Acta Scientiarum

http://periodicos.uem.br/ojs
ISSN on-line: 1807-8664
Doi: 10.4025/actascitechnol.v45i1.60551

Combined Ensemble Intrusion Detection Model using Deep learning with Feature Selection for Fog Computing Environments

Kalaivani Kaliyaperumal[1*], Chinnadurai Murugaiyan[2], Deepan Perumal[3], Ganesh Jayaraman[4] and Kannan Samikannu[2]

[1]School of Computer Science and Engineering, Vellore Institute of Technology, Vellore,Tamilnadu, 632014, India. [2]Department of Computer Science and Engineering, E.G.S. Pillay Engineering College, Nagapattinam, Tamilnadu, India. [3]Department of Computer Science and Engineering, St. Martin's Engineering College (Autonomous), Secunderabad, Telangana, India. [4]Department of Computer Science and Engineering, Anjalai Ammal Mahalingam Engineering College, Kovilvenni, Tamilnadu, India. *Author for correspondence. E-mail: akilkalai12@gmail.com

ABSTRACT. Decentralized architecture known as fog computing is situated between the cloud and data-producing devices. It acts as a conduit between cloud services and IoT devices. In order to reduce latency, fog computing can handle a significant amount of computation for time-sensitive IoT applications. The Fog layer is simultaneously vulnerable to numerous assaults. To defend the fog nodes from attacks, fog computing paradigms may be suited for deep learning-based intrusion detection systems (IDS). In this paper, a combined Ensemble Intrusion Detection Model using Deep learning with Efficient Feature Selection using Random forests is proposed for Fog Computing Environments by using two deep learning models of traditional CNN and IDS-AlexNet model called Ensemble CNN-IDS with Random Forest and showed this model gives high accuracy of attack detection. The respective model implementations demonstrated on the UNSW-NB15 dataset that consists of 9 classes of attacks namely Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcodes and Worms. The proposed combined Ensemble Intrusion Detection Model using Deep learning with Efficient Feature Selection for intrusions detection is shown to be accurate and efficient by using different classifiers. Our proposed model provides high the accuracy in attack detection of about 97.5% that it outperformed various other traditional and recent models.

Keywords: fog computing, deep learning, IDS, CNN, AlexNet, UNSW-NB15.

Received on August 14, 2021.
Accepted on May 11, 2022.

Introduction

IoT is becoming a key technology in daily life. Internet of Things (IoT) consists of large number of networked physical devices such as computers, vehicles, digital devices, sensors etc (Ray, 2018) They transmit the data to the cloud server, and the cloud responds. Time-sensitive and time-insensitive applications can take many different forms. Applications that require a quick response may suffer from the cloud server's delayed response. Fog computing extends cloud computing, which offers services to end users including data, storage, processing, and other things, in order to solve the aforementioned problem. (Yousefpour et al., 2019; Puliafito, Mingozzi, Longo, Puliafito, & Rana 2019). Fog layer serves as an intermediate layer between Cloud and IoT layer. The heterogenous devices are present at IoT layer. The fog computing paradigm is depicted in Figure 1 and is made up of three layers: IoT devices at the bottom, fog computing as an intermediate layer, and clouds at the top. Thus, time sensitive applications of health care, augmented reality and etc. get benefits by fog computing. Fog node can send immediate response to the end devices which gives low latency.

Like the cloud environment, fog layer is also vulnerable to various attacks (Yi, Qin, & Li, 2015). Therefore, having an effective intrusion detection system that includes a variety of techniques and tools is essential. They keep an eye on network activity as well as the computer system. In order to safeguard fog nodes and clouds, it should be able to analyse activities to find potential internal and external intrusions targeting the system in the fog layer. The IDS can be deployed on the fog layer to detect intrusive behaviour and malicious attacks such as denial-of service (DoS), Backdoors and so on. The IDSs are responsible for detection and prevention of attacks. In order to provide low latency through fog layer, it is really a challenge to implement

efficient intrusion detection system. Therefore, a subset of machine learning called deep learning aids in the implementation of intrusion detection systems. Deep learning algorithms use layers to carry out their operations and are made up of several successive interconnected layers (Tu et al., 2018; Zhang, Yang, Chen, & Li, 2018). The following layer's input comes from the output of the previous layer.

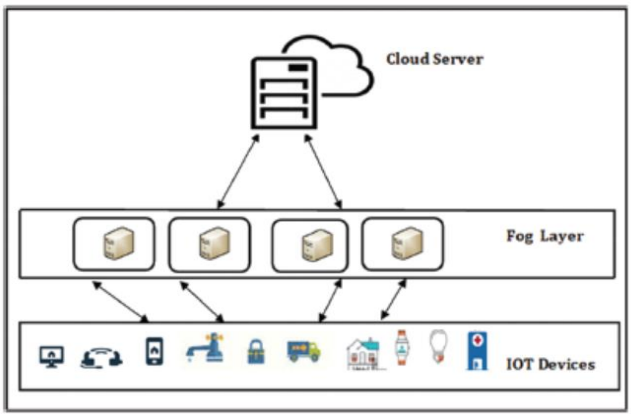

Figure 1. Fog Computing Paradigm.

