Research Article

A deep learning-based multi-agent system for intrusion detection

Faten Louati[1] · Farah Barika Ktata[2]

Received: 5 December 2019 / Accepted: 2 March 2020 / Published online: 17 March 2020

Abstract
Intrusion detection systems play an important role in preventing attacks which have been increased rapidly due to the dependence on network and Internet connectivity. Deep learning algorithms are promising techniques, which have been used in many classification problems. In the same way, multi-agent systems become a new useful approach in intrusion detection field. In this paper, we propose a deep learning-based multi-agent system for intrusion detection which combines the desired features of multi-agent system approach with the precision of deep learning algorithms. Therefore, we created a number of autonomous, intelligent and adaptive agents that implanted three algorithms, namely autoencoder, multilayer perceptron and k-nearest neighbors. Autoencoder is used as features reduction tool, and multilayer perceptron and k-nearest neighbors are used as classifiers. The performance of our model is compared against traditional machine learning approaches and other multi-agent system-based systems. The experiments have shown that our hybrid distributed intrusion detection system achieves the detection with better accuracy rate and it reduces considerably the time of detection.

Keywords Intrusion detection system · Deep learning · Multi-agent system · KDD 99 · Multilayer perceptron · Autoencoder · K-nearest neighbors

1 Introduction

Nowadays, with the increased use of the Internet which becomes an essential tool in daily life, various types of attacks are faced. Therefore, security in networks becomes a serious issue.

Intrusion detection system (IDS) can be an efficient solution for this problem. The concept of intrusion detection (ID) dates back to 1980, and it was proposed by Anderson [1]. An intrusion detection system (IDS) is a type of security software that monitors, analyzes network traffics and alerts administrators automatically when a malicious activity is detected.

Various techniques have been used to develop IDSs. On the one hand, a wide range of machine learning methodologies have been used such as artificial neural network (ANN), support vector machine (SVM) and naive Bayes. On the other hand, multi-agent system (MAS) is an another concept widely investigated in intrusion detection field.

Since most of the traditional machine learning methodologies cannot effectively solve the massive intrusion data classification problem [29], deep learning-based methods have been recently successfully applied to built IDSs. Studies have shown that deep learning completely outperforms traditional methods [29].

Furthermore, most of existing IDSs suffer from their monolithic architectures that contain a central analyzer. If this analyzer fails, other components will be affected. A hopeful feature of an IDS architecture is its ability to implement a distributed format which can make the IDS more robust. For this reason, several researches are oriented

✉ Faten Louati, faten1louati@gmail.com; Farah Barika Ktata, farah.ktata@gmail.com | [1]Institut Supérieur de Gestion de Sousse, Rue Abdelaziz II Behi, 4000 Sousse, Tunisia. [2]Institut Supérieur des Sciences Appliquées et de Technologie de Sousse, Rue ibn Khaldun, Cité Taffala, 4003 Sousse, Tunisia.

toward the multi-agent system (MAS) for intrusion detection task.

For this motivation, a combination of aforementioned approaches is proposed to take advantages of both deep learning and multi-agent system. We present an anomaly intrusion detection scheme based on adaptive and intelligent agents. It is a distributed IDS that integrates the features provided by multi-agent approach with the performance of deep learning technique. Therefore, the agents in the proposed IDS implement three algorithms: autoencoder (AE), multilayer perceptron (MLP) and k-nearest neighbors (K-NN) to identify intrusions on networks: Autoencoder is used as a feature reduction tool that is followed by MLP and K-NN classifiers. KDD CUB 99 benchmark dataset is used for testing the proposed solution.

The remaining part of the paper is organized in the following way: In Sect. 2, we present an overview of the main concepts used in this paper. In Sect. 3, we discuss same related works. The description of the proposed solution is introduced in Sects. 4 and 5. Section 6 presents the experimental results and compares them with other works. Finally, in Sect. 7 we conclude the entire work and present our future works.

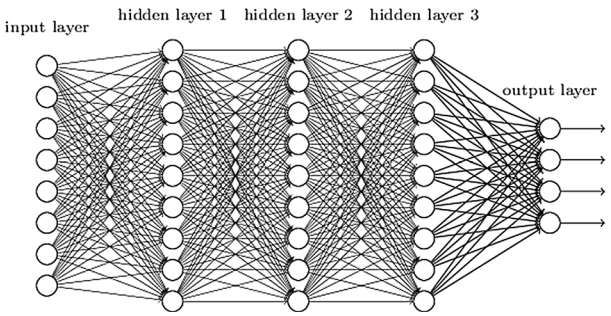

Fig. 1 A deep neural network example [24]

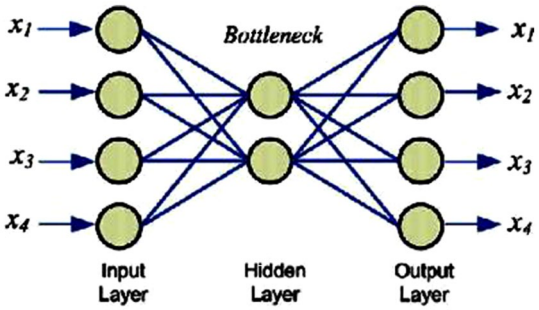

Fig. 2 Autoencoder structure [14]

2 Overview

2.1 Deep learning

Deep learning is a new field of machine learning which has been applied in many areas such as speech and image recognition, natural language processing, drug discovery and recommended systems. In the last few years, deep learning has proven its efficiency in the intrusion detection field and security area in general.

Deep learning is based on artificial neural network which is a computational model inspired from human brain. It consists of a large number of connected nodes called neurons (a.k.a. perceptrons), and each neuron performs a simple mathematical operation (activation function). Each neuron's output is determined by this operation, as well as a set of parameters (weight and bias) that are specific to that node.

Deep neural network is a neural network with more than two layers: an input layer, at least one hidden layer and an output layer (Fig. 1).

2.1.1 Multilayer perceptron

Multilayer perception is a subset of the deep neural network (is a feed-forward neural network). Multilayer perceptron formula is based on *backpropagation* algorithm (short for *"backward propagation of errors"*) which is predicated on the error propagation method. It consists of two phases, a feed-forward phase and a backforward phase. In the first step, data propagate across the network to finally get the output and compare it with the real values to get the error and then to be minimized, the error is backpropagated to the previous layer, and then the weights are adjusted accordingly. This process is repeated until the error is below a predetermined threshold.

