes from the training set. Next, the selected features classify to signal category: safe, danger and PAMP.
- **Phase 2. Detection**: in this phase, the DCA has to generate a signal database by combining the input signals with the antigens to obtain cumulative output signals.
- **Phase 3. Context Assessment**: the generation of cumulative output signals from the detection phase are used to perform context assessment of antigens. If the collected antigens by a DC has a greater Mature DCs (mDC) than its Semi-Mature DCs (smDC) value, it is labeled as 1, otherwise 0.
- **Phase 4. Classification**: the calculated value deriving from the Mature Context Antigen Value (MCAV) for each antigen is used to assess the degree of the anomaly. When the value of MCAV is closer to 1, the antigen probability of been anomalous is higher. The MCAV of antigen is calculated by dividing the number of times an antigen appears in the mature context by the total number of that antigen presentation. When the MCAV is calculated, the classification task starts by comparing the MCAV of each antigen to an anomalous threshold. Antigens with MCAVs greater than the anomaly threshold are classified into the abnormal otherwise are classified into normal.

4. Self-Normalizing Neural Networks

Self-normalizing neural networks are introduced in 2017 by Gnter Klambauer [63]. It is a higher-level abstraction neural networks where the neuron activations automatically concentrate on a fixed mean and variance. Unlike other neural networks algorithms that lack the ability to normalizing the outputs and need further layers such as batch normalization [64]

4.1. SELU Activations

The activation function proposed in SNN is *Scaled exponential linear units* (SELU). It is similar to the *Rectified Linear Units* (ReLU) but with a simple exponential function. The SELU activation function is defined as:

\[
\text{sel}(x) = \lambda \begin{cases} 
  x & \text{if } x > 0 \\
  \alpha e^x - \alpha & \text{if } x \leq 0 
\end{cases}
\]

where \(x\) denotes input \(\alpha (\alpha = 1.6733), \lambda (\lambda = 1.0507)\) are hyper parameters which control the mean and variance of the output distribution.

4.2. Alpha Dropout

Ordinarily, the neurons are dropout in a random way by setting his weight to zero with probability \(1 - p\). In doing this the network is prevented to set mean and variance to an expected value. The ReLUs works very well with the standard dropout for the following reason: zero goes down to the low variance region which is the default value. In the case of SELU, we have that the default low variance is given by \(\lim_{x \to \infty} \text{sel}(x) = -\lambda \alpha = \alpha t\) and for this reason the standard dropout does not fill well. Then for sets that the input values randomly to \(\alpha t\), alpha dropout is the proposed to fit them well. The original values
of mean and variance are restored by alpha dropout and the self-normalizing property is preserved too. Therefore by making activation into negative values saturation at random alpha dropout suits SELU.

5. Related Work: AIS and IoT

The using of AIS approaches to secure the IoT started in 2010, in this section, we will address the AIS methods that have been used to secure the IoT area based on the IoT layers.

5.1. Sense Layer

Many solutions based on AIS have been applied to secure the physical layer communication. The work of Chmielewski and Brzozowski [65], presented a “support system for existing solutions” embedded in a re-programmable FPGA (Field Programmable Gate Array). This model based on hybrid negative selection algorithm, called b-v model to detect the zero-day attacks. Besides, Chen et al. [66] investigated and computed the intensity value of security threats faced by IoT. They addressed a theoretical security situation sense model. This model consists of a security threat sense sub-model (STS) and a security situation assessment sub-model (SSA). This work introduced a notable mathematical theoretical model but this would be more interesting if it describes how to apply it within IoT and what type of data could be used.

5.2. Network Layer

So far, most of the AIS- based studies have been carried out in the network layer to handle the IoT security. A signature-based IDS proposed by Liu et al. [67]. This IDS contains memory detectors that simulate the antigens in the human body and classify datagrams as normal and malicious. In spite of that theory mathematically analyzed and detected a various number of intrusions, it has a high computational running and the researchers did not specify how to implement it in limited resources devices. Additionally, a dynamic approach called Artificial Immune System Response Model (AISRM) was produced by Liu et al. [68]. The proposed model captures the IoT data packets and transforms them into immune antigens then detects and responds to attacks. Although this is an adaptable model proven through a simulation experiment, the central server scalability is a significant problem where all communication passes through.

5.3. Application Layer

For the application layer, smart homes represented the majority. In this context Arrignton et al. [69] proposed a Behavioral Intrusion Detection System based on positive and negative selection algorithms. This work provides an important insight into the process of detecting abnormal behavior related to non-playing characters such as a human. Nonetheless, due to the expanding of IoT network, this would delays the performance and leads to consume the resources. And conversely, in order to reduce the cost and time and provide the optimal solution, Yang et al. [70] developed a multi-objective optimization model.