Deep learning has proven to be effective in a variety of fields, including image and video recognition, audio processing, natural language processing and others. Intrusion detection systems are implemented using machine learning techniques. Due to the development of new technologies and the volume of Internet traffic, shallow machine learning techniques struggle to handle security difficulties and are ineffective when dealing with attack scenarios. Similar to other sectors, the intrusion detection system has demonstrated the effectiveness of deep learning algorithms. Deep networks are able to quickly identify assaults by identifying correlations in network traffic that includes both regular and aberrant records. There is no requirement for human involvement. Deep learning techniques combine feature extraction with classification. The various machine learning methods and deep learning models that were used for network intrusion detection system. In order to accomplish the function of the query, Bu and Cho (2020) proposed a convolutional neural-based learning classifier system that combines traditional learning classifier system with one-dimensional CNN based on the RBAC mechanism. For feature selection rules optimization and classification, the modified Pittsburgh-style LCSs and CNNs were combined. On a simulated query dataset, their work beat previous machine learning techniques.

For WSNs, Butun, Morgera, and Sankar (2014) conducted a survey on intrusion detection systems. They included comprehensive data regarding IDSs for Mobile Ad-Hoc Networks in their survey (MANETs). Each strategy was examined for both its benefits and disadvantages. Their study has revealed problems with the field's research. Multilayered Echo-State Machine, a model for effective intrusion detection that uses RNN, was proposed by Tchakoucht and Ezziyyani (2018). On the KDD'99, NSLKDD, and UNSW NB 15 datasets for binary and multiclass classification, they applied their approach. It was evident during evaluation that the ML-ESM had accomplished a greater accuracy and required less processing time. Akbar, Rao, and Hussain, (2016) introduced Intrusion Prevention Systems (IPS), which use an improved genetic algorithm to detect intrusions in real time and block them. They made use of the KDD Cup Dataset, which was divided between learning and detection phases. Their suggested methodology laid forth a set of guidelines and produced positive outcomes.

A hybrid malicious code detection system based on auto encoder and DBN (Deep Belief Networks) was proposed by Li, Ma, and Jiao (2015). While DBN learning approach discovered dangerous code, the Auto encoder deep learning method reduced the dimension of data. They employed multiple layers of DBN. Finally, the temporal complexity was decreased and improved detection performance on the KDD Cup data set through the optimal hybrid greater detection accuracy. Li, Qin, Huang, Yang, and Ye (2017) utilized the uses of convolutional neural networks (CNN) in intrusion detection. They used NSL-KDD dataset and proposed image conversion method for CNNs. The graphic conversion technique learns the features of NSL-KDD automatically. CNN has performed well on NSL-KDD datasets. A NIDS based on a Hidden Naive Bayes (HNB) multiclass classifier was proposed by Koc, Mazzuchi, and Sarkani (2012). The conditional independence assumption of the Naive Bayes approach is lessened by the HNB data mining model. It outperformed the traditional Naïve Bayes model on KDD Cup data set.

Aydın, Zaim, and Ceylan (2009) proposed a hybrid IDS using two approaches such as packet header anomaly detection (PHAD) and network traffic anomaly detection (NETAD). The hybrid IDS is evaluated using the MIT Lincoln Laboratories network traffic data (IDEVAL) as a test bed. Evaluation shows that the hybrid IDS is a more powerful system. A RNN-based deep learning model for intrusion detection was proposed by Yin, Zhu, Fei, and He (2017). (RNN-IDS). Their model is applied to the classification of binary and multiclass attacks. Using a benchmark data set, their suggested model's performance is compared against machine learning models that have previously been proposed by other academics. High precision is provided by the RNN-IDS. Support vector machine (SVM) was employed by Al Mehedi Hasan, Nasser, and Pal (2013) to create an efficient and reliable IDS approach. On the KDD'99 Dataset, they have so utilised various kernels. To get around the KDD'99 dataset's data redundancy issue, they have created two additional data sets called KDD99Train+ and KDD99Test+. RBF kernel experimental findings outperform other outcomes in terms of detection rate. In order to defend services from attackers, Wahab, Bentahar, Otrok, and Mourad (2019) proposed a Bayesian Stackelberg game comprises risk assessment framework that identified the danger levels of the VMs and defence mechanisms based on live migration. In their work, honey pots are utilised to gather harmful data from VMs and machine learning techniques are applied. One-class Support Vector Machine is used to learn the type distributions of the attackers. Amazon's data centre was used for experiments. The detection rate was increased by their suggested remedy.

