electronics

MDPI

Article

SMO-DNN: Spider Monkey Optimization and Deep Neural Network Hybrid Classifier Model for Intrusion Detection

Neelu Khare [1], Preethi Devan [1], Chiranji Lal Chowdhary [1], Sweta Bhattacharya [1], Geeta Singh [2], Saurabh Singh [3] and Byungun Yoon [3,*]

1 School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, India; neelukh.29@gmail.com (N.K.); preethi.devan17@gmail.com (P.D.); chiranji.lal@vit.ac.in (C.L.C.); sweta.b@vit.ac.in (S.B.)

2 School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, India; geeta.singh@vit.ac.in

3 Department of Industrial & Systems Engineering, College of Engineering, Dongguk University, Seoul 04620, Korea; saurabh89@dongguk.edu

* Correspondence: postman3@dongguk.edu

Received: 19 March 2020; Accepted: 21 April 2020; Published: 24 April 2020

Abstract: The enormous growth in internet usage has led to the development of different malicious software posing serious threats to computer security. The various computational activities carried out over the network have huge chances to be tampered and manipulated and this necessitates the emergence of efficient intrusion detection systems. The network attacks are also dynamic in nature, something which increases the importance of developing appropriate models for classification and predictions. Machine learning (ML) and deep learning algorithms have been prevalent choices in the analysis of intrusion detection systems (IDS) datasets. The issues pertaining to quality and quality of data and the handling of high dimensional data is managed by the use of nature inspired algorithms. The present study uses a NSL-KDD and KDD Cup 99 dataset collected from the Kaggle repository. The dataset was cleansed using the min-max normalization technique and passed through the 1-N encoding method for achieving homogeneity. A spider monkey optimization (SMO) algorithm was used for dimensionality reduction and the reduced dataset was fed into a deep neural network (DNN). The SMO based DNN model generated classification results with 99.4% and 92% accuracy, 99.5%and 92.7% of precision, 99.5% and 92.8% of recall and 99.6%and 92.7% of F1-score, utilizing minimal training time. The model was further compared with principal component analysis (PCA)-based DNN and the classical DNN models, wherein the results justified the advantage of implementing the proposed model over other approaches.

Keywords: network intrusion detection system (NIDS); spider monkey optimization (SMO); dimension reduction; NSL-KDD dataset; deep neural network (DNN)

1. Introduction

The development of different malicious software has posed great challenges in the design of intrusion detection systems (IDS). The attacks hurled by malicious software have become increasingly sophisticated and their detections are presently the highest prioritized area of research in computer security. Computer security and data security should therefore be given prime importance in information technology related applications, to eliminate all possible security threats and evasions. The enormous use of internet and networking has instigated more security issues violating computer security, data integrity, confidentiality, and network availability. These types of invasions are primarily

called network intrusion, referring to any kind of unauthorized activity in a network that might involve tampering, accessing or stealing resources in the network, compromising the integrity of data in the network [1–3]. There are various types of attacks that can possibly occur in a network, such as probe attacks, (denial of service (DoS) attacks, root to local (R2L) attack and user to root (U2R) attacks). In the case of probe attacks, the attacker tends to amass information regarding the target network from an external source in the network [4]. The attacking strategy in DoS attack is such that it stops the normal services in the network overwhelming the network, with irrelevant data packets sent by the attacker. The user targets a network resource by engulfing it with an internet control message protocol (ICMP) using a "ping-t" command [5]. In a U2R attack, the invader gets unauthorized control of the routing privileges and access control lists, and thereby edits the same, as per the requirement [6]. The R2L attacks are the most complicated to be detected, involving both network and host level features. Hence features relevant to both are considered for detection of such attacks and prevention could be accomplished by setting up a virtual private network (VPN) framework [6,7]. Considering the plausible threats relevant to all of the aforementioned detection, the deployment of IDS is an absolute necessity that would provide efficient surveillance and also detect any malicious activity in the network data traffic [8]. There are typically five types of IDS deployment, namely [9–13]:

- Network-based IDS (NIDS)—It is an independent platform that detects intrusions by monitoring the network traffic and hosts.
- Host-based IDS (HIDS)—It has an agent on the host that detects intrusions by monitoring system calls and application logs.
- Protocol-based IDS (PIDS)—It is normally installed on the webserver and it analyses the protocol in the computer system.
- Application protocol-based IDS (APIDS)—Focuses on monitoring of the specific application protocol used by the computer system.
- Hybrid IDS (HIDS)—Combination of multiple IDS approaches frame an HIDS, in which host agent and system data are both used to develop the complete perspective of the network system.

The primary problem pertaining to these IDS is their lower detection rate, wherein the classifiers fail to classify the intrusion events accurately, which has a further derogatory effect on the detection rate and accuracy of the system. The second most important issue with the existing IDS is the generation of false alarms, which puts immense stress on the internal teams handling such activities. A huge number of false positive alerts get generated and organizations fail to render enough time and resources to scrutinize each of the alerts with the same importance and hence leave the chances of intrusive activities slip out without being detected. Machine learning and deep learning techniques have immense potential to boost the existing intrusion detection systems, due to their general ability to detect both novel and variant attacks. The development of such IDS tends to be quite simplistic as they depend on domain specific knowledge. Machine learning-based IDS help to achieve enhanced accuracy in detection when sufficient and quality data are used but deep learning-based systems are more appropriate in dealing with big data. The deep learning-based systems analyze the raw data and learn the feature representations generating output results functioning in a sustained manner. With all these advantages, traditional machine learning algorithms—random forest classifiers, SVM, multi-layer perceptron network, extreme learning machines and various others, have been implemented on IDS datasets to detect intrusions over the network [14,15].

In order to gain more accuracy, deep neural networks have been used in convergence with hybrid optimization framework and nature inspired algorithms to reduce the execution time, ensuring energy optimization with enhanced accuracy in prediction results. Algorithms such as namely particle swarm optimization (PSO), genetic algorithm (GA), ant colony optimization (ACO), cuckoo search algorithm (CSA), harmony search algorithm (HSA), simulated annealing (SA) and various others, are inspired by nature and the characteristics of various typically based on their survival mechanisms. As an example, the spider monkey optimization algorithm has been used in deep learning for the detection

of diseases in tomato plants. A similar approach hasbeen deployed in the case of plant leaf disease detection from a high dimensional dataset, wherein the spider monkey optimization algorithm has been used in convergence with support vector machines (SVM) to achieve optimize classification results. These hybrid frameworks have been successfully deployed to resolve complex real-world problems, providing optimized and resource-efficient solutions [16].