6. DeepDCA: Deep Learning Dendritic Cell Algorithm

This Section presents the DeepDCA model for the automate DCA data pre-processing phase. As shown in Figure 3, the framework consists of three main steps, namely: Features Selection, Signals Categorization, and Deterministic Dendritic Cell algorithm. The proposed approach functions will mainly focus on the pre-processing phase.
6.1. Feature Selection

This framework adopted the Information Gain (IG) approach to decide which features are more important. The $IG(F)$ is a measure of the reduction in entropy of variable $F$ that is archived by learning after the value for the feature is observed. In Data Science the information gain used for ranking the features. A feature with high information gain ranked higher than others and has a strong power in the classification process. The $IG$ can be obtained by [71]:

$$IG(S) = E(F) - \sum_{v \in \text{values}(S)} \frac{|F_v|}{|F|} \cdot E(F_v)$$

(2)

where $IG$ is the gain, $values(S)$ is symbolize all the possible values of an attribute $S$. Moreover, $F_v$ is a subset generated by partitioning $S$ based on feature $F$, and $E(F)$ is the entropy which computed as the following:

$$\text{Entropy} (F) = \sum_{i=1}^{i=2} -p_i \cdot \log_2 p_i$$

(3)

6.2. The SNN Signal Categorization

The SNN module assign each selected attribute to specific signal category (see Figure 4). The guidelines for signal categorization are presented below:

- **Danger Signal**: this signal indicates to the presence of anomalous situation or attack circumstances
- **Safe Signal**: this signal indicates to the presence of normal behavior or non-attack circumstances
Generation of DCA Signals

The SNN is designed to extract the signals as safe (SS) and danger (DS) signal from the features \(f_1, f_2, \ldots, f_N\) as the following parametrized:

\[
SS \left( f_1, f_2, \ldots, f_N \right) = 100 \times \text{sigmoid} \left( b_S + \sum_i w_{S,i} \ast \text{elu} \left( b_{S,i} + v_{S,i} \ast f_i \right) \right)
\]

(4)

\[
SD \left( f_1, f_2, \ldots, f_N \right) = 100 \times \text{sigmoid} \left( b_D + \sum_i w_{D,i} \ast \text{elu} \left( b_{D,i} + v_{D,i} \ast f_i \right) \right)
\]

(5)

where

\[
w_{S,i} \text{ and } w_{D,i} \geq 0
\]

(6)

The \textit{sigmoid} activation function at the output neurons assures that the signals are contained in the range 0–100. Where the \textit{elu} activation function at the hidden layer neurons admits to cut off high or low values of the feature attributes—depending on the signs of the parameters \(v_{S,i}\) or \(v_{D,i}\). The positivity restriction on the weights \(w_{S,i}\) and \(w_{D,i}\) breaks the symmetry between the formulae for the two signals. It prohibits the SNN from choosing a solution where \(w_{S,i} = -w_{D,i}\). These steps illustrate using the Algorithm 1.
Algorithm 1 SSN for Signal Categorization.

**Input:** Features \((f_1, f_2, ..., f_N)\);

**Output:** Signals as safe (SS) and danger (DS)

1. initialise number of hidden layers \(L\), weights \(w\);
2. for \(i\) in Number of inputs do
   3. Add input \(f_i\);
   4. Add neuron \(H_{S,i}\) SeLU activation & connect with \(f_i\);
   5. Add neuron \(H_{D,i}\) SeLU activation & connect with \(f_i\);
   6. Add AlphaDropout Layer for the SS;
   7. Connect it with \(H_{S,i}, i = 1, ..., \text{Number of inputs}\);
   8. Add AlphaDropout Layer for DS;
   9. Connect it with \(H_{D,i}, i = 1, ..., \text{Number of inputs}\);
10. Add neuron \(S\) with sigmoid activation;
11. Connect it with the AlphaDropout Layer for SS;
12. Add neuron \(D\) with sigmoid activation
13. Connect it with the AlphaDropout Layer for DS;
14. end for
15. while no of epochs not complete & condition not fulfilled do
   16. Update weights;
   17. Compute training and validation loss;
   18. Evaluate model performance;
19. end while

6.3. Signal Processing

The combined signals to produce the intermediate output values of \(K\) and \(csm\). The value \(K\) is a measure of the anomaly or irregularity in the cell, by other hands, the \(csm\) value represents the concentration of the complete signal that a cell exposes in all its useful life. When the cell depletes its shelf life will migrate and will be ready to classify all of the antigens collected in his past useful life, at this time produce the classification as normal or abnormal. The addition of safe signals with the danger signals gives the value \(csm\). Therefore, the value \(K\) is obtained subtracting of the danger signals twice the safe signal. The following equation gives the values:

\[
K_i = DS_i - 2SS_i. \tag{7}
\]

6.3.1. Costimulation (CSM)

The generated signals from the SSN module combine to produce two intermediate output values \(CSM\) and \(K\). By Costimulation we mean the process of cumulative concentration of signals within its environment by a DC in a period of time of his life. For a DC in the moment that his life span expires, it immediately migrates to the lymph node and exhibits antigens in certain circumstances. With the following equation the calculation of the value \(csm\) is performed:

\[
csm_i = SS_i + DS_i, \tag{8}
\]

where \(S\) and \(D\) are the input value for the safe and danger signals.

6.3.2. Lifespan

By mean of the term lifespan of a DC we signify the total time that a DC takes to collect all of the signal concentration on its environment previous to the migration to the lymph node. When the value of lifespan results in less than the sum of the concentration the lifespan of the DC stops of subtracting
the accumulated concentration of signals over time. Thus, the value of lifespan is a fixed quantity, but overtime this value is decreasing as the following Equation (9) assures—where $i = 1, \ldots, N$:

$$lifespan = lifespanSS_i + DS_i$$

(9)

6.4. Anomaly Metrics: MCAV and $K_\alpha$

Once all data are processed it is possible to calculate the metric MCAV, the mature context antigen value obtained from the output of the cell that comes out from the run-time process. The value is calculated for each antigen of type $\alpha$, where the symbol $\alpha$ is associated with a collection of antigens that has in common the same value. Clearly we could think by its name that MCAV is indeed a measure of the proportion of antigen contained in a completely mature cell whose value is given by the following equation:

$$MCAV_\alpha = \frac{M}{Ag}$$

(10)

where $MCAV_\alpha$ represents the antigen MCAV of the collection $\alpha$, $M$ is the number of the mature antigen of type $\alpha$, and $Ag$ is the total quantity of antigen presented for the collection of an antigen of type $\alpha$. This is a probabilistic metric with values between zero and one, when the value of this metric goes to one, the probability of maturity of the cell increase. The classification rule applied on as follows in Equation (11) and the deterministic DCA could be described by mean of the Algorithm 2.

$$f(x) = \begin{cases} 
\text{Malicious,} & \text{if } MCAV > at \\
\text{Legitimate,} & \text{otherwise}
\end{cases}$$

(11)
Algorithm 2 Deterministic DCA for Intrusion Detection.