Alom, Bontupalli, and Taha (2015) suggested an IDS based on Deep Belief Neural (DBN) Networks. They performed a number of experiments with the NSL-KDD dataset to examine DBN's capabilities for identifying intrusions. Scale-Hybrid-IDS-AlertNet, a deep neural network-based system, was created by Vinaykumar et al. in 2019. (SHIA). Based on that, it has identified and categorised unanticipated and unpredictable cyberattacks. Rajesh Kanna et al..2021 have developed unified model of Optimized CNN (OCNN) and Hierarchical Multi-scale LSTM (HMLSTM) for effective extraction and learning of spatial–temporal features. In the OCNN–HMLSTM model, the Lion Swarm Optimization (LSO) is used to tune the hyper-parameters of CNN for the optimal configuration of learning spatial features. The HMLSTM has learnt the hierarchical relationships between the different features and extracts the time features. Experiments were demonstrated over public IDS datasets namely NSL-KDD, ISCX-IDS and UNSWNB15.

Also, for the sake of simplicity Table 1 summarizes research works of IDS with the details of used dataset, achieved Accuracy, their detection rate and given False Alarm Rate of various other researchers. The experiments mentioned above show that network intrusion detection's recognition ability and performance have both improved. However, they struggle with low attack detection accuracy and a high number of false alarms. They must therefore be upgraded. It is necessary to design an efficient intrusion detection system for fog computing environment for detecting the intrusions with high accuracy and low false alarm rate and to conduct an experimental study on the developed IDS performance with the following evaluation metrics: Accuracy, Recall, Precision, F-Score, False Alarm Rate and Misclassification Rate.

The main contributions of our work are:

(i) The Deep learning approach is combined with Random forests model.
(ii) The idea is to combine IDS-AlexNet and conventional CNN to create a multiclass ensemble IDS. The suggested model may be able to automatically and intelligently learn and recognise assault features.
(iii) Random forests model is used for important feature selection.
(iv) Using the benchmarked intrusion detection dataset from UNSW-NB15, multiclass attack detection tests are conducted.
(iii) Experimental findings showed that our model has a low false alarm rate and good accuracy.
(v) Experimental results demonstrate that our suggested network intrusion detection model offers greater accuracy when compared to popular machine learning and deep learning models like SVM, kNN, and ANN.

Material and methods

Proposed intrusion detection approach

This section explains the Random Forest model, CNN, IDS-AlexNet and proposed framework of Feature selection with Random forests and Ensemble CNN-IDS. In the proposed framework, Random Forest model is used to select important features from the dataset. The selected features are given as the input to the ensemble model. that is built using IDS-AlexNet and conventional CNN. Traditional CNN and IDS-AlexNet

models are first generated separately, after which an ensemble model is made and intrusions are detected. The benchmark dataset namely University of New South Wales Network Based 2015 (UNSW-NB15) is used for experimentation. Models are constructed for multi-class classifications after data pre-processing and 2D transformations.

Table 1. IDS summarized works.

References	Model Used	Data set used for Experiment	Detection Rate	Accuracy	False alarm Rate
Pfahringer, 2000	bagged boosting	KDD cup	91.8	-	0.60
Beghdad, 2007	RNN	KDD-99	73.1	-	26.85
Ustebay, Turgut, and Aydin, 2018	MLP	CICIDS2017	-	94.5	-
Adebowale, Idowu, and Amarachi, 2013	K-Nearest Neighbour (k-NN) algorithm	NSL-KDD	94.59	-	-
Han and Cho, 2003	K-nearest neighbor	KDD-99	91	-	8
Devaraju and Ramakrishnan, 2014	Recurrent neural network with hessian-free optimization	KDD-99	95.37	2.1	-
Ahmed, Mahmood, and Hu, 2016	Expectation Maximization (EM) clustering	NSL-KDD	-	78	-
Beghdad, 2007	Jordan ANN	KDD-99	62.9	-	37.09
Ye, Emran, Chen, and Vilbert, 2002	Multivariate statistical analysis of audit data	KDDCUP-99	90	-	-
Tajbakhsh, Rahmati, and Mirzaei, 2009	Fuzzy association rules	KDD-99	91	-	3.34
Khraisat, Gondal, Vamplew, and Kamruzzaman, 2019	ANN analysis system calls	DARPA 98	96	-	-
SyarifI, Prugel-Bennett and Wills, 2012	k-medoids	NSL-KDD	-	76.71	-

Random Forest Model

A supervised model called Random Forest employs both bagging and decision trees. The concept is to resample the training dataset using the "bootstrap" method. Each sample is used to fit a decision tree and includes a random subset of the original columns. According to its capacity to improve the purity of the leaves, each tree in the random forest can estimate the significance of a trait. It's a subject pertaining to the operation of Classification And Regression Trees (CART). The significance of the characteristic increases with the increase in leaf purity. This is done for every tree, averaged across all the trees, and then set to 1. Consequently, a Random Forest's calculation of the importance scores as a sum equals 1.