These bio-inspired metaheuristic algorithms help in providing an optimized solution through dimensionality reduction, but have associated challenges in handling inactivity and early convergence in their classical form, especially in machine learning deployment scenarios [17]. Moreover, these algorithms have their associated issues of handling poor quality and large datasets, as datasets are mostly cross-institutional in heterogeneous format. Data standardization is thus a major requirement for the successful implementation of the existing dimensionality reduction algorithms acting as the second prioritized challenge [17]. Speed and accuracy are the major challenges in building an efficient intrusion detection system. The intrusion detection dataset consists of a higher dimensional dataset, which leads to lowering the speed of the IDS. To address this issue, we focus on the dimensionality reduction technique to reduce the number of dimensions. Thereby, the speed of the IDS can be improved. In order to overcome these challenges, the motivations of our study are as given below:

- Propose an efficient approach for data transformation and standardization
- Propose an optimized dimensionality reduction technique, to select the best features for training the machine learning model, to generate output with enhanced accuracy

The dataset used in the study is a publicly available NSL-KDD dataset collected from the Kaggle repository. This dataset was cleansed using the min-max normalization technique and passed through the 1-N encoding method for achieving homogeneity. Since the present study dealt with high dimensional data, the implementation of an efficient dimensionality reduction algorithm was absolutely necessary. To ensure only the significant attributes in the data are used for classification in deep neural network (DNN), the spider monkey optimization technique is chosen for dimensionality reduction. The reduced dataset is then subjected to a deep neural network ensuring hyper parameter tuning for optimizing the activity function. The generated classified output is evaluated with the principal component analysis (PCA)+DNN and DNN model, considering the performance metrics. The spider monkey optimization (SMO)-DNN classifier outperforms the above considered models with 99.4% accuracy, 99.5% of precision, 99.5% of recall, and 99.6% of F1-score in less training time.

The contribution of the proposed work is as follows:

- Efficient pre-processing to select optimal features influencing the classification results,
- using a hybrid SMO-DNN approach.
- Introducing a novel fitness function for spider monkey optimization.
- Optimizing the training time by dimensionality reduction and hence, lessening the burden
- of the DNN.

The organization of the paper has the sequence of information as follows—Section 2 of the paper provides an explicit review of literature and Section 3 describes the proposed SMO-DNN classifier model, along with the basic discussion of the techniques and algorithms used in this work. Section 4 discusses the results and Section 5 collates the conclusion and highlights the direction of future research works.

2. Related Work

This section deals with the study of intrusion detection systems based on the spider monkey optimization (SMO) algorithm as a dimension reduction mechanism and deep neural network (DNN) as a binary classification approach to estimate selected features.

The spider monkey optimization (SMO) method depends on the intelligent way of behaving by the spider monkey [18]. The SMO algorithm is a population-based algorithm that is mainly inspired

by the social actions which are commonly performed by spider monkeys. With its high efficiency, SMO and its variants deal effectively and successfully with complex real-world optimization problems. In a group of spider monkeys, the members of the group are from each age group. As per their aging, changes in the swiftness and agility are noticed in spider monkeys.

Sharma et al. [19] presented the ageist spider monkey optimization (ASMO) approach. This new variant is working according to the age differences of the spider monkey population, and they are more practical in biological terms.

In [20], a hybridization module of GWO and CNN to detect network anomalies in the cloud is presented. In this process, the authors are using the grey wolf optimization (GWO) algorithm for multi-objective feature extraction, which is followed by a convolutional neural network (CNN) for anomaly classification. They proposed an improved form of GWO to enhance exploitation, exploration, and ability by first population generations of the GWO as per streamed data. This new version was called improvised GWO (ImGWO). In this work, the CNN capability was improved with the help of by uniform distribution approach to revamp the dropout layer functionality. This improved CNN is named as imCNN. Finally, authors are used deep learning and multi-objective optimization to detect anomalies, followed by extracting the features with real-time network traffic streaming.

Benmessahel et al. [21] proposed enhancement in the IDS model anomaly. This model used a multi-objective grey wolf optimization (GWO) method for the identification of related features from the dataset to achieve a higher classification accuracy. In this process, the authors have used the support vector machine for estimation of the capability of predicting accurate attacks by feature selection.

In [22], the authors proposed an effective and efficient anomaly network intrusion system (ANIDS), for the detection and prevention of the inside and outside attacks in the cloud environment by low false warnings and high detection precision. This proposed method is a combination of the simulated annealing algorithm (SAA) and improved genetic algorithm (IGA), to obtain the optimal standards of the parameters involved in the construction of IDS-based DNN (IDSDNN).

In [23], the authors used a lion optimization algorithm (LOA) in combination with the social lion behavior and convolution neural network (CNN). The proposed lion optimization algorithm (LOA) is mainly contributing to the combined social lion behavior with CNN. This method is ensuring no information loss by demonstrating an optimal solution by LOA and feature selection from the feature subsets. The authors worked on this LOA model with combining ACO (ant colony optimization) + CNN and BCO (bee colony optimization algorithm) + CNN for proper evaluation. The experimental analysis of the lion optimization algorithm (LOA) shows 96% accuracy.

The authors have developed an intrusion detection system (IDS) [2], by combining the particle swarm optimization and neural network algorithms in [17]. First, KDDCUP99, NSL-KDD, and CIDD datasets are pre-processed for choosing a subset of the features, followed by dimension reduction, and finally, they perform a normalization of the data. In [24], the authors propose a random neural network and an artificial bee colony algorithm (RNN-ABC), based on the intrusion detection system (IDS). A NSL-KDD data set is used for this model. The experimental analysis of RNN-ABC-based IDS shows 95.02% accuracy.

In [25], the authors divided their work in two-folds. First, they performed a cancer gene feature selection with a spider monkey optimization algorithm for minimizing the number of features in cancer data. In the second fold, cancer data are classified by the subset of gene features. The results of this model showed that it outperforms over existing models to attain maximum classification accuracy and the minimum number of features.

Furthermore, many researchers proposed a deep neural network-nature inspired scheme-based network intrusion detection system. In [26], deep neural network approaches were used in predicting the attacks on the network intrusion detection system (N-IDS). The DNN used by authors in this paper uses learning rate of 0.1, 1000 epochs, and the KDDCup99 dataset to benchmark the network. The authors promise their direction towards cybersecurity tasks by using deep learning methods based

on [27]. A different type of a deep neural network (DNN) model-based intrusion detection method helps improve the accuracy and intelligence of network intrusion detection by reducing false alarms.

A hybridize deep neural network (DNN) algorithm and spectral clustering (SC) is the work of Ma et al. [28] In this paper, there is a division of the dataset into k subsets depending on the cluster center wise sample similarity, which calculates the distance among data points in the testing and training dataset. Then, similar features are identified and fed to the deep neural network for intrusion detection. This approach yielded high accuracy. Some more research work on the neural network and knowledge reductions are mentioned in ref [29–37].

As per the above literature survey, it observed that there is a need to address the issues in intrusion detection mechanism, such as less false-positive rate, reduction of computational complexity, improved accuracy and speed. In this paper, these issues are addressed by the proposed SMO-DNN hybrid classifier model, by exploiting the standard benchmark NSL-KDD dataset. To enhance the speed and accuracy of the intrusion detection process, the dimensionality reduction technique is applied to remove the noisy, irrelevant data. This is because nature-inspired algorithms are applied as dimensionality reduction techniques in the literature. In this paper, we introduce SMO as a dimensionality reduction technique for intrusion detection problem, which is the first of its kind.

3. Background

In this section, a detailed discussion on the basic algorithms and techniques used in this work are presented.

3.1. Network Intrusion Detection System

Intrusion is an action to access a system in an unauthorized manner, aiming to badly affect confidentiality integrity and availability of the system. Intrusion detection systems (IDSs) are designed to detect intrusion activities. IDSs detect suspicious patterns in observed events. IDSs can be divided broadly into two classes based on the algorithms adopted for intrusion detection: signature-based IDS and anomaly-based IDS. Data mining techniques not only automated intrusion detection mechanisms, but also improved efficiency and accuracy significantly. This technique can expose new intrusion signs and policy violations [38]. It can expose the unknown behavior of attackers. It also helps in decision support for intrusion management.