**Input** Antigens and Signals;

**Output** Antigens Types and accumulative k values

1: set number of cells;
2: initialise DCs()
3: while data do
4:   for input do
5:     if Antigens then
6:       antigenCounterC++;
7:       cell index = antigen counter modulus cells number;
8:       DC of cell index assigned antigen;
9:       update DCs antigen profile;
10:   else
11:     calculate csm and k;
12:     for all DCs do
13:       DC.lifspan- = csm;
14:       DC.k+ = k;
15:       if DC.lifespan less or equal to then
16:         logDC.k, number of antigens and iterations
17:       reset DC();
18:     end if
19:   end for
20: end if
21: end for
22: end while
23: for antigen Type do
24:   calculate anomaly metrics;
25: end for

7. Experimental Setup

To conduct this experiment, we performed it on the High-Performance Computing (HPC) called Aziz. Aziz is a Fujitsu PRIMERGY CX400, Intel True Scale QDR, Intel Xeon E5-2695v2 12C 2.4GHz which provides a distributed computing facility. Moreover, for data exploration and visualization we used ggplot framework [72] and Seaborn [73]. For preprocessing steps and feature engineering, Pandas framework [74] and Numpy framework [75] have been used. To calculate performance metrics, scikit-learn [76] was used, and finally, for data analysis, scikit-learn framework and Keras [77] were used. We followed the Cross-Industry Standard Process (CRISP) methodology [78]. CRISP is a structured methodology for Data Mining projects conceived in 1996. which contain the following steps: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment.

7.1. Data Acquisition

To illustrate the effectiveness of our model we selected The BoT-IoT dataset [79]. This data was created in the Cyber Range Lab of The center of UNSW Canberra Cyber and has more than 72,000,000 records which include DDoS, DoS, OS and Service Scan, Keylogging and Data exfiltration attacks. Table 3 illustrate the statistics distribution of considered features.
7.2. Exploratory Data Analysis (Understanding the Data)

Exploratory Data Analysis (EDA) is “the process of examining a dataset without preconceived assumptions about the data and its behavior” [80]. The goal of the EDA is to evaluate the cleanliness and missing data and explore the relationships among variables which give us a deep insight (see Table 4).

| Name          | Value      |
|---------------|------------|
| Rows          | 3,668,045  |
| Columns       | 47         |
| Discrete columns | 7          |
| Continuous columns | 40        |
| All missing columns | 0         |
| Missing observations | 0         |
| Complete Rows | 3,668,045  |
| Memory allocation | 13.6 Mb  |

7.3. Preparation Steps of Bot-IoT Dataset

Handling Categorical Variables

The categorical feature values (‘saddr’, ‘sport’, ‘daddr’, ‘dport’, ‘state’, ‘category’, and ‘subcategory’) have converted into numeric values for easily applying feature selection method and the DeepDCA algorithm.

| Feature                     | Attack Type  | Frequency Count | Training Count | Test Count  |
|-----------------------------|--------------|-----------------|----------------|-------------|
| Normal                      | Non          | 3,668,045       | 2,567,631      | 1,100,413   |
| Information Theft           | Keylogging   | 73              | 51             | 22          |
|                             | Data theft   | 6               | 4              | 2           |
| Information gathering       | Service      | 73,168          | 51,217         | 21,951      |
|                             | Fingerprinting | 17,914         | 12,540         | 5374        |
| DoS                         | DoS TCP      | 615,800         | 431,060        | 184,740     |
|                             | DoS UDP      | 1,032,975       | 723,082        | 30,9893     |
|                             | DoS HTTP     | 1485            | 1039           | 446         |
| DDoS                        | DDoS TCP     | 977,380         | 684,166        | 293,214     |
|                             | DDoS UDP     | 948,255         | 663,778        | 284,477     |
|                             | DDoS HTTP    | 989             | 692            | 297         |
7.4. Feature Selection

Feature selection is a primary step to enhance IDS performance, reduce the computational cost and improve accuracy. In the original dataset, a selection of the 10 best features has been provided (see Table 5). In this work, we used the best 10 features and adopted the Information Gain (IG) approach to decide which features are more important. Figure 5 shows information gain for each feature. The features ‘seq’, ‘DstIP’, ‘srate’, ‘SrcIP’, ‘max’, are the most discriminative attribute. While the rest (‘mean’, ‘stddev’, ‘min’, ‘state_number’, ‘drate’) have small maximum information gain (smaller than 0.5), which little contribute to intrusion detection.