The purpose of MLP in this paper is to solve a five-class problem, i.e., assign the input patterns to one of the categories that are represented in terms of neural networks outputs (the four types of attacks, namely DoS, Probe, U2R and R2L + normal.)

2.1.2 Autoencoder

Autoencoder is a feed-forward neural network , very similar to MLP except that output layer must have the same number of nodes as the input layer (Fig. 2) since the purpose of the autoencoder is reconstructing its own inputs (instead of predicting the target values Y given inputs X). Typically, autoencoder is used for dimensionality reduction.

Dimensionality reduction attempts to reduce the number of variables in the data, and it facilitates the classification, visualization, communication and storage of high-dimensional data [7]. There are two types of dimensionality

reduction: feature selection and feature extraction. Feature selection consists of removing unnecessary features. However, feature extraction means the transformation of raw data into features suitable for modeling.

In this work, an autoencoder is used for features selection task and reduces efficiently the dimension of the dataset from 120 to only 10.

2.1.3 Overfitting problem and solution

Is a problem that can occur during neural network training when the classification error on the training set is driven to a very small value, but when unknown data are faced, the error increases. This can be explained by the fact that the neural network has just memorized the training examples not learn to generalize the solution to new samples.

To solve this problem, we used a technique known as *Early Stopping*. This technique consists of dividing the dataset into three subsets, namely training set, testing set and validating set. The training set is used to train the neural network, the testing set is used to test the neural network, and the validating set is used to monitor the error during the training phase. The validating error will decrease similarly to the training error. However, when the error begins to rise, then the neural network begins to overfit the data. Thus, the training process is stopped, and the weights generating the minimum error on the validating set are stored.

2.1.4 K-nearest neighbor

K-nearest neighbor (K-NN) is one of the popular machine learning algorithms. Despite its simplicity, K-NN is a powerful algorithm which can be used in classification task and in a variety of applications such as intrusion detection. It separates instances in a given dataset into several classes so that it can predict the classification of a new sample. K-NN memorizes the training instances to be used further in prediction phase. Therefore, it does not explicitly learn a model [30]. The concept consists of calculating distance between two data points to make a vote between the K most similar data points to a given "new" data point.

2.2 Multi-agent system

Multi-agent system is a system composed of multiple interactive computing elements called agents [26]. An agent is a computer system (software or robot) with two important capabilities: autonomous and interaction [26].

2.2.1 Features of an agent

According to [27, 28], an agent has the following characteristics:

- *Autonomy* An agent can decide for itself what it needs to do in order to satisfy its goals.
- *Re-activity* An agent perceives and acts on its environment.
- *Pro-activity* An agent may be able to take the initiative.
- *Sociability* An agent can interact with other agents, using an agent communication language (ACL). Therefore, an agent is able to provide and ask for services, can cooperate, coordinate, negotiate and so on.
- *Mobility* An agent may be able to move from one system to another.
- *Adaptation* An agent may—if needed—attempt to adapt itself to new or changing environment or to deal with new or changing goals.
- *Learning* An agent may learn from past occurrences in the environment to predict the future.

2.2.2 Advantages of MAS

According to [5], there are two main advantages of MAS:

- *Robustness* The ability that the system can tolerate failures of one or more agents.
- it Scalability It should be easier to add new agents to a MAS than to add new capabilities to a monolithic system.

2.3 Intrusion detection system

The concept of intrusion detection (ID) dates back to 1980, when it was proposed by Anderson [1]. There are three types of IDS, namely host-based IDS, network-based IDS and distributed IDS. Host-based intrusion detection system (HIDS) is placed on a particular computer or server (host) and monitors activity only on that system. However, network-based intrusion detection system (NIDS) analyzes network traffic and monitors multiple hosts to identify intrusions. Once an attack is identified, or abnormal behavior is sensed, the alert can be sent to the administrator. Finally, a distributed IDS (DIDS) consists of multiple intrusion detection systems monitoring a large network, and all of them communicate with each other, or with a central server.

IDSs usually are built using two major techniques: signature-based detection (or misuse detection) and anomaly detection.

Signature-based detection attempts to define a set of rules (or signatures) that can be used to decide that a given pattern is an attack. Therefore, signature-based systems are able to attain high levels of accuracy and minimal number of false positives in identifying intrusions [16]. However, for unknown attacks it gives a very high false alarm rate. As a result, signature-based intrusion detection is

not sweet for detecting new attacks, even though a slight variation of known attack can deceive it while anomaly-based detection is able to search the abnormal traffic by comparing the actual behavior with the normal system behavior. Unlike to misuse detection method, anomaly detection method can efficiently detect unknown attacks; therefore, a low false alarm rate for unknown attacks can be obtained.

3 Related works

A number of approaches based on deep learning methodologies have been proposed and have proved its success in the intrusion detection field. Yin et al. [29] have performed a deep learning approach for intrusion detection using recurrent neural networks which is a kind of ANN that represents loops between layers and between nodes. Two experiments have been performed to study the performance of the model on the NSL KDD dataset for binary classification (normal, anomaly) and five-category classification (normal, DoS, R2L, U2R and Probe). Also a comparison between the performance of the RNN-IDS with an ANN, naive Bayesian, random forest, multilayer perceptron and support vector machine is also performed in both binary classification and multi-classification categories. The accuracy rate was better when using RNN than other machine learning techniques.

RNN is also has been chosen by Kim and Kim [9]; however, it has been improved by Hessian-free optimization.

Javaid et al. [8] have proposed a self-taught learning-based IDS, and STL is one of the most popular DL algorithms that consists of two phases. In the first step, the model learns a feature representation from a large collection of unlabeled data known as unsupervised feature learning (UFL); then, this learnt representation is applied to labeled data for classification. In their proposed model, the implemented STL was composed by sparse auto encoder for UFL and soft max regression (SMR) for classification. They verify the performance of their model on NSL KDD dataset. Their approach was based on using a separate datasets for training and testing. A comparison between SMR classifier preceded by the sparse auto encoder (the proposed STL) and SMR as a single classifier was performed. SLT achieved better accuracy rate than SMR for the binary classification; however, it achieved lesser precision. For the multi-classification, STL achieved a better accuracy than SMR.

Salama et al. [20] developed an IDS which combines the advantages of deep belief network (DBN) and support vector machine (SVM), and DBN is used to reduce the dimensionality of the features set and was followed by SVM to classify the intrusions. A comparison between SVM, DBN and their DBN-SVM model is performed: The result shows that using both DBN-SVM gives better classification accuracy than using SVM or DBN as a single classifier also, and it enhances the testing time due to data dimensionality reduction. The evaluation of the system was performed on NSL KDD dataset.