3.2. Deep Neural Network

An artificial neural network (ANN) has begun as a revolutionary step towards the development of new concepts of artificial intelligence (AI). For several years, researchers have been putting in successive efforts for the technical advancement in ANN computing. The result of these continuous efforts is the principle of the deep learning framework. Multiple hidden layers of the feed-forward ANN constitute a deep neural network (DNN) architecture [39]. Figure 1 shows the general architecture of the deep neural network, consisting of multiple hidden layers.

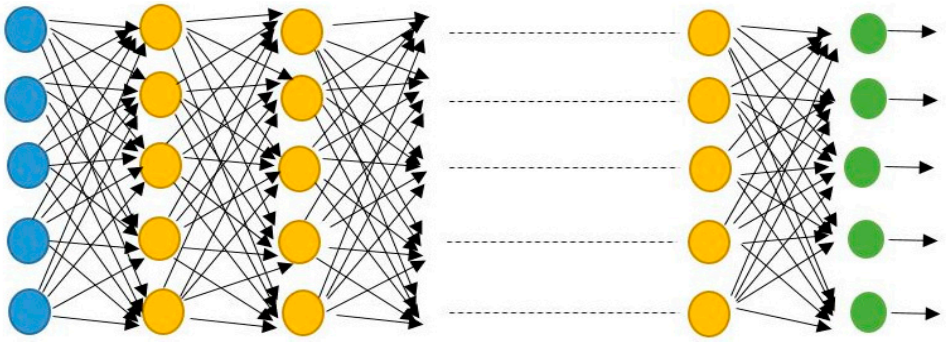

Figure 1. Architecture of a Deep Neural Network.

DNN is comprised of one input layer and one output layer, with more than threehidden layers. At present variety of deep learning constructs exist; deep belief network (DBN), convolutional neural network (CNN) [20], deep recurrent neural network (DRNN) [4], generative adversarial network (GAN) [40], deep restricted Boltzmann machine (DE-RBM), stacked auto-encoder (SAE), etc.

3.3. Nature-Inspired Algorithms

Many researchers get inspiration from natural activities, and accordingly they developed some algorithms that mimic nature's activities. These algorithms are usually referred to as nature-inspired algorithms (NIAs). NIAs have been used in conjunction with machine learning algorithms in several applications like healthcare, insurance, weather forecasting, crop yield prediction, etc. [41–43]. NIAs have been applied to build different deep learning models in many recent works [44] as described in the table below. Some researchers argued that the use of NIAs in the training phase only solve the trivial local minima problem, that can be resolved simply by just doing some alteration in errors. Some arguments favor this approach of modeling, as NIAs can find the best solutions at an optimal computational cost. But this hybrid approach of modeling needs deep examination to reveal the actual benefits of applying NIAs to build such models.

NIAs are highly efficient algorithms for solving highly nonlinear and complex real-world optimization problems effectively. NIA adopts metaheuristic approaches that aim to efficiently travel the search space, so that a near-optimal solution can be reached. Main characteristics of metaheuristic approaches are as follows:

- Metaheuristic techniques can be as simple as a local search process oras complex as a learning process.
- Metaheuristic algorithms generally result in an approximate solution.
- Metaheuristic algorithms are mostly non-deterministic.
- Metaheuristics are non-problem-specific algorithms.

3.4. Spider Monkey Optimization Algorithm

The spider monkey optimization algorithm (SMO) is one of the metaheuristic methods [41,44–46] based on the spider monkey's social behavior, adopting the fission and fusion swarm intelligence tactic for foraging [47]. Spider monkeys usually live in a swarm of 40 to 50 members. A leader decides to divide the task of food searching in a territory. Usually, a female leads the swarm as a global leader and creates mutable smaller groups, in case of food insufficiency. The group size relies on the availability of the food from a specific territory. The size is directly proportional to the amount of available food. The SMO-based algorithm satisfies the following necessary requisites of swarm intelligence (SI):

1) Labor division: Spider monkeys divide their foraging work by making smaller groups.
2) Self-organization: Size of the groups is selected to meet the required food availability.

This intelligent foraging behavior helps in making an intelligent decision (see Figure 2). Foraging behavior can be described with the following steps in short:

1. The swarm initiates food searching.
2. Computing distance of individuals from food sources.
3. The distance of the individuals from the food group members when altering their locations should be taken into consideration.
4. Again, the distance of the individuals from a food source is calculated [44].

The local leader occupies the optimal location among the sub swarm. This location changes over time according to food availability. If a local leader does not change its location within its sub swarm in a fixed number of times, then its subgroup members initiate independent foraging by moving freely in various directions.

The global leader occupies the optimal location among the whole members of the swarm. This position also changes over time according to food availability. In the situation of immobility, it decides to break the swarm into a sub swarm of reduced size. The above steps are iterated until the preferred output is attained.

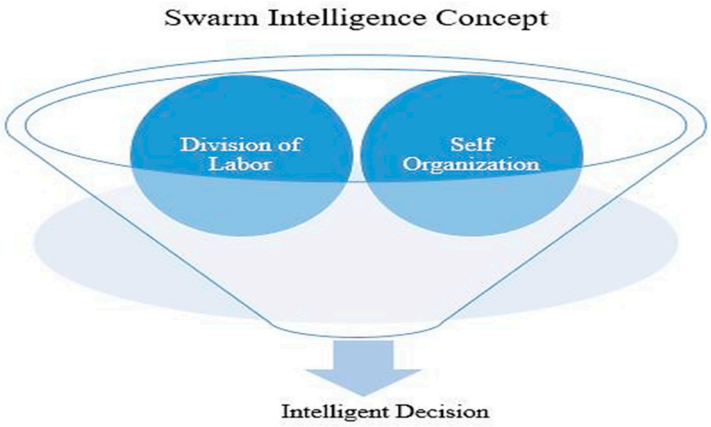

Figure 2. Swarm intelligence Concept.

Thus, a SMO-based algorithm can be categorized as a nature-inspired algorithm, based on swarm intelligence. The relationship between NIA, SIA (Swarm Intelligence Algorithm) and SMO is depicted in Figure 3.

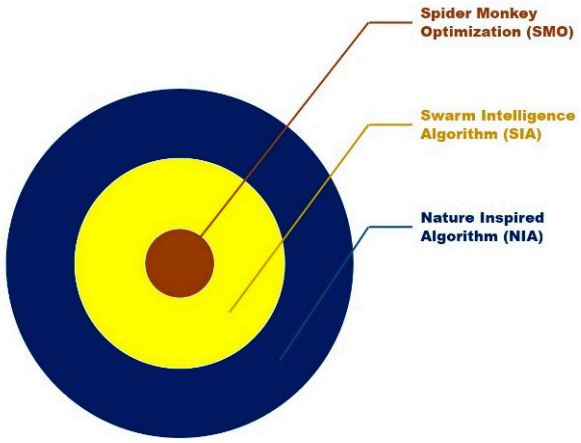

Figure 3. Relationship between spider monkey optimization (SMO), SIA and nature-inspired algorithms (NIA).