![Figure 5. Features Ranking Based on Information Gain.](image)

| Feature   | Data Type | Description                     |
|-----------|-----------|---------------------------------|
| pkSeqID   | Ordinal   | Row Identifier                  |
| Seq       | Numerical | Argus sequence number           |
| Mean      | Numerical | Average duration of aggregated records |
| Stddev    | Numerical | Standard deviation of aggregated records |
| Min       | Numerical | Minimum duration of aggregated records |
| Max       | Numerical | Maximum duration of aggregated records |
| Srate     | Numerical | Source-to-destination packets per second |
| Drate     | Numerical | Destination-to-source packets per second |
| NINConn PSrcIP | Numerical | Total Number of packets per source IP |
| NINConn PDstIP | Numerical | Total Number of packets per Destination IP |

7.5. DeepDCA-Based Classification

7.5.1. Initialization

In this phase, we initialized the population of DCA with size up to a limit of 100 cells. Then, an array size named antigens set to store antigen per iteration. Finally, initialize the output parameters $K$ and CSMK to zero.
7.5.2. Signals and Antigen

The antigen represented by an attribute of the dataset which identifies the traffic packets uniquely, in our case the antigen is “pkSeqID” attribute. For the Signals, SNN models meant to implement a parametrized signal extraction process for the DCA. It defined to have 1 input layer with six neurons equal to the number of input features, one hidden layer, and 1 output neuron for the binary classification. The Model was trained in 125 epochs. The task of the hidden layer neurons would be to encode the decision for a threshold and transform the input attributes into signals normalized into the interval \([0, 1]\) (hence, sigmoid activation). Therefore, the hidden layer neurons should decide for a sign and threshold for each feature. Then, we used selu activation functions for the hidden layer and an \(\alpha - dropout\) layer between the hidden layer and the output neurons. The output neurons can choose a sign and weight per input signal, and would again yield signals within the interval \([0, 1]\). Figure 6 represents the accuracy and loss of SNN model.

7.5.3. Dendritic Cell Algorithm Module

Once the data pre-processing phase is performed, the model moves to the next stages -as described in Section 3, which are the Signal Processing, the Context Assessment, and the Classification Procedure.

![Figure 6. SNN Model Accuracy and Loss.](image)

7.6. Evaluation Criteria

The confusion matrix is usually used to evaluate the performance of the classification model. The confusion matrix relies on the four terms of True Positive (TP), True Negative (TN), False Negative (FN) and False Positive (FP) [81] as shown in Table 6.

- **TP**: is the number of actual malicious records classified as attacks.
- **FP**: is the number of actual legitimate records classified as attacks.
- **TN**: is the number of actual legitimate records identified as normal,
- **FN**: is the actual anomalous records categorized as normal

Also, we evaluate the performance of DeepDCA model in terms of Accuracy, Precision, Recall, F-measure, and False alarm rate:

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{12}
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{13}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{14}
\]
\[ F - \text{measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \] (15)

Table 6. Confusion Matrix.

| Actual | Predicted |        |        |
|--------|-----------|--------|--------|
|        | Positive  | Negative|        |
| Positive| True Positive | False Negative|        |
| Negative| False Positive | True Negative|        |

8. Result and Analysis

This section presents the results obtained when applying the DeepDCA model for intrusion detection. Several hyper-parameters are examined such as the selected features and the attack types.

8.1. Impact of Features

Table 7 illustrate the influence of the features employed in the learning process. The first three records in the table represent the result of the imbalanced data. We examined all features of the BoT-IoT then the best 10 features, and finally, the selected features using information gain (see Section 7.4). As the accuracy is misleading metrics when dealing with imbalanced classes problem, Recall, Precision F-measure are telling a more truthful story. Precision is a measure of (exactness), recall for (completeness) of a model and the F-measure is a harmonic mean of the two. As shown in Table 7, even though the accuracy for all features of the imbalanced classes gets a better result, the rest of the metrics are not in line with it. The IG-selected features have the best indication of detection performance for imbalanced classes. On the other hand with balanced data, as expected fewer features render better results in general. The results produce slightly worse when added to the full features set. Consequently, DeepDCA yields a better result when dealing with balanced classes and features that have a higher importance in the detection process. For make it easier to compare the result, Figure 7 shows the same results in a different format.

Table 7. DeepDCA performance based on Features Impact among Imbalanced and Balanced Data.

| Features | Accuracy | Precision | Recall | F-Measure |
|----------|----------|-----------|--------|-----------|
| **Im-Data** |          |           |        |           |
| All      | 0.7613   | 0.5091    | 0.7368 | 0.6022    |
| Best 10  | 0.5800   | 0.3200    | 0.6667 | 0.4324    |
| IG       | 0.6550   | 0.7000    | 0.6422 | 0.6699    |
| **Ba-Data** |          |           |        |           |
| All      | 0.8950   | 0.7900    | 0.9836 | 0.8827    |
| Best 10  | 0.9500   | 0.9000    | 1.0000 | 0.9474    |
| IG       | 0.9873   | 0.9917    | 0.9836 | 0.9877    |
8.2. Impact of Attack Scenarios

We evaluated the proposed IDS by measuring the performance metrics in different attack scenarios as shown in Table 8. The results illustrated that DeepDCA performed well in detecting various attack types although its performance was better in DDoS/DoS attacks which may be due to the abundance of data about this attack in the BoT-IoT dataset.

| Attack        | Accuracy | Precision | Recall  | F-Measure |
|---------------|----------|-----------|---------|-----------|
| DoS           | 99.8%    | 99.5%     | 98.53%  | 99.012%   |
| DDoS          | 99.9%    | 100%      | 100%    | 100%      |
| Reconnaissance| 99.10%   | 98.88%    | 98.22%  | 98.54%    |
| Information Theft | 98.56% | 99.01%    | 98.9%   | 98.95%    |