Chaurasia and Jain [2] proposed an ensemble intrusion detection system that combines two classifiers: k-nearest neighbors and artificial neural network. They used bagging technique. They compared the results in the case of using bagging technique and in the case of using ANN or K-NN as single classifiers: Bagging provides better accuracy and lower false-positive rate.

Sammany and Sharawi [21] developed an IDS using MLP with two hidden layers and three classes output neurons. However, this IDS is able to distinguish only two types of attacks (Neptune, Satan) from normal traffic.

Siddiqui and Farooqui [23] have proposed an IDS based on the combination of support vector machine and neural network.

Ugtakhbayar et al. [25] proposed a hybrid system that combines the advantages of both anomaly-based and signature-based techniques. The signatures-based detection consists of using Snort IDS to detect know intrusion in real time while anomaly-based detection consists of applying naive Bayes algorithm as classifier. Experiments were performed on KDD 99 and NUM15 datasets. Authors also utilized features selection process to reduce the number of features of the dataset from 41 to 25 using information gain technique. The proposed model evaluation results show that the accuracy rates are 97.5%.

Ren et al. [18] used K-means algorithm to prepare KDD 99 dataset before prediction. They used neural network for detection. Experimental results on KDD 99 dataset show that the proposed model gives almost 90% accuracy rate.

Sarnovsky and Paralic [22] proposed a hierarchical detection system based on the combination of machine learning with knowledge-based approaches in the form of ontology. After being evaluated on KDD 99 dataset, the model achieves the detection task with 97.5% accuracy rate.

Ding et al. [3] used an other deep learning algorithm, namely convolutional neural network (CNN), to detect attacks on networks. Experiments performed on KDD 99 dataset show a high accuracy rate: 99.84%.

Kumar et al. [10] used meanshift clustering algorithm to detect networking attacks. Meanshift clustering is an unsupervised machine learning algorithm based on finding the center of each group in the dataset by calculating the mean of all data points until convergence is met. Experiments on KDD 99 dataset present 81.2% accuracy.

On the other hand, many early researches were interested in multi-agent system-based IDSs, such as

Sadhasivan and Balasubramanian [19], who have combined the definition of adaptive rules and the responsibilities for each agent for anomaly and misuse-based detection (ARMA-IDS) in which a combination of data mining techniques (clustering and rules) with multi-agent system is performed. They created five agents: sniffer agent which capture the packets; filter agent which receives captured packets from the first agent and try to isolate the irrelevant packets; anomaly detection agent which uses clustering technique to identify the intrusions in the network; association rule-based agent which uses the association rule technique to identify the relationship between the selected features and traffic characteristics; and sequential rule-based agent which defines the usual and unusual patterns of normal traffic using sequential rules technique. Experimental results of the framework on KDD 99 dataset give a good accuracy rate with low false-positive rate and false-negative rate.

Lui et al. [11] developed an adaptive NIDS using data mining and five types of agents based on clustering, association rules and sequential rules approaches with the adaptive learning.

Riyad et al. [17] proposed a distributed IDS using multi-agent system approach. The MAS used in this work is composed of four types of agents: (1) coordinator agent which pass information the agents of the network; (2) sniffer agent which collects the data; (3) filtering agent whose role is preprocessing the data collected by the sniffer agent; and (4) analysis and detection agent which analyzes the data and detects attacks if exists using a number of classifiers. Experimental results show a good accuracy rate: in average 95.8%.

4 The proposed solution

The main drawback of the existing IDSs is their central architecture, and this leads to a single point of failure. Furthermore, centralized IDSs usually fail in distributed types of attacks such as DDoS (distributed denial of services) [17]. For this reason, several recent studies were directed toward distributed systems, e.g., multi-agent systems, to build IDSs. Such architecture gives the system more robustness so that the fault tolerant becomes important because an agent can substitute another, and also the system becomes easily scalable since the number of agents can be easily increased if needed. In addition, the analysis of the data can be achieved in parallel, and this reduces the time considerably. On the other hand, distributed IDSs suffer from many problemes, e.g., false-positive rates, low efficiency, etc. [13], because most of them are signature based so they are able to detect only previously known attacks [11]. To tackle this problem, we propose a distributed IDS that combines the advantages of multi-agent approach with the high accuracy of deep learning algorithms.

The choice of deep learning with multi-agent methodology was for many reasons: firstly, because Intrusion detection is usually equivalent to a binary or multi-classification problem, i.e., identifying whether network traffic behavior is normal or not [29]. Therefore, deep learning is very sweet to attack detection problem, especially with its generalization feature; neural networks could be a good solution for detection of known as well as unknown attacks unlike traditional IDSs which are usually signatures based.

Also, artificial neural networks (ANNs) are the most commonly used approaches in intrusion detection systems and it surpasses traditional methods. Moradi and Zulkernine [12] and Yin et al. [29] because it is a powerful tool in multiple class classification [21].

Furthermore, deep learning allows feature selection which helps in the elimination of redundant features and noises and extracts a subset of relevant features of the traffic dataset to enhance classification results [4].

In case of using multi-agent system, deep learning allows agents to be more intelligent and adaptive: It increases significantly the detection rate and the accuracy of detection, and it allows them to learn a new pattern of attacks.

Overall, the contribution of this work consists of combining DL and MAS approaches to create an intelligent and distributed IDS. This idea will cure the flaws of both machine learning-based IDSs and multi-agent system-based IDSs and decrease considerably the detection time. We leverage the intelligent algorithms of DL to provide intelligence to the agents of MAS and . Thus, our solution consists of building a distributed IDS that integrates the desired features of MAS approach with the performance and exactitude of DL.

Hence, our IDS is composed of a number of agents that implanted three algorithms, namely autoencoder (AE), multilayer perceptron (MLP) and k-nearest neighbor (K-NN). Autoencoder performs the feature reduction task, and MLP and K-NN perform the classification task. KDD 99 benchmark dataset is used to evaluate the proposed model.

5 DL-MAFID scheme

Deep learning approach is largely used for intrusion detection system; in this paper, we used three algorithms: autoencoder, multilayer perceptron and k-nearest neighbors. Combining two or more algorithms is a technique used in many previous works. The contribution of this

work is that in addition to implementing more than one DL algorithm, we used multi-agent approach to build our IDS.

5.1 The deep learning-based IDS scheme

Our model DL-MAFID is composed of three main phases: preprocessing phase, feature reduction phase and classification phase.