3.4.1. Key Steps of SMO algorithm implementation

SMO is apopulation based algorithm [48], which adopts a trial-and-error based collaborative iterative process which comprises six phases: local leader phase, local leader learning phase, local leader decision phase and global leader phase, global leader, learning phase and global leader decision phase. Figure 4 represents the work flow of the SMO algorithm.

The step-by-step procedure of SMO implementation is described below:

Initializing the population

SMO uniformly distributes the population of P spider monkeys SMp (where p =1, 2... P and SMp denotes the p^{th} monkey of the population). Monkeys are considered M-dimensional vectors, where M

defines total number of variables in the problem domains [19]. Each SMp is related to one possible solution to the given problem. SMO initializes each SMp using the following equation:

$$SM_{pq} = SM_{minq} + UR(0,\ 1)\ \times\ \left(SM_{maxq} - SM_{minq}\right) \quad (1)$$

where,

SM_{pq} is the q^{th} dimension of the p^{th} SM.

SM_{minq} and $SM_{maxq}$are lower and upper bounds of SM_{p}in the q^{th} direction (where q =1, 2... M)

UR (0, 1) is a random number, distributed uniformly within the range [0, 1].

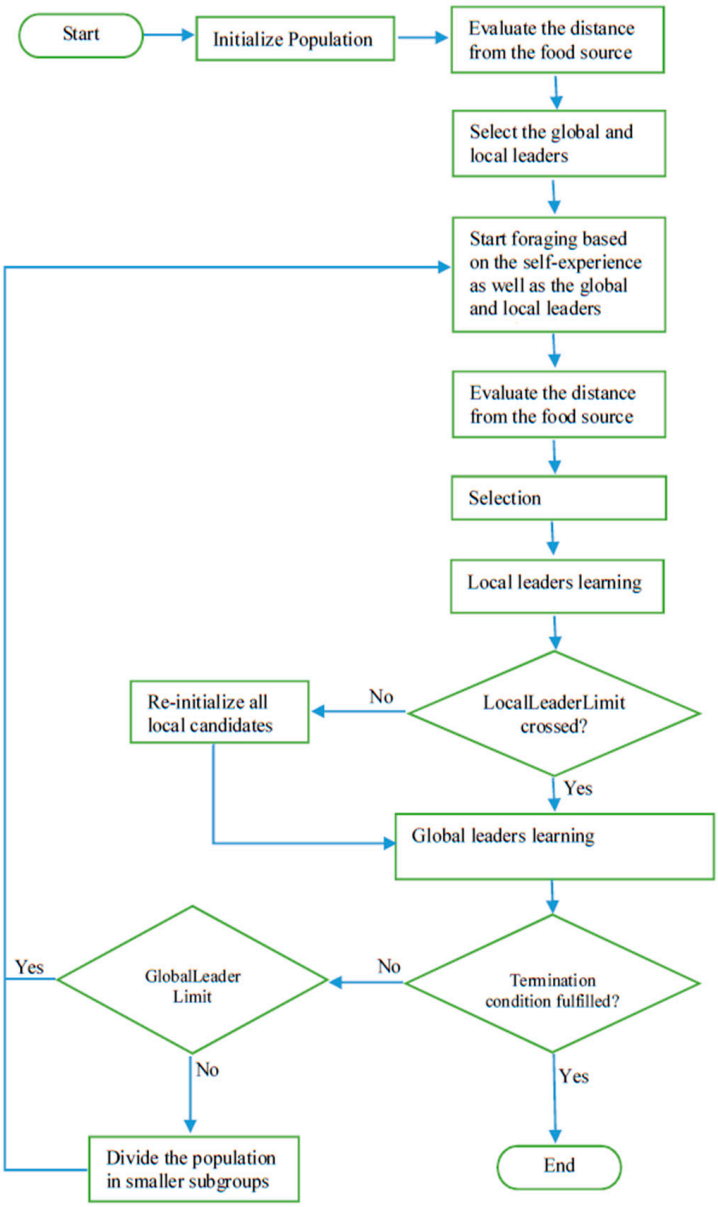

Figure 4. SMO algorithm work flow diagram.

Local Leader Phase (LLP)

In LLP, the SM changes its current location by utilizing from the past occurrences of both the local leader and the local group members. The location of the SM is updated with the new location,

only when the new location has a fitness value higher than that of the previous location. The equation for the location update of the p^{th} SM of the l^{th} local group is given below:

$$SMnew_{pq} = SM_{pq} + UR(0, 1) \times (LL_{lq} - SM_{pq}) + UR(-1, 1) \times (SM_{rq} - SM_{pq}) \quad (2)$$

where,

LL_{lq} denotes the qth dimension of the lth local group leader location.

SM_{rq} denotes the q^{th} dimension of the randomly chosen l^{th} SM of the l^{th} local group, such that $r \neq p$.

Global Leader Phase (GLP)

The global leader phase (GLP) gets started after LLP. The global leader experience and local group members' experiences are utilized to update the location of all the SM. The equation for the location update is as follows:

$$SMnew_{pq} = SM_{pq} + UR(0, 1) \times (GL_{lq} - SM_{pq}) + UR(-1, 1) \times (SM_{rq} - SM_{pq}) \quad (3)$$

Where GLq denotes the global leader location in q^{th} dimension and (q= 1, 2, 3, … M) is an arbitrarily selected index.

In this phase, the fitness of SM is used to calculate probability prbp. Based on the probability value, SM_p location is updated in this way. The better location candidates have access to an increased number of possibilities to make themselves better. The probability calculation equation is as follows:

$$prb_p = \frac{fn_p}{\sum_{p=1}^{N} fn_p} \quad (4)$$

Where fn_p is the fitness value of the p^{th}SM. Further, the fitness of the new location of the SM's is calculated and compared with that of the old location. Location with the best fitness value is adopted.

Global Leader Learning (GLL) Phase

A greedy selection method is applied to update the global leader location. The global leader location is updated with the location of the SM, with best fitness value in the population. The optimum location is assigned to the global leader. If no further updates are encountered, an increment of 1 is added to GlobalLimitCount.

Local Leader Learning (LLL) Phase

The greedy selection method is applied in the local group to update a local leader location. Local leader location is updated with the SM location with the best fitness in a particular local group. The optimum location is assigned to the local leader. If no further updates are encountered, the increment of 1 is added to LocalLimitCount.

Local Leader Decision (LLD) Phase

When a local leader does not update its location within a fixed LocalLeaderLimit, then all the candidates of that local group modify their locations randomly as per step 1, or by utilizing the past information from global leader and local leader, based on the pr through given Equation (5).

$$SMnew_{pq} = SM_{pq} + UR(0, 1) \times (GL_{lq} - SM_{pq}) + UR(0, 1) \times (SM_{rq} - LL_{pq}) \quad (5)$$

Global Leader Decision (GLD) Phase

When a global leader does not update its location up to theGlobalLeaderLimit, then the population splits into small-sized groups, as per the decision of the global leader. This group splitting process continues, until an allowed maximum number of groups (MG) is received. At each iteration, a local leader is selected for the newly shaped group. If the maximum number of allowed groups are created and the global leader does not update its position till the prefixed allowed limit, then the global leader decides to merge entire groups into a single group [29].