8.3. Comparison with Classifiers

The performance evaluation results of the DeepDCA model are compared with four commonly used methods for intrusion detection, namely the Support-Vector Machines (SVM), Naive Bayes (NB), K Nearest Neighbor (KNN) and Multilayer Perceptron (MLP). The comparison made is in terms of Accuracy, F-measure, Recall/sensitivity and Precision. Table 9 shows that DeepDCA slightly better than MLP and outperformed other classifiers SVM, NB and KNN. To sum up, applying the DeepDCA for the Intrusion detection system was validated against an IoT dataset demonstrating over 98.73% accuracy. It was able to identify successfully different types of attacks and showed good performances in terms of detection rate and false-positive rates.
Table 9. Comparison of Classifiers Performance.

| Classifiers  | Accuracy  | Precision | Recall  | F-Measure |
|--------------|-----------|-----------|---------|-----------|
| KNN          | 91.69%    | 91.99%    | 93.75%  | 94.31%    |
| NB           | 95.00%    | 95.00%    | 94.60%  | 94.30%    |
| SVM          | 96.00%    | 96.80%    | 96.00%  | 97.00%    |
| MLP          | 97.59%    | 97.99%    | 96.00%  | 96.37%    |
| Proposed IDS | 98.73%    | 99.17%    | 98.36%  | 98.77%    |

9. Conclusions

In this research, we develop a Deep Learning Dendritic Cell Algorithm (DeepDCA). Our framework adopts DCA and Self Normalizing Neural Network. The aim of this research is to classify IoT intrusion and minimize the false alarm generation. Also, automate and smoothe the signal extraction phase which improves the classification performance. The proposed IDS selects the convenient set of features from the IoT-Bot dataset and to perform their signal categorization using the SNN. The experimentation results show that our DeepDCA performed well in detecting the IoT attacks with a high detection rate demonstrating over 98.73% accuracy and low false-positive rate. Also, capable of performing better classification tasks than SVM, NB, KNN and MLP classifiers. We plan to carry out further experiments to verify the framework using more challenging datasets with missing and noisy data and make further comparisons with other signal extraction approaches. Also, involve in real-time (online) attack detection.

Author Contributions: Conceptualization, S.A.; Formal analysis, S.A.; Funding acquisition, D.A., B.A. and A.A.-B.; Investigation, S.A.; Methodology, S.A.; Project administration, D.A.; Supervision, D.A., L.C., B.A. and A.A.-B.; Validation, S.A.; Visualization, S.A.; Writing—original draft, S.A.; Writing—review and editing, D.A. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under Grant no. (RG-6-611-39).

Acknowledgments: The authors, therefore, acknowledge with thanks DSR technical and financial support. Also, Computation for the work described in this paper was supported by High-Performance Computing Center Aziz Supercomputer (https://hpc.kau.edu.sa). The authors would like to thank the members of Aziz support team for the continuous support and guidance.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

| Abbreviation | Definition |
|--------------|-----------|
| IG           | Information Gain |
| SS           | Safe Signals |
| DS           | danger Signals |
| NB           | Naive Bayes |
| DC           | Dendritic Cell |
| TP           | True Positive |
| TN           | True Negative |
| FN           | False Negative |
| FP           | False Positive |
| IoT          | Internet of Things |
| AIS          | Artificial Immune Systems |
| CSM          | Costimulation |
| HPC          | High Performance Computing |
| IDS          | Intrusion Detection System |
| SNN          | Self Normalizing Neural Network |
| DCA          | Dendritic Cell Algorithm |
RoT Ransomware32of Things  
APT Advanced Persistent Threat  
EDA Exploratory Data Analysis  
WSN Wireless Sensor Networks  
CIA Confidentiality, Integrity, and Availability  
CEP Complex Event Processing  
LWC Lightweight Cryptography  
mDC Mature DCs  
MCAV Mature Context Antigen Value  
SELU Scaled exponential linear units  
ReLU Rectified Linear Units  
SVM Support-Vector Machines  
KNN K Nearest Neighbor  
smDC Semi-Mature DCs  
RTOS Real-Time Operating Systems  
MANET Mobile Ad Hoc Network  
CRISP Cross-Industry 231 Standard Process  
DeepDCA Deep Learning and Dendritic Cell Algorithm