5.1.1 Data preprocessing

There are three categories of feature in the KDD 99 dataset. The first type is a symbolic feature (e.g., protocol type, service and flag). The second type is a binary feature (e.g., land, logged_in and root_shell) and the numerical features (Table 1) . Therefore, the dataset should be prepared before use.

KDD CUP99 dataset preprocessing contains four processes:

1. Convert symbolic features to numerical values: Numericalization is necessary since the feature vector fed to the input of the neural network must be numerical.
2. Removing attributes with missing data.
3. Data scaling: The data have large varying ranges, so they have to be normalized. The normalization range used in this paper is from − 1 to 1. z score is used for normalization task:

$$z = \frac{x - \mu}{\sigma}$$

where:
μ is the mean of the data.
σ is the standard deviation of the data.

4. The class attribute is in binary format (normal or a specific kind of attack). However, in this work, a five-classification model is performed. Therefore, we assign attack names to one of the five classes, 0 for DoS (denial of service), 1 for Probe, 2 for R2L (remote to local), 3 for U2R (user to root) and 4 for normal.

The total number of features after performing the above preprocessing steps becomes 120.

Table 1 Features with different data types in KDD 99 dataset

Feature type	Features
Nominal	2 ,3, 4
Binary	7, 12, 14, 15, 21, 22
Numeric	1 ,5, 6, 8, 9, 10, 11, 13, 16, 17, 18, 19, 20, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41

5.1.2 Autoencoder feature reduction

In this work, autoencoder reduces efficiently the dimension of the dataset from 120 to only 10. It is composed of one input layer with 120 nodes representing the feature vector obtained after preprocessing phase, three hidden layers with 80, 40 and 20 nodes, respectively, composing the encode part, also the bottleneck layer which is the middle layer whose output represents the new reduced data that will be used in classification phase. Bottleneck layer is composed of ten nodes representing the reduced dataset. The decode part is also composed of three hidden layers which are symmetrical to those of encode parts with 20, 40 and 80 nodes, respectively. Finally, the output layer is composed of 120 nodes representing the reconstructing data. Figure 3 depicts the structure of our autoencoder.

5.1.3 Intrusion classification

1. *MLP classification* MLP is a kind of deep neural network consists of an input layer, one or more hidden layers and an output layer. MLP in this work is able to well distinguish attacks pattern from benign and even recognize attack type (DoS, Probe, U2R, R2L). We build a four-layer MLP classifier composed of an input layer with ten nodes representing the relevant features resulting from feature reduction phase, three hidden layers with 20, 15 and 20 nodes, respectively, and the output layer with five nodes representing the five classes: the four attack types and the normal class. Figure 4 depicts the structure of our MLP.
2. *K-NN classification* Since we build a distributed IDS, using one classifier is not enough, we need a number of classifiers in such way each classifier is located in a different segment in the network; if one classifier fails, other classifiers can give better results.

   To increase the effectiveness and robustness of the IDS, we are not limited to one solution; therefore, we used another machine learning classifier based on k-nearest neighbor algorithm.

   After several experiments and fine-tuning of parameter K, we conclude that for our case the best value giving the highest accuracy rate is $K = 5$.

5.2 Multi-agent IDS scheme

Several works have been proposed using MAS approach in the intrusion detection field such as in [4, 11, 17, 19]. The use of MAS enables taking advantage of some of the properties of agents such as re-activity, pro-activity and sociability and makes the task of intrusion detection more robust, faster and easier since the tasks are shared between several agents in the system.