Parameters controlling the SMO processing are as follows:

- Value of LocalLeaderLimit
- GlobalLeaderLimit
- Max number of group (MG)
- Perturbation rate (pr)

4. SMO-DNN Hybrid Classifier Model

In this section, the overview of the proposed SMO-DNN hybrid classifier model is discussed in detail for detecting intrusion in the network traffic data. The proposed model is depicted in Figure 5.

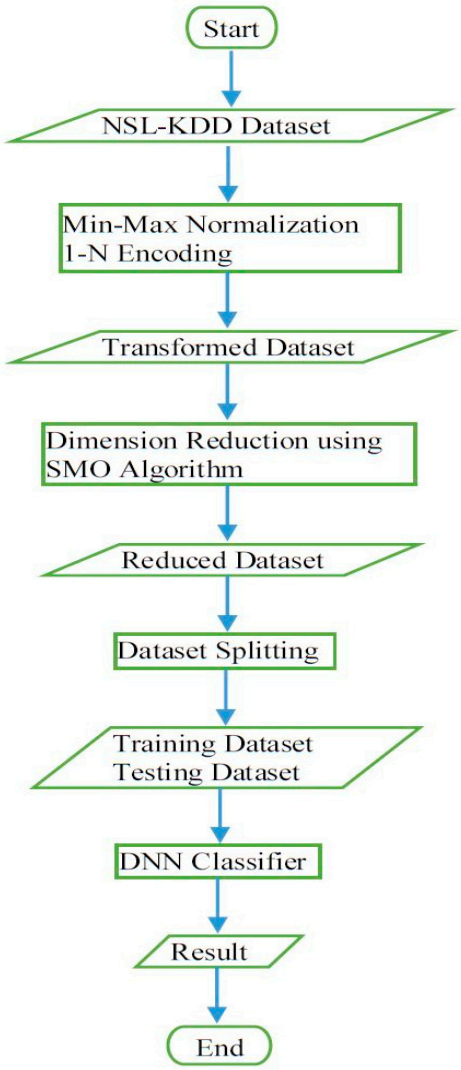

Figure 5. SMO–deep neural network (DNN) Hybrid Classifier for network intrusion detection.

The workflow of the proposed SMO-DNN hybrid classifier model is as in the following steps:

4.1. Dataset Selection

Dataset selection is the first step of the proposed SMO–DNN hybrid classifier. We studied different datasets and finally applied the NSL-KDD dataset. The standard benchmark dataset NSL-KDD [43] is utilized for illustrating the proposed SMO-DNN hybrid system, for classifying network intrusion detection data.

4.2. Data pre-processing

Data pre-processing is carried out on NSL-KDD data by applying the min-max normalization technique to normalizing the data set, which is followed by the 1-N encoding method to get an appropriate form of the data for the classifier.

4.2.1. Min-Max Normalization

Machine learning techniques are employed to discover tendencies in the dataset, by comparative evaluation among the dimension's data points. While endeavoring to use machine learning, a significant issue is that there dimensions which have drastically diverse scales.

In this paper, the min-max normalization is used to reduce the diverse scales of the dimensions. Normalization changes the data in a specific small range by carrying out linear transformation on original data. The NSL-KDD dataset consists of 41 dimensions, with values of huge variation. The dimension values of the data are normalized in the range of [0, 1] using min-max normalization. The min-max performs the transformation of data by the following equation:

$$t = \frac{v - \min_d}{\max_d - \min_d}(tran\_\max_d - tran\_\min_d) + tran\_\min_d \quad (6)$$

wheret is the transformed value of data value v in dimension d, refers to the original minimum value and $\max_d$ refers to the original maximum value of the dimension d.

Similarly, $tran\_\min_d$ refers to the transformed minimum value and $tran\_\max_d$ refers to the transformed maximum value of the dimension d.

4.2.2. 1-N Encoding Technique

It is an important pre-processing step for the structured dataset in supervised learning. The 1-n encoding converts the labels into a numeric form, so as to convert it into a machine-readable form. The machine learning algorithm can then decide, in a better way, how those labels must be operated. This technique converts the labels in a feature set between 0 and n_classes-1, where n is the number of distinct labels.

4.3. Dimension Reduction using SMO

In this step, the spider monkey optimization algorithm is utilized for the dimension reduction of the input NSL-KDD dataset. SMO is a combined iterative process, it uses the foraging behavior of spider monkeys, which is the fission-fusion social structure (FFSS) of animals. Many research works have developed algorithms using a deep neural network without feature engineering or using traditional dimension reduction techniques. During the training phase of DNN, if the number of features is high, then the model over-fitting problem arises. In order to address this issue, the proposed model exploits the spider monkey optimization algorithm, which is an FFSS behavior-based algorithm, applied for reducing the number of dimensions.

For a dimension set sized D, the assorted dimension subsets would be higher in number, and the number of dimensions will make a massive space of dimensions, which is to be searched meticulously. Hence, SMO is incorporated to screen the search space in an adaptive manner for an optimal dimension

subset. The best dimension set will be the one which has the highest accuracy or lowest error rate and a minimum number of extracted dimensions. Generally, in intrusion detection problems, dimension reduction is subjected to objective conflicts; to minimalize the dimension count and maximize the accuracy of classification or diminished error rate. The existence of trade-offs among conflicting objectives caused difficulty in obtaining optimal results. Therefore, various constraints with one objective cannot present this situation appropriately. In this situation, in order to minimize or maximize the group of objective functions, it is essential to apply a multi-objective optimization technique [20].

The proposed method intended to evaluate a dimension subset, to receive the highest accuracy (A) for the classifier. This paper utilizes accuracy as a performance metric for calculating the classification error (E) of the SMO. In the evolutionary training phase, the fitness function F_D examines every probable dimension subset for finding one which maximizes the classification accuracy included in dimension reduction.

The fitness function FD is utilized in SMO to measure individual cause, defined as follows [36];

$$F_D = \beta * (1 - A) + (1 - \beta) * \frac{S}{D} \tag{7}$$

where A is the classifier accuracy of extracted dimension subset, β is a constant for regulating the accuracy of the classification and the dimension reduction, S is the number of extracted dimension subsets, D is the overall number of dimensions in the dataset and in the [0, 1] range. In this work, $\beta = 0.9$. Then, the resultant dimensions of the NSL-KDD dataset is passed onto DNN for further classification.

4.4. Data Output

After performing the dimension reduction on NSL-KDD data by SMO, the generated data are collected at this stage.

4.5. Data Splitting

The dimensionally reduced NSL-KDD and KDD Cup 99 data are then divided into two subsets; training data and testing data. The 10% dataset is selected from the entire NSL-KDD and KDD CUP 99 for experimentation. Among them are 70% of the data used for training processes and 30% of the data utilized for testing processes.

4.6. Intrusion Detection Using Deep Neural Network

DNN consists of one input, a minimum of three hidden layers, and one output layer. The first hidden layer comprises 32 nodes, the second hidden layer 64 nodes, and the third hidden layer 128 nodes. In the present study, the reduced feature set generated from SMO is fed to the DNN as input. The input layer passes the input feature vectors into the hidden layers. The hyper parameters of hidden layers, such as learning rate, activation function, etc. are tuned before processing the inputs. The learning rate is optimized using an Adam optimizer. The ReLU (rectified linear units) activation function does not have the vanishing gradient problem, as in the case of the sigmoid and tanh activation function. Thus, in the proposed classifier model, ReLU activation function is incorporated on the hidden layers. The Softmax classifier is applied to the output layer, to classify the incoming network traffic as either normal or an intrusion.