**References and Notes**

1. Kim, N.; Lee, I.; Zazo, J.; Belei, B. Internet of Things EDITOR. Technical report, 2019.
2. Airehrou, D.; Gutierrez, J.; Ray, S.K. Secure routing for internet of things: A survey. *J. Netw. Comput. Appl.* 2016, 66, 198–213. [CrossRef]
3. Yang, Y.; Wu, L.; Yin, G.; Li, L.; Zhao, H. A Survey on Security and Privacy Issues in Internet-of-Things. *IEEE Internet Things J.* 2017, 4, 1250–1258. [CrossRef]
4. RTE Ltd. FreeRTOS—Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions, 2018.
5. Malwarebytes LABS. 2019 State of Malware. Technical report, 2019; p. 33.
6. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and other botnets. *Computer* 2017, 50, 80–84. [CrossRef]
7. Zahra, A.; Shah, M.A. IoT based ransomware growth rate evaluation and detection using command and control blacklisting. In Proceedings of the ICAC 2017—2017 23rd IEEE International Conference on Automation and Computing: Addressing Global Challenges through Automation and Computing, Huddersfield, UK, 7-8 September 2017; pp. 1–6. [CrossRef]
8. Yaqoob, I.; Ahmed, E.; ur Rehman, M.H.; Ahmed, A.I.A.; Al-garadi, M.A.; Imran, M.; Guizani, M. The rise of ransomware and emerging security challenges in the Internet of Things. *Comput. Netw.* 2017, 129, 444–458. [CrossRef]
9. Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H.; Zhao, W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet Things J.* 2017, 4, 1125–1142. [CrossRef]
10. Timmis, J.; Hone, A.; Stibor, T.; Clark, E. Theoretical advances in artificial immune systems. *Theor. Comput. Sci.* 2008, 403, 11–32. [CrossRef]
11. Seredyński, F.; Bouvry, P. Anomaly detection in TCP/IP networks using immune systems paradigm. *Comput. Commun.* 2007, 30, 740–749. [CrossRef]
12. Kim, J.; Bentley, P. An Artificial Immune Model for Network Intrusion Detection. In Proceedings of the 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT’99), Aachen, Germany, 13–16 September 1999.
13. Saleem, K.; Fisal, N.; Hafizah, S.; Rashid, R.A. An intelligent information security mechanism for the network layer of WSN: BIOSARP. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6694 LNCS, pp. 118–126. [CrossRef]
14. Suthaharan, S. Big data classification. *ACM SIGMETRICS Perform. Eval. Rev.* 2014, 41, 70–73. [CrossRef]
15. Mazhar, N.; Farooq, M. BeeAIS: Artificial immune system security for nature inspired, MANET routing protocol, BeeAdHoc. In Proceedings of the 6th International Conference on Artificial Immune Systems, Santos, Brazil, 26–29 August 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 370–381. [CrossRef]

16. Mazhar, N.; Farooq, M. A sense of danger. In Proceedings of the 10th Annual Conference on Genetic And Evolutionary Computation—GECCO ’08, Atlanta, GA, USA, 12–16 July 2008; ACM Press: New York, NY, USA, 2008; p. 63. [CrossRef]

17. Aldhaheri, S.; Alghazzawi, D.; Cheng, L.; Barnawi, A.; Alzahrani, B.A. Artificial Immune Systems approaches to secure the internet of things: A systematic review of the literature and recommendations for future research. *J. Netw. Comput. App.* 2020, 102537. [CrossRef]

18. Harmer, P.K.; Williams, P.D.; Gunsch, G.H.; Lamont, G.B. An artificial immune system architecture for computer security applications. *IEEE Trans. Evolut. Comput.* 2002, 6, 252–280. [CrossRef]

19. Jungwon, K.; Bentley, P. Towards an artificial immune system for network intrusion detection: An investigation of clonal selection with a negative selection operator. In Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Seoul, Korea, 27–30 May 2001; Volume 2, pp. 1244–1252. [CrossRef]

20. Kim, J.; Bentley, P. An evaluation of negative selection in an artificial immune system for network intrusion detection. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), San Francisco, CA, USA, 7–11 July 2001; pp. 1330–1337. [CrossRef]

21. Greensmith, J.; Ackelien, U.; Cayzer, S. Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection. 2016. [CrossRef]

22. Alves, R.T.; Delgado, M.R.; Lopes, H.S.; Freitas, A.A. An Artificial Immune System for Fuzzy-Rule Induction in Data Mining; Springer: Berlin/Heidelberg, Germany, 2004; pp. 1011–1020. [CrossRef]

23. Freitas, A.A.; Timmis, J. Revisiting the foundations of artificial immune systems for data mining. *IEEE Trans. Evolut. Comput.* 2007, 11, 521–540. [CrossRef]

24. De Castro, L.N.; Timmis, J. Artificial Immune Systems: A Novel Paradigm to Pattern Recognition. *Neural Netw. Pattern Recognit.* 2002, 67–84.

25. Hart, E.; Ross, P.; Nelson, J. Producing robust schedules via an artificial immune system. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), Anchorage, AK, USA, 4–9 May 1998; pp. 464–469. [CrossRef]

26. Engin, O.; Döyen, A. A new approach to solve hybrid flow shop scheduling problems by artificial immune system. *Future Gener. Comput. Syst.* 2004, 20, 1083–1095. [CrossRef]

27. Coello, C.A.; Cortés, N.C. Solving multiobjective optimization problems using an artificial immune system. *Genet. Program. Evolvable Mach.* 2005, 6, 163–190. [CrossRef]

28. Alonso, F.R.; Oliveira, D.Q.; Zambroni De Souza, A.C. Artificial immune systems optimization approach for multiobjective distribution system reconfiguration. *IEEE Trans. Power Syst.* 2015, 30, 840–847. [CrossRef]

29. Huang, X.; Tan, Y.; He, X. An intelligent multifeature statistical approach for the discrimination of driving conditions of a hybrid electric vehicle. *IEEE Trans. Intell. Transp. Syst.* 2011, 12, 453–465. [CrossRef]

30. Gu, S.; Tan, Y.; He, X. Recencyness biased learning for time series forecasting. *Inf. Sci.* 2013, 237, 29–38. [CrossRef]

31. Ishiguro, A.; Kondo, T.; Watanabe, Y.; Shirai, Y.; Uchikawa, Y. Emergent construction of artificial immune networks for autonomous mobile robots. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; Volume 2, pp. 1222–1228. [CrossRef]

32. Greensmith, J. The Dendritic Cell Algorithm. Ph.D. Thesis, University of Nottingham, Nottingham, UK, 2007.

33. Hernández-Pereira, E.; Suárez-Romero, J.A.; Fontenla-Romero, O.; Alonso-Betanzos, A. Conversion methods for symbolic features: A comparison applied to an intrusion detection problem. *Expert Syst. Appl.* 2009, 36, 10612–10617. [CrossRef]