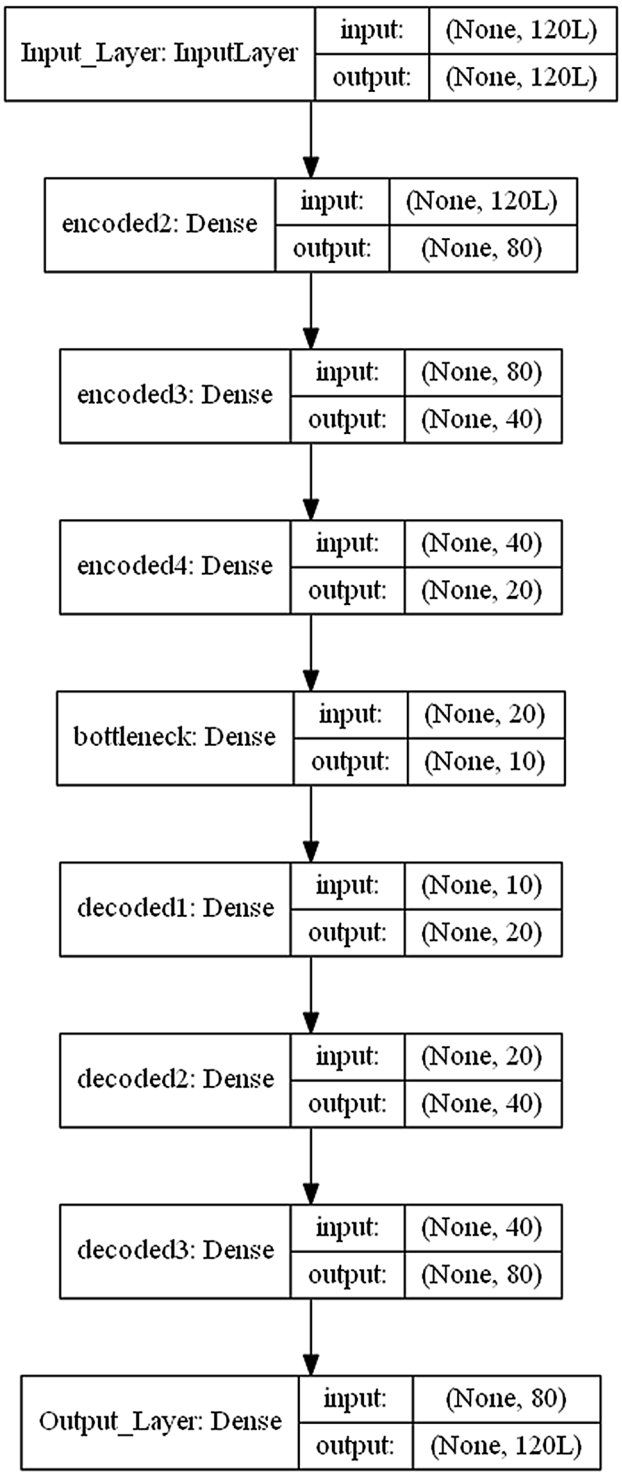

Fig. 3 Structure of the proposed autoencoder

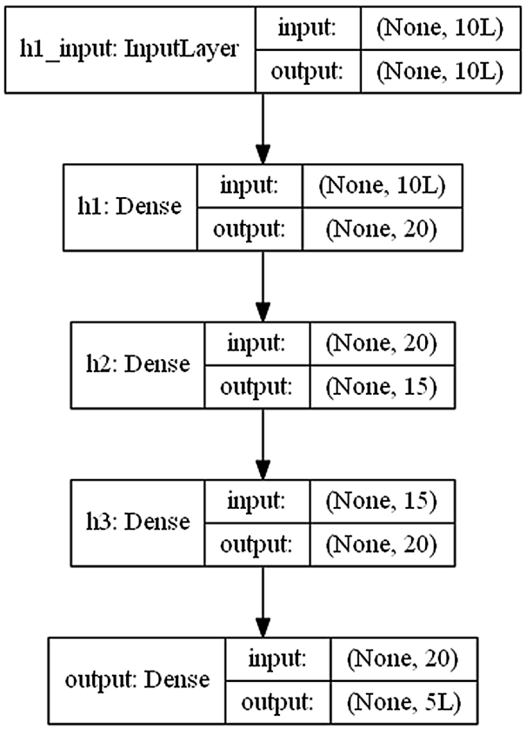

Fig. 4 Structure of the proposed MLP

However, generally previous works used signature-based technique with which the system does not enable to detect unknown or a variation of known attacks [6].

In this work, we use deep learning algorithms which are known for their effectiveness in detecting unseen attacks. In the case of using the multi-agent system based on deep learning, we make the agents more intelligent because they used intelligent DL algorithms to detect the known attacks and more adaptive because deep learning allows them to learn and adapt themselves in such a way they can detect unknown attacks. The proposed deep learning-based multi-agent for IDS is composed of four types of agents:

- Preprocessor agent
- Reducer agent
- Agent classifier
- Decision-maker agent

5.2.1 Preprocessor agent (PA)

The task of the agent preprocessor is to pre-process the data as already explained in Sect. 5.1.1 and shown in Fig. 6.

This agent can be extensible so it can be adapted to an online version, and then, it will have to collect the data end information about the traffic of the network. After preparing the data, PA sent it in a format accepted by a neural network to the agent reducer.

5.2.2 Reducer agent (RA)

Agent reducer executes the autoencoder algorithm to reduce the dimension of the data. The structure of our autoencoder is represented in Sect. 5.1.2 and Fig. 3. Then, the reduced data are sent to the agents classifier to perform the classification task.

5.2.3 Agents classifier (CA)

The number of the agents classifiers can be more than 1 (from 1 to n) located on the same node of the network; each one builds its model; and once the model is built, it can be used for next predictions. Finally, each agent sends its result to the decision-maker agent. In experiments, two agents classifiers are built: K-NN agent and MLP agent. Since we made experiments on an off-line traffic using the KDD 99 data benchmark dataset, we assume that both K-NN and MLP agents are situated in the same location in the network and that the test set representing the unknown data is the real network traffic. K-NN runs the k-nearest neighbors algorithm explained and MLP agent runs the multilayer perceptron algorithm. Both K-NN and MLP algorithms are detailed in Sect. 5.1.3. They make their predictions and send their results with the values of the accuracy rate as a justification to the decision-maker agent.

5.2.4 Decision-maker agent (DA)

DA asks periodically for the decisions of agents classifiers. Once they are received, DA analyzes the situation: If one CA sent, its prediction for a given location; DA takes this prediction as a final decision. In the case of the existing of several CA in the same location, the DA compares the results: If they are identical, DA takes it as a final decision; if there is a difference, DA takes the prediction having the highest accuracy rate as a final decision.

Note that in each segment of the network, we have to find one PA, one RA and at least one CA which send periodically its report (results of prediction) to the DA to give a conclusion (final decision) about the state of the system (Fig. 5).

The overall workflow of the proposed IDS is shown in Fig. 6 and described as follows:

Step 1: PA receives the dataset (41 features). In case of an online traffic, it has to collect the data.
Step 2: PA converts symbolic features to numerical values.
Step 3: PA removes attributes with missing values.

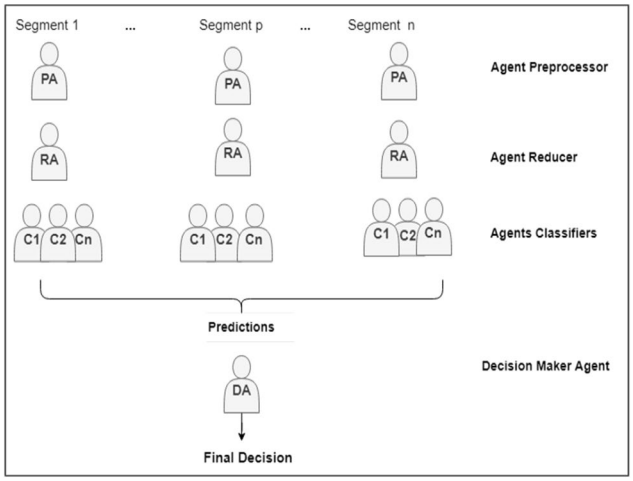

Fig. 5 Agents architecture in the network

Step 4: PA scales the data.
Step 5: PA sends the preprocessed data (120 features) to the reducer agent (RA).
Step 6: RA reduces the data from 120 features to 10 features using autoencoder algorithm.
Step 7: RA sends the reduced data to the agents classifier (CA).
Step 8: If first time, MLP and K-NN classifiers enter in training phase in order to build their own models. Else, both use their built models to make predictions on new patterns (testing set).
Step 9: Agents classifier sent the results of prediction with the accuracy rates to the decision-maker agent (DA).
Step 10: DA analyzes the situation:
if predictions are received from one agent in a given location then: final decision = this prediction.
Otherwise,
if agents are agree then: final decision = prediction.
else: final decision = prediction having the maximum accuracy rate.

6 Experimental results

6.1 KDD CUP99 dataset

To evaluate the performance of our proposed approach, we present tests on KDD CUP99 anomaly intrusion detection dataset. KDD 99 dataset, created in 1999, is widely used in academic research, especially machine learning research and intrusion detection systems. KDD 99 is publicly available and is considered benchmark dataset for testing of intrusion detection algorithms. It has labeled attack samples which are obtained by passive monitoring.