The performance of the proposed system is evaluated in terms of accuracy and time complexity during the training phase of the DNN. Further, the performance of the proposed system is compared with PCA+DNN and DNN algorithms. The experimental result and analysis are discussed in the next section.

5. Results and Discussion

In this section, the results of the experimentation are evaluated and analyzed. The rest of the section discusses the dataset used for experimentation, evaluation metrics, the performance evaluation

of the proposed model, comparative analysis of the proposed model with existing models, and a detailed discussion about the results of the experimentation.

5.1. Dataset Description

The standard benchmark dataset NSL-KDDand KDD Cup 99 are applied for evaluating the proposed system for network intrusion detection. Many literature works [10,11,16,28] have utilized this benchmark dataset for evaluating their research on intrusion detection. NSL-KDD, an enhanced dataset of the KDD CUP'99, was employed in this research work. This dataset has reduced duplicate records, as in KDD CUP'99. The NSL-KDD dataset comprises 41 features, where 39 features are numeric records and the other three features are symbolic records.

Table 1 depicts the features present in the NSL-KDD dataset. NSL-KDD and KDD Cup 99dataset features can be categorized into four groups:

Basic features: 1–9
Traffic features:23–31
Host features: 32–41
Content features: 10–22

Table 1. List of features in the NSL-KDD and KDD Cup 99 dataset.

No	Feature Name	No	Feature Name	No	Feature Name
1	Duration	15	Su_attempted	29	Same_srv_rate
2	Protocol_type	16	Num_root	30	Diff_srv_rate
3	Service	17	Num_file_creations	31	Srv_diff_host_ rate
4	Flag	18	Num_shells	32	Dst_host_count
5	Src_bytes	19	Num_access_files	33	Dst_host_srv_ count
6	Dst_bytes	20	Num_outbound_cmds	34	Dst_host_same _srv_rate
7	Land	21	Is_hot_login	35	Dst_host_diff_ srv_rate
8	Wrong_fragment	22	Is_guest_login	36	Dst_host_same _src_port_rate
9	Urgent	23	Count	37	Dst_host_srv_ diff_host_rate
10	Host	24	Srv_count	38	Dst_host_serror_rate
11	Num_failed _logins	25	Serror_rate	39	Dst_host_srv_serror_rate
12	Logged_in	26	Srv_serror_rate	40	Dst_host_rerror_rate
13	Num_compromised	27	Rerror_rate	41	Dst_host_rerror_rate
14	Root_shell	28	Srv_rerror_rate		

5.2. Evaluation Metrics

Accuracy, precision, recall, F1-score and time complexity formulas are shown in Table 2. These are used as the evaluation metrics for validating the proposed system. The computation of the accuracy metric involves the following attributes, namely true positive (TPos), false positive (FPos), true negative (TNeg) and false negative (FNeg).

Table 2. Evaluation Metrics used in the proposed system.

Evaluation Metric	Equation
Accuracy	$\frac{TPos+TNeg}{TPos+TNeg+FPos+FNeg}$
Precision	$\frac{TPos}{TPos+FPos}$
Recall	$\frac{TPos}{TPos+FNeg}$
F1—score	$\frac{Precision \times Recall}{Precision+Recall}$
Sensitivity	$\frac{TPos}{TPos+FNeg}$
Specificity	$\frac{TNeg}{TNeg+FPos}$

5.3. Performance Evaluation

The experimental setup and outcomes of the proposed SMO-DNN hybrid classifier model are briefly discussed in this section. The proposed model is used for network intrusion classification. The evaluations are performed using the NSL-KDD benchmark dataset [49]. The proposed system functions in two phases, namely; (1) Dimensionality Reduction phase (2) Classification phase. In the dimensionality reduction phase, to reduce the overfitting occurrence during DNN training, firstly, the SMO technique is applied, which reduces the number of dimensions. In the classification phase, the DNN is utilized for the classification of the NSL-KDD benchmark network intrusion dataset. Hardware with the following specifications was used to conduct the experimentation: the 64-bit OS, Intel (R) Core (TM) i7-8750H CPU@ 2.20 GHz and 8.00 GB RAM.

DNN is constructed with the resultant dataset generated by the SMO algorithm as input nodes, three hidden layers with the activation function ReLU, and two output nodes with activation function Softmax. The hyper parameter tuning is performed for the learning rate using a grid search method. After applying the grid search method, the learning rate of 0.01 is applied for training DNN. The number of epochs used in training is 100, with 10-fold cross-validation for validating the proposed system. The performance of the proposed system is compared with PCA+DNN and DNN. The PCA+DNN were chosen for comparison with traditional PCA algorithm for dimensionality reduction and classification by DNN. The performance of the proposed system is also tested with DNN, without any dimensional reduction technique to show how well the proposed system performs in terms of speed and accuracy in detecting the intrusion.

Tables 3 and 4 show the performance of the proposed system, with respect to accuracy, precision, recall, F1-score, sensitivity, specificity and time complexity, using the NSL-KDD dataset and KDD Cup 99. Moreover, these results are also compared with other methods. It can be observed from the experimental results that the proposed SMO+DNN performed well compared to PCA+DNN and DNN. An improved accuracy of 97% using NSL-KDD and 92% using KDD Cup 99 are achieved with the proposed SMO+DNN, and furthermore, the change in the dataset has not affected much to the performance of the proposed model.

It has been observed that the proposed SMO+DNN exhibit improved results compared with PCA+DNN and DNN in terms of precision, recall, and F1-score. The proposed system was achieved using NSL-KDD and KDD Cup 99 with 99.5% and 92.7% of precision, 99.5% and 92.8% of recall and 99.6% and 92.7% of F1-score.

It can be observed that the proposed model shows increased sensitivity and specificity of 99.4% and 92.8%, and 99.6% and 93%, compared with other models considered for comparison.

Furthermore, the training time of the proposed system is less when compared with other existing methods. Thus, it has been observed that the time complexity is greatly reduced while experimenting with the proposed SMO-DNN hybrid classifier system, as compared to PCA+DNN and DNN.

Table 3. Detailed performance evaluation of the SMO-DNN using NSL-KDD.

Model	SMO+DNN	PCA+DNN	DNN
Accuracy	0.994	0.938	0.914
Precision	0.995	0.934	0.891
Recall	0.995	0.918	0.882
F-score	0.996	0.937	0.905
Sensitivity	0.994	0.938	0.908
Specificity	0.996	0.926	0.898
Time complexity (min)	65	72	90

Table 4. Detailed performance evaluation of the SMO-DNN using KDDCup 99.