34. Zarpelão, B.B.; Miani, R.S.; Kawakani, C.T.; de Alvarenga, S.C. A survey of intrusion detection in Internet of Things. *J. Netw. Comput. Appl.* 2017, 84, 25–37. [CrossRef]

35. García-Teodoro, P.; Díaz-Verdejo, J.; Maciá-Fernández, G.; Vázquez, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Comput. Secur.* 2009, 28, 18–28. [CrossRef]
36. Kasinathan, P.; Pastrone, C.; Spirito, M.A.; Vinkovits, M. Denial-of-Service detection in 6LoWPAN based Internet of Things. In Proceedings of the International Conference on Wireless and Mobile Computing, Networking and Communications, Lyon, France, 7–9 October 2013; pp. 600–607. [CrossRef]
37. Deogirikar, J.; Vidhate, A. Security attacks in IoT: A survey. In Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017, Palladam, India, 10–11 February 2017; pp. 32–37. [CrossRef]
38. Ivan, C.; Vujic, M.; Husnjak, S. Classification of Security Risks in the IoT Environment. In Proceedings of the 26th DAAAM International Symposium on Intelligent Manufacturing and Automation, Zadar, Croatia, 21–24 October 2015.
39. Tambe, A.; Aung, Y.L.; Sridharan, R.; Ochoa, M.; Tippenhauer, N.O.; Shabtai, A.; Elovici, Y. Detection of threats to IoT devices using scalable VPN-forwarded honeypots. In Proceedings of the CODASPY 2019—9th ACM Conference on Data and Application Security and Privacy, Dallas, TX, USA, 25–27 March 2019; pp. 85–96. [CrossRef]
40. Al Salami, S.; Baek, J.; Salah, K.; Damiani, E. Lightweight encryption for smart home. In Proceedings of the 2016 11th International Conference on Availability, Reliability and Security, ARES 2016, Salzburg, Austria, 31 August–2 September 2016; pp. 382–388. [CrossRef]
41. Cervantes, C.; Poplade, D.; Nogueira, M.; Santos, A. Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015, Ottawa, ON, Canada, 11–15 May 2015; pp. 606–611. [CrossRef]
42. Notra, S.; Siddiqi, M.; Gharakheili, H.H.; Sivaraman, V.; Boreli, R. An experimental study of security and privacy risks with emerging household appliances. In Proceedings of the 2014 IEEE Conference on Communications and Network Security (CNS), San Francisco, CA, USA, 29–31 October 2014; pp. 79–84. [CrossRef]
43. Chen, J.; Chen, C. Design of complex event-processing IDS in internet of things. In Proceedings of the 2014 6th International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2014, Zhangjiajie, China, 10–11 January 2014; pp. 226–229. [CrossRef]
44. Wen, Q.; Dong, X.; Zhang, R. Application of dynamic variable cipher security certificate in Internet of Things. In Proceedings of the 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, IEEE CCIS 2012, Hangzhou, China, 30 October–1 November 2012; Volume 3, pp. 1062–1066. [CrossRef]
45. Kasinathan, P.; Costamagna, G.; Khaleel, H.; Pastrone, C.; Spirito, M.A. DEMO: An IDS framework for internet of things empowered by 6LoWPAN. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security—CCS ’13, Berlin, Germany, 4–8 November 2013; ACM Press: New York, NY, USA, 2013; pp. 1337–1340. [CrossRef]
46. Raza, S.; Wallgren, L.; Voigt, T. SVELTE: Real-time intrusion detection in the Internet of Things. Ad Hoc Netw. 2013, 11, 2661–2674. [CrossRef]
47. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J.; Alazab, A. A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. Electronics 2019, 8, 1210. [CrossRef]
48. Khat, B.S.; Wahab, A.W.B.A.; Idris, M.Y.B.; Hussain, M.A.; Ibrahim, A.A. A lightweight perceptron-based intrusion detection system for fog computing. Appl. Sci. 2019, 9, 178. [CrossRef]
49. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J. Network Traffic Classifier with Convolutional and Recurrent Neural Networks for Internet of Things. IEEE Access 2017, 5, 18042–18050. [CrossRef]
50. AL-Hawawreh, M.; Moustafa, N.; Sitnikova, E. Identification of malicious activities in industrial internet of things based on deep learning models. J. Inf. Secur. Appl. 2018, 41, 1–11. [CrossRef]
51. Tama, B.A.; Rhee, K.H. Attack Classification Analysis of IoT Network via Deep Learning Approach. Res. Briefs Inf. Commun. Technol. Evol. (ReBICTE) 2017. [CrossRef]
52. Su, J.; Vargas, D.V.; Prasad, S.; Sgandurra, D.; Feng, Y.; Sakurai, K. Lightweight Classification of IoT Malware based on Image Recognition. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23–27 July 2018; pp. 664–669.
53. Bansal, A.; Mahapatra, S. A comparative analysis of machine learning techniques for botnet detection. In Proceedings of the 10th International Conference on Security of Information and Networks—SIN ’17, Jaipur, India, 17 July 2017; ACM Press: New York, NY, USA, 2017; pp. 91–98. [CrossRef]
54. Aminanto, M.E.; Kim, K. Deep Learning-based Feature Selection for Intrusion Detection System in Transport Layer. Available online: https://pdfs.semanticscholar.org/bf07/e753401b36662eeec7b8cd6c65cb8cfe31562.pdf (accessed on 23 February 2019).