Convolutional Neural Network (CNN)

One of the deep learning algorithms is the convolutional neural network (Deepan & Sudha, 2019). It has two components, as seen in Figure 2, which illustrates this. These are the parts for feature extraction and classification. In order to detect the features, a sequence of convolution operations and pooling operations are performed using hidden layers. The number of convolution and pooling layers in a CNN's structure might vary. Fully linked layers make up the classification component, which classifies extracted characteristics. By mathematically combining two functions, the phrase ‘convolution’ creates a third function. Convolution is conducted using a filter by swiping the filter over the input data. Every position does a matrix multiplication. The input characteristics of the feature map of the layer i is assumed as Mi (M_0=x). Using various window values, the various feature information is recovered from the data matrix M_{i-1}. Convolution is expressed as the following:

$$M_i = f(M_{i-1} \otimes W_i + b_i) \quad (1)$$

where f(x), Wi, $\otimes$ and bi represents activation function, convolution kernel vector of i layer, convolution operation and offset vector of i layer. The activation function passes the convolution output. A pooling layer

is inserted between CNN layers to minimise dimensionality, which in turn reduces the size of the network's parameters and the amount of processing needed. As a result, training time is cut down, and overfitting is also managed. The maximum value for each window is taken using the max pooling method. The size of the feature map is shrunk while maintaining the important data. The feature map is typically sampled by the pooling layer using various sampling rules. The output from the convolutional layer or final pooling is flattened. The fully linked layer, which learns the high-level features, receives it. Every training iteration uses the backpropagation method. In the last layer, the soft-max activation function is employed. The dominant features over a number of epochs can be categorised.

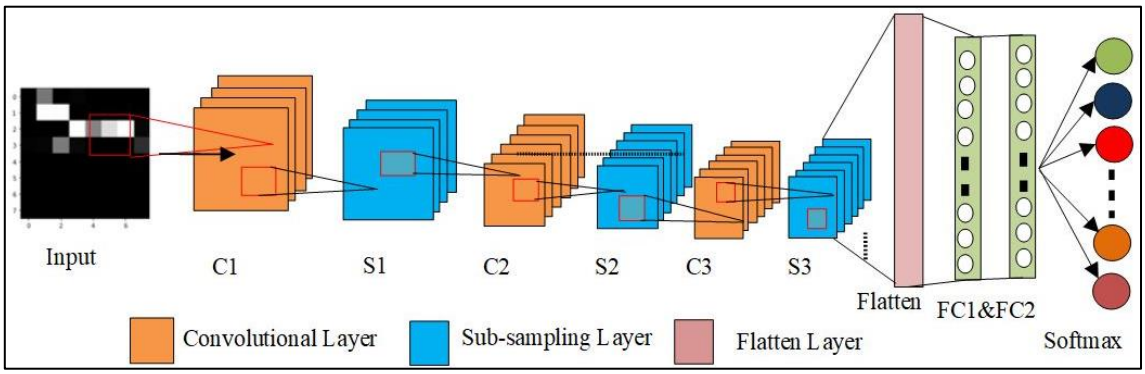

Figure 2. Architecture of CNN.

We have used three convolution layers, three subsampling layers, one flatten layer, two fully connected layer and an output layer in the traditional CNN. ReLU activation function is used in all the layers except last layer where softmax activation function is used. Adam optimization function is used to avoid overfitting.

IDS-AlexNet Model

Our proposed IDS-AlexNet is a CNN model of five convolutional layers, three max-pooling layers, two fully connected layers, and one softmax layer, respectively. (Guo, Pang, Du, Jiang, & Hu, 2020). It is shown in the Figure 3. In IDS-AlexNet, size of the input is reshaped into 8x8x1. In first convolution layer, 8X8X32 window shape is used. In the second layer, window shape is reduced to 7x7x64 and then by 6×6×64 in fifth layer. The network is added with maximum pooling layers with a window shape of 7×7×32,6×6×64 and 5×5×64 after the first, second, and fifth convolutional layers respectively. There are two fully connected layers of 512 and 256 outputs. With the exception of the output layer, IDS-AlexNet employs Relu activation in each of these layers to simplify model training. Dropout layers were also employed by IDS-AlexNet to keep their model from overfitting. The final fully linked layer or output layer uses the softmax activation algorithm.

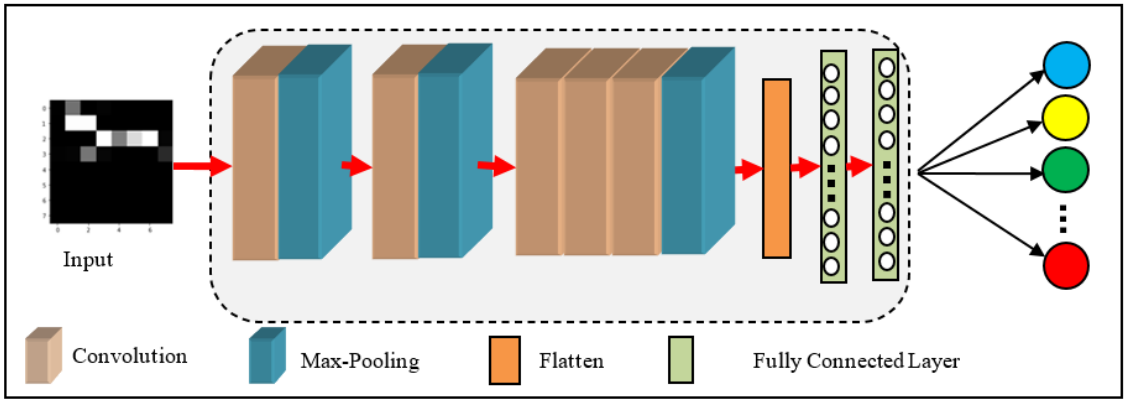

Figure 3. Architecture of IDS-AlexNet Model.