Model	SMO+DNN	PCA+DNN	DNN
Accuracy	0.928	0.898	0.909
Precision	0.927	0.884	0.896
Recall	0.928	0.898	0.909
F-score	0.927	0.882	0.894
Sensitivity	0.928	0.898	0.909
Specificity	0.930	0.885	0.882
Time complexity (min)	80	120	170

5.4. Discussion

The dataset used had several attributes with values of different ranges. This may result in biased predictions. So, we used min-max normalization to fit the values of the attributes into a common range. As the attributes protocol_type, service, and flag in the dataset are categorical, the dataset was not suitable for training, as machine learning (ML) algorithms will process only numerical data. Hence, in this work, the 1-n encoding technique is used for transforming the NSL KDD dataset. This transformed dataset had 40 participating attributes. As all of these attributes might not affect the class label in a positive way, a rigorous optimization algorithm, namely the spider monkey meta-heuristic algorithm, was used in this work. The main benefit of the spider monkey algorithm (SMO) is that it helps the attributes to converge quickly, with optimal parameters for efficient classification, which makes it an optimal choice for dimensionality reduction. Hence, the transformed dataset is dimensionally reduced by SMO. The feature engineering process followed above resulted in 28 optimal attributes: Duration, Protocol_type, Service, Flag, Src_bytes, Land, Wrong_fragment, Urgent, Logged_in, Num_compromised, Su_attempted, Num_root, Num_file_creations, Num_shells, Num_access_files, Is_hot_login, Is_guest_login, Count, Srv_serror_rate, Srv_rerror_rate, Same_srv_rate, Diff_srv_rate, Srv_diff_host_ rate, Dst_host_count, Dst_host_srv_ count, Dst_host_same _src_port_rate, Dst_host_srv_ diff_host_rate and Dst_host_srv_serror_rate. The resultant attributes are then fed to DNN for classification. The proposed model is evaluated using metrics accuracy, precision, recall, F1-Score. An accuracy of 99.4%, precision 99.5, recall 99.5%, and F1-score 99.6 have been attained. Using KDD Cup 99, these were; an accuracy of 92%, precision 92.7%, recall 92.8% and F1-score 92.7%. The results achieved are compared with PCA+DNN and DNN. The comparative results proved that the proposed SMO-DNN model expressed an indicative increase over the existing considered algorithms. Even the training time of the proposed model outperformed the other two models, PCA+DNN by 7 min and DNN by 25 minutes. Furthermore, KDD CUP 99 outperformed the other two models, PCA+DNN by 7 minutes and 40 minutes and DNN by 25 minutes and 90 minutes.

From the above discussion, the contributions made by the proposed work can be summarized as follows:

- An extensive pre-processing has been performed to normalize, transform the dataset for better predictions.
- A novel SMO-DNN model is proposed for feature engineering for selecting the best features (attributes) which positively affect the classification accuracy. The convergence rate and the fitness function used in SMO resulted in better classification results when compared with existing models.

6. Conclusions

In this research work, a novel SMO-DNN hybrid classifier model for intrusion detection is presented, which is the first of its kind. The proposed model was implemented and deployed using a standard benchmark NSL-KDD and KDD Cup 99 dataset. This model leverages the advantages of the

spider monkey optimizer in reducing the dimension, and then the binary classification is performed by applying the deep neural network. The evaluations on the proposed model were performed using the performance metrics, namely, accuracy, precision, recall and, F1-score. Moreover, the training time for the DNN is computed in determining the performance of the proposed SMO-DNN model. Further, the results are compared with the PCA+DNN and DNN models. The proposed model resulted in an enhanced accuracy of 97%and92%, 99.5% and92.7% of precision, 99.5% and92.8% of recall and 99.6% and92.7% of F1-score, and less training time than other models considered for comparison. The limitation of the proposed model is that it is only applied for binary classification.

In the future, the effectiveness of the proposed work can be examined by extending the work with multiclass classification. Furthermore, the proposed method can be extended for detecting anomalies in IoT networks and malware in foggy and cloudy environments. The inherent complexity present in the foggy and cloudy environments is induced because heterogeneous inward traffic and hardware usage makes the intrusion detection task troublesome.

Author Contributions: Conceptualization, N.K. and P.D.; methodology, N.K. and C.L.C.; software, S.B, S.S.; validation, N.K. and B.Y.; formal analysis, N.K., P.D and C.L.C; investigation, S.S. and B.Y.; resources, N.K.; data curation, D.K. and P.D.; writing—original draft preparation, N.K. and P.D.; writing—review and editing, C.L.C., S.B, G.S., S.S and B.Y.; visualization, G.S.; supervision, S.B. and B.Y.; project administration, N.K.; funding acquisition, B.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Dongguk University Research Fund of 2016 (S-2019-G0001-00043).

Acknowledgments: This work was partially supported by the National Research Foundation of Korea(NRF-2019R1A2C1085388).

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

References

1. Liao, H.J.; Lin, C.H.R.; Lin, Y.C.; Tung, K.Y. Intrusion detection system: A comprehensive review. *J. Netw. Comput. Appl.* **2013**, *36*, 16–24. [CrossRef]
2. Iwendi, C.; Uddin, M.; Ansere, J.A.; Nkurunziza, P.; Anajemba, J.H.; Bashir, A.K. On detection of Sybil attack in large-scale VANETs using spider-monkey technique. *IEEE Access* **2018**, *6*, 47258–47267. [CrossRef]
3. Kuang, F.; Xu, W.; Zhang, S. A novel hybrid KPCA and SVM with GA model for intrusion detection. *Appl. Soft Comput.* **2014**, *18*, 178–184. [CrossRef]
4. Yin, C.; Zhu, Y.; Fei, J.; He, X. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access* **2017**, *5*, 21954–21961. [CrossRef]
5. Zhang, H.; Qi, Y.; Zhou, H.; Zhang, J.; Sun, J. Testing and defending methods against DoS attack in state estimation. *Asian J. Control* **2017**, *19*, 1295–1305. [CrossRef]
6. Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J. Comput. Sci.* **2018**, *25*, 152–160. [CrossRef]
7. Peng, F.; Qin, L.; Long, M. Face presentation attack detection using guided scale texture. *Multimed. Tools Appl.* **2018**, *77*, 8883–8909. [CrossRef]
8. Arul, R.; Moorthy, R.S.; Bashir, A.K. Ensemble Learning Mechanisms for Threat Detection: A Survey. In *Machine Learning and Cognitive Science Applications in Cyber Security*; IGI Global: Hershey, PA, USA, 2019; pp. 240–281.
9. Agrawal, G.; Soni, S.K.; Agrawal, C. A survey on attacks and approaches of intrusion detection systems. *Int. J. Adv. Res. Comput. Sci.* **2017**, *8*, 499–504. [CrossRef]
10. Benkhelifa, E.; Welsh, T.; Hamouda, W. A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3496–3509. [CrossRef]
11. Jin, H.; Xiang, G.; Zou, D.; Wu, S.; Zhao, F.; Li, M.; Zheng, W. A VMM-based intrusion prevention system in cloud computing environment. *J. Supercomput.* **2013**, *66*, 1133–1151. [CrossRef]
12. 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]