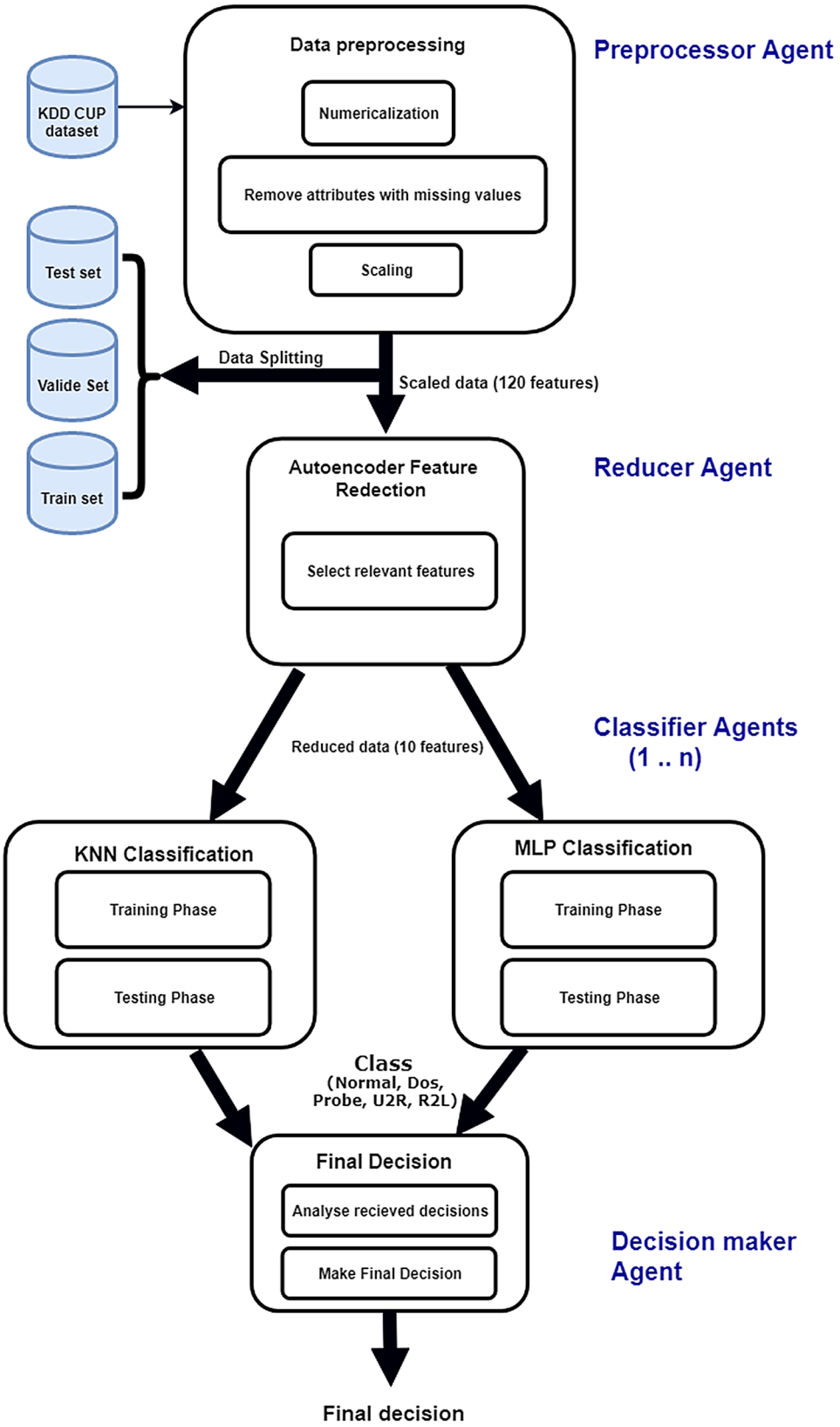

Fig. 6 The structure of DL-MAFID

Table 2 Number of samples for each class

Class	Counts
Dos	391458
Probe	4107
R2L	1126
U2R	52
Normal	97278

Table 3 Types of KDD 99 dataset

Dataset	Number of records
Corrected KDD	311 029
10-percent Corrected KDD	494 021
Whole KDD	4 898 430

Each record of the dataset is labeled as either normal or a specific kind of attack (Table 2). The attacks can be classified in one of the four known categories, namely denial of service (DoS), user to root (U2R), remote to local (R2L) and Probe.

- **Denial of service (DoS):** This kind of attack deprives legitimate users of the service or resource they expected. DoS attacks accomplish this by flooding the target with traffic and requests in order to make the resources too busy so that the system becomes overloaded, e.g., back, land, Neptune.
- **Remote to user attacks (R2L):** In this type of attack, an intruder sends packets to a computer through the Internet so that the machine's vulnerabilities exposes, and thus, it could exploit the privileges of the user.
- **User to root attacks (U2R):** In this type of attack, a hacker holds the account and password information of an authorized user and can own the privilege of access to the whole system, e.g., loadmodule, perl, rootkit.
- **Probing (Probe):** The hacker scans computer in order to determine a weak point through which it gain access to the system.

Table 3 presents the different types of KDD 99 dataset. In this paper, 10-percent corrected KDD is used. It has 494 021 records, in each record, there are 41 attributes describing different features of the data, and the 42nd attribute contains label which assigns to each record either an attack type or normal.

6.2 Experiments and analysis

We take an experiment for measuring the performance of our model. We developed IDS using Python programming language.

Our experiment environment is as follows:

- CPU: Intel i3, 2.20GHZ
- GPU: GPU acceleration not used
- RAM: 4GB
- OS: Windows 10

To evaluate the performance of a model, there are numbers of performance metrics such as accuracy rate, detection rate, false-positive rate (FPR) and true-positive rate (TPR).

In the proposed model, we have taken accuracy rate, precision, TPR, TNR, FPR, FNR and detection rate. Also, we back to the confusion matrix to get our metrics. The confusion matrix is composed of true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

- *True positive* legitimate attack and IDS gives alarm
- *True negative* no attack and IDS gives no alarm
- *False positive* no attack and IDS gives alarm
- *False negative* legitimate attack and IDS gives no alarm

All metrics are calculated using the following formulas:

$$\text{FPR} = \frac{\text{FP}}{(\text{FP} + \text{TN})}$$

$$\text{FNR} = \frac{\text{FN}}{(\text{FN} + \text{TP})}$$

$$\text{Recall or TPR} = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$

$$\text{Specificity or TNR} = \frac{\text{TN}}{(\text{TN} + \text{FP})}$$

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{\text{TP} + \text{TN} + \text{FN} + \text{FP}}$$

$$\text{Detection rate} = \frac{(\text{TP})}{\text{TP} + \text{TN} + \text{FN} + \text{FP}}$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

The results show that both proposed MLP and K-NN classifiers are performing; however, K-NN agent is a bit better, and it achieves its task with lower values of FPR and FNR and higher values of accuracy, precision, TPR and TNR. For example, values of accuracy are 99.73% for MLP agent versus 99.95% for K-NN agent.