Combined Ensemble Intrusion Detection Model with Random Forest algorithm

The goal of the ensemble technique is to use group learning to benefit from various classifiers. Each classifier has various advantages and disadvantages. Some classifiers may work well for identifying a

particular kind of attack but poorly for identifying other types of attack. In this paper, we have proposed ensemble model based on traditional CNN and IDS-AlexNet which is used with random forest for feature selection. The selected features are given as input to the ensemble model that combines classifiers by training multiple classifiers and then form a stronger classifier to obtain reliable and more accurate attack predictions. In our proposed model, first we train two classifiers of traditional CNN and IDS-AlexNet CNN individually. The same data was used to train the two classifiers, CNN and IDS-AlexNet. The two classifiers mentioned above are then combined to generate the ensemble CNN-IDS model. Given that each classifier might have a distinct area in the feature space, they might all perform at their best. As a result, an ensemble model can integrate different networks as opposed to picking the best network and ignoring the rest. The suggested model can identify the global solution that is utilised to lower the false alarm rate and increase detection accuracy since it uses two classifiers.

Proposed Ensemble CNN-IDS Framework with Random Forest Algorithm

The overall structure of the ensemble CNN-IDS with RF model is shown in the Figure 4 that contain four processes.

Step 1: Label encoding and normalisation are used for data preprocessing. Numeric columns' values are altered using normalisation, which keeps track of different value ranges. For the purpose of transforming categorical data into numerical data, label encoding is used.

Step 2: Feature Selection Process is used to select important features using Random Forest algorithm.

Step 3: There are two classifiers built: CNN and IDS-AlexNet models. They each receive independent training using the input subset.

Step 4: The classic CNN and IDS-AlexNet models mentioned above are combined to create the ensemble model.

The ensemble model is used to obtain the final result. The performance of the model is assessed by the classification results.

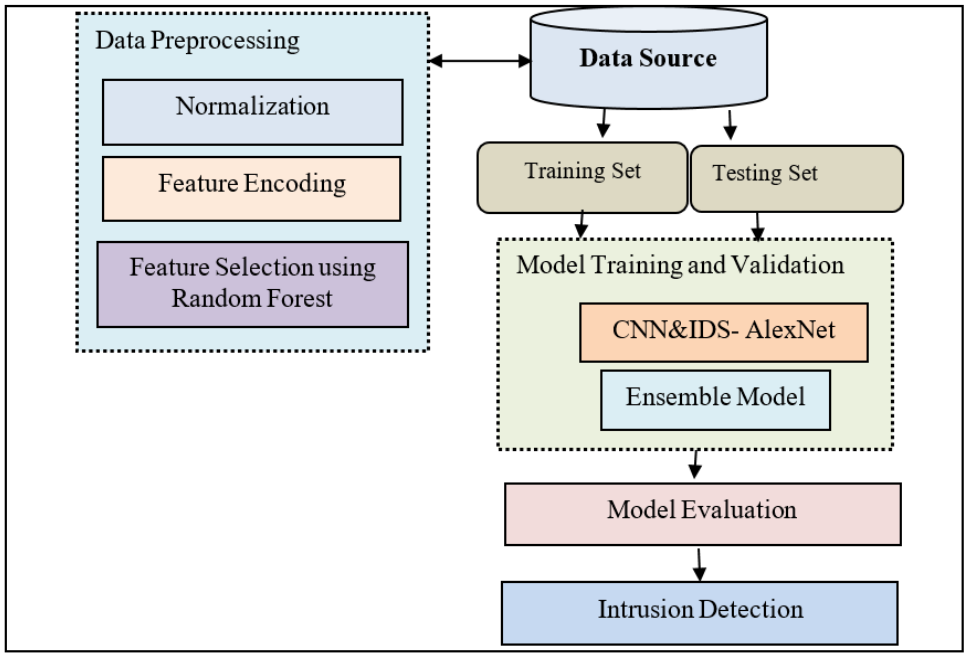

Figure 4. Block Diagram of Ensemble CNN-IDS with Random Forest Model.

Dataset

The Australian Center for Cyber Security (ACCS) produced the benchmark IDS dataset UNSW-NB15 (University of New South Wales Network Based) in 2015. The dataset is captured using TCP dump tool. The attacks in this dataset (Moustafa & Slay, 2015) include Backdoors, Generic, Reconnaissance, Analysis, DoS, Exploits, Shell code Fuzzers and Worms, as well as diverse sorts of actual normal traffic. Since the original entire dataset was partitioned, the training and testing datasets are now accessible for IDS evaluation. Flow,

Basic, Content, Time, and Additional Generated Features are the five categories of features. As shown in Table 2, the preconfigured UNSW-NB15 training set and testing set contain 175,341 and 82,332 records, respectively. Both have 45 features, including 2 classes and 43 features.