13. Raza, S.; Wallgren, L.; Voigt, T. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad hoc Netw.* **2013**, *11*, 2661–2674. [CrossRef]
14. Haq, N.F.; Onik, A.R.; Hridoy, M.A.K.; Rafni, M.; Shah, F.M.; Farid, D.M. Application of machine learning approaches in intrusion detection system: A survey. *Ijarai-Int. J. Adv. Res. Artif. Intell.* **2015**, *4*, 9–18.
15. Buczak, A.L.; Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1153–1176. [CrossRef]
16. Chiba, Z.; Abghour, N.; Moussaid, K.; Rida, M. Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms. *Comput. Secur.* **2019**, *86*, 291–317. [CrossRef]
17. Bansal, J.C.; Sharma, H.; Jadon, S.S.; Clerc, M. Spider monkey optimization algorithm for numerical optimization. *Memetic Comput.* **2014**, *6*, 31–47. [CrossRef]
18. Agrawal, V.; Rastogi, R.; Tiwari, D. Spider monkey optimization: A survey. *Int. J. Syst. Assur. Eng. Manag.* **2018**, *9*, 929–941. [CrossRef]
19. Sharma, A.; Sharma, A.; Panigrahi, B.K.; Kiran, D.; Kumar, R. Ageist spider monkey optimization algorithm. *Swarm Evol. Comput.* **2016**, *28*, 58–77. [CrossRef]
20. Garg, S.; Kaur, K.; Kumar, N.; Kaddoum, G.; Zomaya, A.Y.; Ranjan, R. A hybrid deep learning-based model for anomaly detection in cloud data center networks. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 924–935. [CrossRef]
21. Benmessahel, I.; Xie, K.; Chellal, M.; Semong, T. A new evolutionary neural networks based on intrusion detection systems using locust swarm optimization. *Evol. Intell.* **2019**, *12*, 131–146. [CrossRef]
22. Alamiedy, T.A.; Anbar, M.; Alqattan, Z.N.; Alzubi, Q.M. Anomaly based intrusion detection system using multi-objective grey wolf optimisation algorithm. *J. Ambient Intell. Humaniz. Comput.* **2019**, 1–22. [CrossRef]
23. Selvakumar, B.; Muneeswaran, K. Firefly algorithm based feature selection for network intrusion detection. *Comput. Secur.* **2019**, *81*, 148–155.
24. Kancharla, G.R.; Eluri, N.R.; Dara, S.; Ansari, N. An efficient algorithm for feature selection problem in gene expression data: A spider monkey optimization approach. *SSRN Electron. J.* **2019**. [CrossRef]
25. Gupta, K.; Deep, K.; Bansal, J.C. Spider monkey optimization algorithm for constrained optimization problems. *Soft Comput.* **2017**, *21*, 6933–6962. [CrossRef]
26. Arivudainambi, D.; Kumar, V.K.A.; Chakkaravarthy, S.S. Lion ids: A Meta heuristics approach to detect ddos attacks against software-defined networks. *Neural Comput. Appl.* **2019**, *31*, 1491–1501. [CrossRef]
27. Shokoohsaljooghi, A.; Mirvaziri, H. Performance improvement of intrusion detection system using neural networks and particle swarm optimization algorithms. *Int. J. Inf. Technol.* **2019**, 1–12. [CrossRef]
28. Ma, T.; Wang, F.; Cheng, J.; Yu, Y.; Chen, X. A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. *Sensors* **2016**, *16*, 1701. [CrossRef]
29. Lang, G.; Li, Q.; Cai, M.; Yang, T.; Xiao, Q. Incremental approaches to knowledge reduction based on characteristic matrices. *Int. J. Mach. Learn. Cybern.* **2017**, *8*, 203–222. [CrossRef]
30. Cai, Z.-W.; Huang, L.-H. Finite-time synchronization by switching state-feedback control for discontinuous Cohen–Grossberg neural networks with mixed delays. *Int. J. Mach. Learn. Cybern.* **2018**, *9*, 1683–1695. [CrossRef]
31. Wang, D.; Huang, L.; Tang, L. Dissipativity and synchronization of generalized BAM neural networks with multivariate discontinuous activations. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 3815–3827.
32. Kuang, F.; Zhang, S.; Jin, Z.; Xu, W. A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection. *Soft Comput.* **2015**, *19*, 1187–1199. [CrossRef]
33. Wang, X.; Mao, H.; Hu, H.; Zhang, Z. Crack localization in hydraulic turbine blades based on kernel independent component analysis and wavelet neural network. *Int. J. Comput. Intell. Syst.* **2013**, *6*, 1116–1124. [CrossRef]
34. Lang, G.; Cai, M.; Fujita, H.; Xiao, Q. Related families-based attribute reduction of dynamic covering decision information systems. *Knowl.-Based Syst.* **2018**, *162*, 161–173. [CrossRef]
35. Huang, C.; Liu, B. New studies on dynamic analysis of inertial neural networks involving non-reduced order method. *Neurocomputing* **2019**, *325*, 283–287. [CrossRef]
36. Reddy, G.T.; Reddy, M.P.K.; Lakshmanna, K.; Kaluri, R.; Rajput, D.S.; Srivastava, G.; Baker, T. Analysis of Dimensionality Reduction Techniques on Big Data. *IEEE Access* **2020**, *8*, 54776–54788. [CrossRef]

37. Hegazy, A.E.; Makhlouf, M.A.; El-Tawel, G.S. Dimensionality Reduction Using an Improved Whale Optimization Algorithm for Data Classification. *Int. J. Mod. Educ. Comput. Sci.* **2018**, *7*, 37–49. [CrossRef]
38. Mehibs, S.M.; Hashim, S.H. Proposed network intrusion detection system in cloud environment based on back propagation neural network. *J. Univ. Babylon Pure Appl. Sci.* **2018**, *26*, 29–40.
39. Deng, L.; Yu, D. Deep learning: Methods and applications. *Found. Trends® Signal. Process.* **2014**, *7*, 197–387. [CrossRef]
40. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
41. Reddy, G.T.; Khare, N. Heart disease classification system using optimised fuzzy rule based algorithm. *Int. J. Biomed. Eng. Technol.* **2018**, *27*, 183–202. [CrossRef]
42. Bhattacharya, S.; Kaluri, R.; Singh, S.; Alazab, M.; Tariq, U. A Novel PCA-Firefly based XGBoost classification model for Intrusion Detection in Networks using GPU. *Electronics* **2020**, *9*, 219. [CrossRef]
43. Gadekallu, T.R.; Khare, N.; Bhattacharya, S.; Singh, S.; Reddy Maddikunta, P.K.; Ra, I.H.; Alazab, M. Early Detection of Diabetic Retinopathy Using PCA-Firefly Based Deep Learning Model. *Electronics* **2020**, *9*, 274. [CrossRef]
44. Spider Monkey Optimisation Algorithm. Available online: http://smo.scrs.in/ (accessed on 19 March 2020).
45. Iwendi, C.; Maddikunta, P.K.R.; Gadekallu, T.R.; Lakshmanna, K.; Bashir, A.K.; Piran, M.J. A metaheuristic optimization approach for energy efficiency in the IoT networks. *Softw. Pract. Exp.* **2020**. [CrossRef]
46. Ji, Y.; Liu, L.; Wang, H.; Liu, Z.; Niu, Z.; Denby, B. Updating the Silent Speech Challenge benchmark with deep learning. *Speech Commun.* **2018**, *98*, 42–50. [CrossRef]
47. Yang, X.-S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214.
48. Sultan, S.; Javed, A.; Irtaza, A.; Dawood, H.; Dawood, H.; Bashir, A.K. A hybrid egocentric video summarization method to improve the healthcare for Alzheimer patients. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 4197–4206. [CrossRef]
49. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.