Also, the time of both training and testing phases is considerably reduced due to the reduction in the dimensionality of the data which was performed by the agent reducer using autoencoder algorithm. Agent reducer achieved its task with good accuracy rate: 89.42%.

Figures 7 and 8 show two confusion matrices. The first matrix shows the result of the model using only MLP

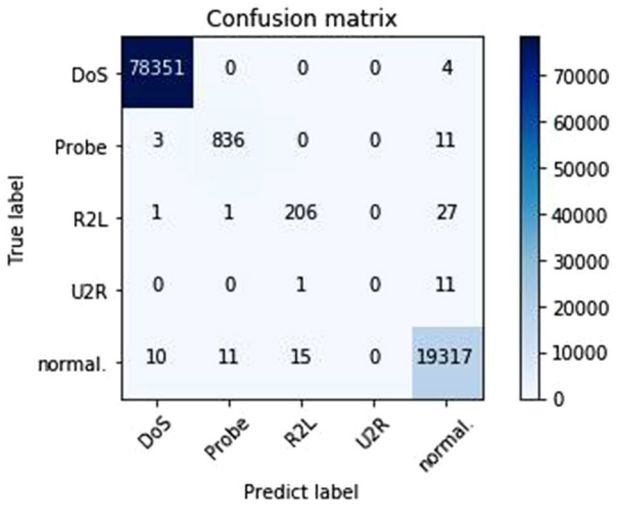

Fig. 7 Confusion matrix (model without autoencoder)

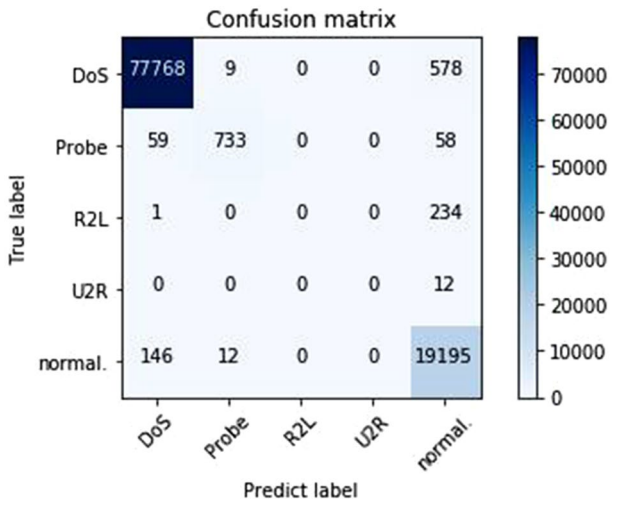

Fig. 8 Confusion matrix (model using autoencoder)

algorithm performed by the MLP agent (without the contribution of the reducer agent); however, the second one shows the result of the model using both autoencoder and MLP algorithms.

We observe that the metrics in the first confusion matrix (Fig. 7) are a bit better than the second matrix (Fig. 8), and this is due to the fact that the data has been compressed by the autoencoder. However, the use of autoencoder allows us to benefit from the advantages of dimensionality reduction, such as removal of irrelevant and redundant features and the reduction in both training and testing time, and also this difference does not affect the accuracy rate of the classification of MLP agent.

Early stopping criterion for validation set was applied by both reducer agent and MLP agent classifier to stop the training process and prevent the model from over-fitting (Sect. 5.1.3).

Figure 9 describes the evaluation of the loss function during the training phase of MLP agent with respect to the progress of training epochs for the training and validating sets. In this work, we used categorical cross-entropy loss function. Furthermore, to more evaluate our solution, we compare it with other related works in which either ML and DL algorithms or MAS are leveraged. This comparison is based on accuracy rate. Since we evaluate our model on KDD 99 dataset, we choose to compare results with some existing (presented in Sect. 3) which used the same dataset or its extension NSL-KDD dataset.

Table 4 shows that our intelligent distributed IDS based on combining DL and MAS outperforms others approaches in terms of accuracy rate also, and Table 5 shows that our system achieves the detection task with very small false alarm rate. This proves the efficiency of our solution.

7 Conclusions

DL-MAFID model is aimed to solve a multi-class problem of intrusion detection using multi-agent system based on deep learning in which not only the attack records are distinguished from normal ones, but also the attack type is recognized. We have used three algorithms: autoencoder for dimensionality reduction, which provides a good result in this task, reduces efficiently the dimensional of KDD CUP99 dataset to nearly 91% of its original size, and then, it was followed by two classifiers, namely K-NN and MLP classifiers, to classify the reduced data into five classes, i.e., normal or one of the main types of networking attacks (DoS, R2L, Probe and U2R). Also, we benefited from the desired features of MAS approach which make our IDS

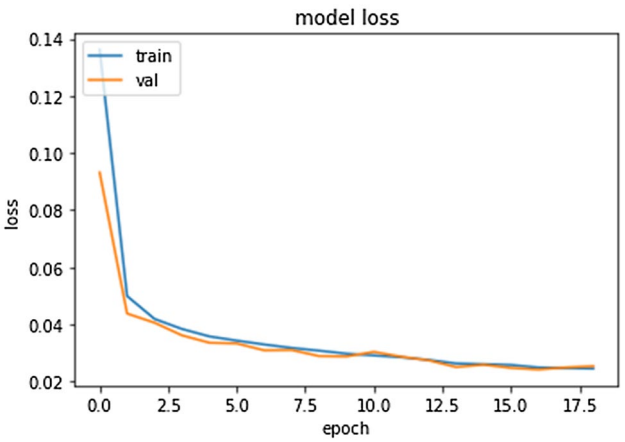

Fig. 9 Categorical cross-entropy loss function

Table 4 Comparison between the intelligent distributed IDS using MAS based on DL with other previous approaches on KDD 99 or NSL KDD datasets