Table 2. UNSW NB15 dataset description.

Attack Class	No. of Instances Training Set	No. of Instances Testing Set
Normal	56,000	37,000
Fuzzers	18,184	6,062
Analysis	2,000	677
Backdoors	1,746	583
DoS	12,264	4,089
Exploits	33,393	11,132
Generic	40,000	18,871
Reconnaissance	10,491	3,496
Shell Code	1,133	378
Worms	130	44
Total	1,75,341	82,332

Data preprocessing

Normalization of data

There are two categories of numerical and nominal features in the UNSW-NB15 dataset. It contains 4 nominal qualities and 41 numerical ones. In order to prevent classifier bias because they have wide value ranges, 15 of the 41 numeric features are normalised using min-max normalisation to convert the values with a range of 0 to 1. This is done using Equation (2) below.

$$x_n = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2)$$

where x and xn represent the old value and new value of numeric features.

Feature encoding

Three nominal features proto, service, and state are present in our dataset. For the purpose of turning the aforementioned three nominal (categorical) attributes into numerical values, they are label encoded.

Feature Selection

The need for feature selection arises from the possibility that some of the features in our dataset are more significant than others or maybe wholly unimportant. By eliminating extraneous characteristics, the feature selection procedure generates candidate feature subsets for our research. Three features—id, label, and attack cat attributes—have been eliminated from the dataset. Random forest algorithm is used in our framework for selecting important features. The features f10 sttl, f41 ct_srv_dst, f7 sbytes, f27 smean, f2 proto, f32 ct_state_ttl, f14 sloss, f25 synack, f35 ct_dst_src_ltm, f28 dmean, f31 ct_srv_src, f3 service, f13 ct_dst_sport_ltm, f8 dbytes, f15 dloss, f4 state, f24 tcprtt, f34 ct_src_dport_ltm and f9 rate have been selected as important features of Random Forest algorithm.

Now the dataset consists of 19 features and padded with 0 for resizing to 64 dimensions and it is transformed to two-dimensional matrix of size 8 X 8. Then each record is converted as 8x8 gray scale image as image dataset for giving input to convolutional neural network where 8 represents height and width of the image.

Algorithm: Ensemble CNN-IDS with RF

Step 1. Load Training Data, UNSWNB-15.

Step 2. Preprocess the data. Numerical data is normalized and Categorical data is converted into numeric form using Label encoding.

Step 3. Select important features using Random Forest algorithm.

Step 4. Map preprocessed network traffic features into two-dimensional feature vectors.

Step 5. Input the transformed two-dimensional feature matrix to CNN, IDS-AlexNet and Ensemble CNN-IDS Model

Step 6. Create traditional CNN and IDS-AlexNet Models

Step 7. Train the Models
Step 8. Create Ensemble CNN-IDS with RF model using CNN and IDS-AlexNet Model.
Step 9. Evaluate the Model
Step 10. Detect the Intrusion.

Results and discussion

The proposed Ensemble CNN-IDS with RF model is implemented using Python and Keras (the deep learning library of Python) on a computer equipped with an Intel Core i7 CPU, 16 GB of RAM and Windows 10.

Evaluation metrics

The performance of the model is evaluated by using the metrics of accuracy, precision, misclassification Rate and False Alarm Rate, F-measure and recall. True Positive (TPw) is the no. of correctly predicted anomalous instances. True Negative (TNw) represents the no. of correctly predicted normal records. False Positive (FPw) indicates the no. of incorrectly predicted as attacks, but they are actually normal records. False Negative (FNw) is the no. of incorrectly predicted as normal that they are actually anomalous records. From the confusion matrix, we can define performance metrics mathematically as follows.

Accuracy indicates the ratio of the number of records that have been correctly classified to the total no. of instances as given below.

$$\text{Acc}(A) = \frac{\text{TPw+TNw}}{\text{TPw+TNw+FPw+FNw}} \quad (3)$$

Recall is the ratio of the anomalous records have been correctly predicted divided by the total no. of anomalous records as given below.

$$\text{Recall}(R) = \frac{\text{TPw}}{\text{TPw+FNw}} \quad (4)$$

Precision indicates the number of attack class predictions that actually anomalous records as given below.

$$\text{Precision}(P) = \frac{\text{TPw}}{\text{TPw+FPw}} \quad (5)$$

F-measure: It is the harmonic mean of A and R, which provides a measurement of derived effectiveness as given below.

$$F - \text{measure}(F - \text{Score}) = 2^{*}\frac{\text{R*P}}{R+P} \quad (6)$$

False Alarm Rate: This is the misprediction of normal records as anomalous records as given below.