55. Canedo, J.; Skjellum, A. Using machine learning to secure IoT systems. In Proceedings of the 2016 14th Annual Conference on Privacy, Security and Trust, PST 2016, Auckland, New Zealand, 12–14 December 2016; pp. 219–222. [CrossRef]

56. Hodo, E.; Bellekens, X.; Hamilton, A.; Dubouillé, P.L.; Iorkyase, E.; Tachtatzis, C.; Atkinson, R. Threat analysis of IoT networks using artificial neural network intrusion detection system. In Proceedings of the 2016 International Symposium on Networks, Computers and Communications (ISNCC), Yasmine Hammamet, Tunisia, 11–13 May 2016; pp. 1–6. [CrossRef]

57. Chawla, S.; Thamilarasu, G. Security as a Service: Real-time Intrusion Detection in Internet of Things. In Proceedings of the Fifth Cybersecurity Symposium, CyberSec '18, Coeur d’ Alene, ID, USA, 9–11 April 2018; pp. 12:1–12:4

58. Jin, Y.L.; Ding, Y.S.; Hao, K.R.; Liu, Y.J. Efficient service request detection algorithm based on hormone regulation mechanism in the Internet of things. J. China Univ. Posts Telecommun. 2013, 20, 86–90. [CrossRef]

59. Matzinger, P. Tolerance, Danger, and the Extended Family. Ann. Rev. Immunol. 1994, 12, 991–1045. [CrossRef]

60. Matzinger, P. The danger model: A renewed sense of self. Science 2002, 296, 301–305.

61. Aickelin, U.; Cayzer, S. The Danger Theory and Its Application to Artificial Immune Systems. In Proceedings of the 7th International Conference on Artificial Immune Systems (ICARIS-2008), Phuket, Thailand, 10–13 August 2008; pp. 141–148. [CrossRef]

62. Chelly, Z.; Elouedi, Z. A survey of the dendritic cell algorithm. Knowl. Inf. Syst. 2016, 48, 505–535. [CrossRef]

63. Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-Normalizing Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.

64. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015; Volume 1, pp. 448–456.

65. Chmielewski, A.; Brzozowski, M. Immune approach to the protection of IoT devices. In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, Switzerland, 2016; Volume 10018 LNCS, pp. 75–92. [CrossRef]

66. Liu, C.; Yang, J.; Chen, R.; Zhang, Y.; Zeng, J. Research on immunity-based intrusion detection technology for the Internet of Things. In Proceedings of the 2011 7th International Conference on Natural Computation, ICNC 2011, Shanghai, China, 26–28 July 2011; Volume 1, pp. 212–216. [CrossRef]

67. Liu, C.; Yang, J.; Chen, R.; Zhang, Y.; Zeng, J. Research on immunity-based intrusion detection technology for the Internet of Things. J. Comput. 2013, 8, 3111–3118. [CrossRef]

68. Arrington, B.; Barnett, L.E.; Rufus, R.; Esterline, A. Behavioral modeling intrusion detection system (BMIDS) using internet of things (IoT) behavior-based anomaly detection via immunity-inspired algorithms. In Proceedings of the 2016 25th International Conference on Computer Communications and Networks, ICCCN 2016, Waikoloa, HI, USA, 1–4 August 2016; pp. 1–6. [CrossRef]

69. Yang, Z.; Ding, Y.; Jin, Y.; Hao, K. Immune-Endocrine System Inspired Hierarchical Coevolutionary Multiobjective Optimization Algorithm for IoT Service. IEEE Trans. Cybern. 2018, 50, 164–177. [CrossRef] [PubMed]

70. Kayacik, H.G.; Zincir-Heywood, A.N.; Heywood, M.I. 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, Fredericton, NB, Canada, 12–14 October 2005.

71. Wickham, H.; Winston, C. Create Elegant Data Visualisations Using the Grammar of Graphics. R Package Version 3.1.0. Available online: https://CRAN.R-project.org/package=ggplot2 (accessed on 12 May 2019).

72. Waskom, M.; Botvinnik, O.; Hobson, P.; Cole, J.B.; Halchenko, Y.; Hoyer, S.; Miles, A.; Augspurger, T.; Yarkoni, T.; Megies, T.; et al. Seaborn: Statistical data visualization. 2018.
74. McKinney, W. pandas: A Foundational Python Library for Data Analysis and Statistics. In Proceedings of the Workshop Python for High Performance and Scientific Computing (SC11), Seattle, WA, USA, 18 November 2011.
75. Oliphant, T.E. Guide to NumPy. *Methods* **2010**, *1*, 378. [CrossRef]
76. Pedregosa, F.; Michel, V.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Vanderplas, J.; Cournapeau, D.; Pedregosa, F.; Varoquaux, G.; et al. Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot. Technical report, 2011.
77. van Merriënboer, B.; Bahdanau, D.; Dumoulin, V.; Serdyuk, D.; Warde-Farley, D.; Chorowski, J.; Bengio, Y. Blocks and Fuel: Frameworks for deep learning. 2015.
78. Chapman, P.; Clinton, J.; Kerber, R.; Khabaza, T.; Reinartz, T.; Shearer, C.R.H.; Wirth, R. CRISP-DM 1.0 Step-by-Step Data Mining Guide. 1999.
79. Koroniotsis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
80. Collins, M. *Network Security Through Data Analysis II*; O’Reilly Media: Sebastopol, CA, USA, 2014; p. 347.
81. Lippmann, R.; Haines, J.W.; Fried, D.J.; Korba, J.; Das, K. 1999 DARPA off-line intrusion detection evaluation. *Comput. Netw.* **2000**, *34*, 579–595. [CrossRef]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).