References	Approach	Dataset	Accuracy rate (%)
[20]	Deep belief network + support vector machine	NSL KDD	92.84
[15]	Naive Bayes	KDD 99	95
[29]	Recurrent neural network	NSL KDD	91.29
[8]	Self-taught learning	NSL KDD	79.10
[19]	Multi-agent system + data mining	KDD 99	97.47
[25]	Combining anomaly-based (naive Bayes) and signature-based detection (Snort system)	KDD 99 + NUM15	97.5
[18]	K-means + artificial neural network	KDD 99	97.5
[22]	Combining ML with ontology	KDD 99	99.8
[3]	Convolutional neural network K-NN	KDD 99	99.84
[10]	Meanshift clustering algorithm	KDD 99	81.2
[17]	Multi-agent system	KDD 99	95.82
Our model	Deep learning-based multi-agent system	KDD 99	99.95

Table 5 Metrics of both K-NN and MLP classifiers

	FPR (%)	FNR (%)	TNR (%)	TPR (%)	Acc. (%)	DR (%)	Precision (%)
K-NN	0.03	0.12	99.97	99.88	99.95	99.88	99.88
MLP	0.17	0.68	99.83	99.32	99.73	99.32	99.32

more robust and more efficient than using a monolithic system. Experimental results show that the proposed system is capable of classifying records with 99.95% accuracy rate.

For future work, we plan to:

- Apply this solution in a real network traffic (i.e., online IDS).
- Benefit from other machine leaning algorithms by adding more agents classifiers to increase the accuracy of intrusion detection.
- Take more advantages of MAS such as mobility and cloning.
- Extend our IDS so that it can be used with cloud computing, fog computing and Internet of things for security purpose.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Anderson JP (1980) Computer security threat monitoring and surveillance. Technical report, James P. Anderson Company, Fort Washington
2. Chaurasia S, Jain A (2014) Ensemble neural network and K-NN classifiers for intrusion detection. Int J Comput Sci Inf Technol 5:2481–2485
3. Ding N, Liu Y, Fan Y, Jie D (2020) Network attack detection method based on convolutional neural network, chapter, vol 68. Springer, Berlin, pp 610–620
4. Erlank AO, Bridges CP (2017) A hybrid real-time agent platform for fault-tolerant, embedded applications. Auton Agents Multi-Agent Syst 32(2):252–274
5. Glavic M (2006) Agents and multi-agent systems: a short introduction for power engineers. Technical report, University of Liege, 4000 Liege, Belgium
6. Herrero Á, Corchado E (2009) Multiagent systems for network intrusion detection: a review. In: Herrero Á, Gastaldo P, Zunino R, Corchado E (eds) Advances in intelligent and soft computing, vol 63. Springer, Berlin, pp 143–154
7. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313:7
8. Javaid A, Niyaz Q, Sun W, Alam M (2015) A deep learning approach for network intrusion detection system. In: BICT'15 proceedings of the 9th EAI international conference on bio-inspired information and communications technologies (formerly BIONETICS), pp 21–26, New York City, United States
9. Kim J, Kim H (2015) Applying recurrent neural network to intrusion detection with hessian free optimization. In: 16th international workshop, WISA 2015, pp 357–369
10. Kumar A, Glisson W, Cho H (2020) Network attack detection using an unsupervised machine learning algorithm
11. Lui C-L, Fu T-C, Cheung T-Y (2005) Agent-based network intrusion detection system using data mining approaches. In: Third international conference on information technology and applications (ICITA'05), Sydney, NSW, Australia. IEEE
12. Moradi M, Zulkernine M (2004) A neural network based system for intrusion detection and classification of attacks. In: Proceedings of 2004 IEEE international conference on advances in intelligent systems

13. Mukhin V, Kornaga Y, Steshyn V, Mostovoy Y (2016) Adaptive security system based on intelligent agents for distributed computer systems. In: International conference of development and application systems (DAS), pp 320–325
14. Nelwamondo FV, Golding D, Marwala T (2009) A dynamic programming approach to missing data estimation using neural networks. Inf Sci 237:49–58
15. Panda M, Patra MR (2007) Network intrusion detection using Naïve Bayes. Int J Comput Sci Netw Secur 7(12):258–262
16. Patel HJ, Patel R (2014) A survey on intrusion detection system in cloud) based on data mining. Int J Eng Techn Res 2:5
17. Riyad AM, Irfan Ahmed MS, Raheemaa Khan RL (2019) An adaptive distributed intrusion detection system architecture using multi agents. Int J Electr Comput Eng 9(6):4951–4960
18. Ren B, Hu M, Yan H, Yu P (2019) Classification and prediction of network abnormal data based on machine learning. In: International conference on robots and intelligent system (ICRIS). IEEE
19. Sadhasivan DK, Balasubramanian K (2017) A fusion of multiagent functionalities for effective intrusion detection system. Secur Commun Netw
20. Salama MA, Eid HF, Ramadan RA, Darwish A, Hassanien AE (2011) Hybrid intelligent intrusion detection scheme. Soft computing in industrial application, vol 96. Springer, Berlin, pp 293–303
21. Sammany M, Sharawi I Saroit M, El-Beltagy M (2011) Artificial neural networks architecture for intrusion detection systems and classification of attacks. Technical report, Faculty of Science, Cairo University, Egypt
22. Sarnovsky M, Paralic J (2020) Hierarchical intrusion detection using machine learning and knowledge model. Symmetry 12:203
23. Siddiqui AK, Farooqui T (2017) Improved ensemble technique based on support vector machine and neural network for intrusion detection system. Int J Online Sci 3:12
24. Templeton G (2017) Google reveals automatic machine learning: A.i. can create itself. https://www.inverse.com/article/31952-ai-google-machine-learning-automl
25. Ugtakhbayar N, Usukhbayar B, Baigaltugs S (2020) A hybrid model for anomaly-based Intrusion detection system, chapter 44. Springer, Berlin, pp 419–431
26. Wooldridge M (2009) An introduction to multiagent systems, chapter preface. Willey, Hoboken
27. Wooldridge M, Jennings NR (1995) Ecai-94 proceedings of the workshop on agent theories, architectures, and languages on intelligent agents. In: Agent theories, architectures and languages: a survey, Amsterdam, The Netherlands. Springer, Berlin, pp 1–39
28. Wooldridge M, Jennings NR (1995) Intelligent agents: theories, architectures and languages, vol 890. ACM Digital Library, New York
29. Yin C, Zhu Y, Fei J, He X (2017) A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5:7
30. Zakka K (2016) A complete guide to k-nearest-neighbors with applications in python and r. https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/