$$\text{False Alarm Rate(FAR)} = \frac{\text{FPw}}{\text{FPw}+TNw} \quad (7)$$

Misclassification Rate provides the number of incorrectly classified records as given below.

$$\text{Misclassification Rate(MR)} = \frac{\text{FPw+FNw}}{\text{TPw+TNw+FPw+FNw}} \quad (8)$$

For traditional CNN we have used convolutional layers, max pooling layers and fully connected layers of three, three and two respectively. Its parameter summary is shown on Table 3. Likewise, the parameter summary for IDS-AlexNet model is shown in Table 4. We labeled Generic attack, Exploits attack, Fuzzers attack, DoS attack, Reconnaissance attack, Analysis attack, Backdoors attack, Shellcode attack and Worms attack from 1 to 9 while the normal connections were labeled as 0. The confusion matrix for UNSW-NB15 testing dataset of the ensemble CNN-FCID with RF is shown in the Table 5. The performance of the ensemble CNN-IDS with RF model is shown in Table 6 with accuracy, precision, recall, F-Score, false alarm rate, and misclassification rate for each class. In our multiclass classification, 72729 records were classified correctly. It yields accuracy of approximately equal to 97.5%. From the assessment, it is clear that the highest F-score values have been obtained for normal, and generic classes and the lowest values for Worms, Shellcode and Back classes which have the lowest number of the records. The UNSW-NB15 dataset has the imbalanced records for all the classes. It is considered as the potential factors for the difference among various classes performance.

Table 3. Parameters of CNN Model.

Layer Name	Details	Output Shape	Parameters
Input	Grey Images	8,8,1	-
Conv_2D	Conv (32)	8,8, 32	320
Activation	ReLU	8,8, 32	-
MaxPooling_2D	Pool Size (2,2)	7,7,32	0
Conv_2D	Conv (64)	7,7,64	18496
Activation	ReLU	6, 6, 64	-
MaxPooling_2D	Pool Size (2,2)	6,6,64	0
Conv_2D	Conv (64)	6,6,64	36928
Activation	ReLU	5,5,64	-
MaxPooling_2D	Pool Size (2,2)	5,5,64	0
Flatten	Convert 2D to 1D	1600	-
Fully Connected1	Dense (512)	512	819712
Fully Connected2	Dense (256)	256	131328
Dense	Input class=10	10	2570
Total Parameters :		9,42,410	
Trainable Parameters :		9,42,410	
Non-Trainable Parameters:		0	

Table 4. Parameters of IDS-AlexNet Model.

Layer Name	Details	Output Shape	Parameters
Input	Grey Images	8,8,1	-
Conv_2D	Conv (32)	8,8, 32	320
Activation	ReLU	8,8, 32	-
MaxPooling_2D	Pool Size (2,2)	7,7,32	0
Conv_2D	Conv (64)	7,7,64	18496
Activation	ReLU	6, 6, 64	-
MaxPooling_2D	Pool Size (2,2)	6,6,64	0
Conv_2D	Conv (64)	6,6,64	36928
Conv_2D	Conv (64)	6,6,64	36928
Conv_2D	Conv (64)	6,6,64	36928
Activation	ReLU	5,5,64	-
MaxPooling_2D	Pool Size (2,2)	5,5,64	0
Flatten	Convert 2D to 1D	1600	-
Fully Connected1	Dense (512)	512	819712
Fully Connected2	Dense (128)	128	65664
Dense	Input class=10	10	1290
Total Parameters :		1,083,210	
Trainable Parameters :		1,083,210	
Non-TrainableParameters:		0	

Table 5. Confusion Matrix for UNSW-NB15 dataset.

Predicted / Actual	Normal	Generic	Exploits	Fuzzers	DoS	Reconnai-ssance	Analysis	Backdoors	Shellcode	Worms
Normal	34100	305	462	131	296	739	512	132	219	104
Generic	84	18091	419	150	51	41	2	9	21	3
Exploits	134	123	9009	103	150	745	163	394	153	158
Fuzzers	310	134	189	4051	121	500	191	187	267	112
DoS	107	101	102	2	3500	55	20	85	46	71
Reconnaissance	254	13	728	53	105	2871	133	354	68	1
Analysis	158	11	21	10	10	41	502	37	5	4
Backdoors	24	3	46	1	0	166	0	300	37	6
Shellcode	63	0	12	2	2	4	0	0	295	0
Worms	2	2	3	2	5	4	3	7	6	10

Table 6. Evaluation of Ensemble CNN-IDS for UNSW-NB15.

	Accuracy	Precision	Recall	F-Score	DR	FAR	MAR
Normal	0.9517	0.9678	0.9216	0.94412	0.9216	0.02441	0.0483
Generic	0.9824	0.9632	0.9587	0.9609	0.9587	0.0107	0.0176
Exploits	0.9508	0.8197	0.8093	0.8145	0.8092	0.0273	0.0491
Fuzzers	0.9705	0.8992	0.6683	0.7667	0.6683	0.0059	0.0295
DoS	0.9841	0.8255	0.8560	0.8404	0.8559	0.0093	0.0159
Reconnaissance	0.9521	0.5558	0.6269	0.5891	0.6269	0.0291	0.04793
Analysis	0.9858	0.3604	0.6283	0.4580	0.6283	0.0108	0.0142
Backdoors	0.9822	0.1993	0.5146	0.2873	0.5146	0.0145	0.0178
Shellcode	0.9892	0.2641	0.7804	0.3946	0.7804	0.0099	0.0108
Worms	0.9941	0.0213	0.2273	0.0389	0.2273	0.0055	0.0059

Performance comparison

Our proposed model's superior performance over other models is demonstrated by comparison with other recent works. The performance analysis of our model is done with the machine learning models of SVM and kNN (Kasongo & Sun, 2020). which provided 61.09% and 70.09% respectively. The light weight IDS based on ANN Multi-layer perceptron (Olamantanmi, Alowolodu, Mebawondu, & Adetunmbi, 2020) was proposed by Olamantanmi et.al. and provided the accuracy of 76.96 %. An integrated rule based model for IDS has been proposed by Kumar, Sinha, Das, Pandey, and Goswami, (2020) and has given accuracy of 84.83%. The accuracies of models are presented in Table 7.

Table 7. Accuracy Comparison.

Model	Accuracy (%)
SVM	61.09
kNN	70.09
ANN	76.96
Integrated IDS	84.83
Our Proposed Model	97.5

Conclusion

In this paper. A deep learning based ensemble model with Random Forest algorithm is proposed for Fog computing intrusion detection systems Traditional CNN and IDS-AlexNet models serve as the foundation for the proposed ensemble model. For training and testing our model for multiclass classification, we used UNSW-NB15 datasets. In phase I, The Random Forest algorithm is used to select important features form the dataset. We converted a one-dimensional dataset into images for the CNN model's convention, which enables CNN to perform precise prediction attacks. In phase II, we trained and evaluated the IDS-AlexNet and conventional CNN models. In Phase III we have combined these models by creating ensemble CNN-IDS model. We have evaluated our model using various metrics of Accuracy, Precision, Recall, F-Score, False Alarm Rate, Misclassification Rate. After evaluation, it is clear that our model improves accuracy and reduces false alarm rate. Our proposed model is compared with machine learning and deep learning models of other researchers and it is proved that it outperformed other models. Our future work will be towards working of oversampling of datasets for improved prediction since some of the attack classes in UNSW-NB15 dataset like Worms and Shellcode, have minimum number of records. The oversampling can improve the accuracy of prediction.

Reference

Adebowale, A., Idowu, S. A., & Amarachi, A. A. (2013). Comparative study of selected data mining algorithms used for intrusion detection. *International Journal of Soft Computing and Engineering (IJSCE), 3*(3), 237-241.

Ahmed, A., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications, 60*, 19-31. DOI: https://doi.org/10.1016/j.jnca.2015.11.016

Akbar, S., Rao, T. S., & Hussain, M. A. (2016). A hybrid scheme based on Big Data analytics using intrusion detection system. *Indian Journal of Science and Technology, 9*(33), 1-4. DOI: https://doi.org/10.17485/ijst/2016/v9i33/97037

Alom, M. Z., Bontupalli, V., & Taha, T. M. (2015). Intrusion detection using deep belief networks. In *National Aerospace and Electronics Conference [NAECON]* (p. 1-22). Dayton, OH: IEEE.

Al Mehedi Hasan, M., Nasser, M., & Pal, B. (2013). On the KDD'99 dataset: Support vector machine based intrusion detection system (ids) with different kernels. *International Journal of Electronics Communication and Computer Engineering*, *4*(4), 1164–1170.

Aydın, M. A., Zaim, A. H., & Ceylan, K. G. (2009). A hybrid intrusion detection system design for computer network security. *Computers & Electrical Engineering, 35*(3), 517-526. DOI: https://doi.org/10.1016/j.compeleceng.2008.12.005

Beghdad, R. (2007). Training all the KDD data set to classify and detect attacks. *Neural Network World, 17*(2), 81-91.

Bu, S.-J., & Cho, S.-B. (2020). A convolutional neural-based learning classifier system for detecting database intrusion via insider attack. *Information Sciences, 512,* 123-136. DOI: https://doi.org/10.1016/j.ins.2019.09.055

Butun, I., Morgera, S. D., & Sankar, R. (2014). A survey of intrusion detection systems in wireless sensor networks. In *6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)* (p. 266-282). Istanbul, TR: IEEE.

Deepan, P., & Sudha, L. R. (2019). Fusion of deep learning models for improving classification accuracy of remote sensing images. *Journal of Mechanics of Continua and Mathematical Sciences, 14*(5), 189-201. DOI: http://doi.org/10.26782/jmcms.2019.10.00015

Devaraju, S., & Ramakrishnan, S. (2014). Performance comparison for intrusion detection system using neural network with KDD dataset. *ICTACT Journal on Soft Computing, 4*(3), 743-752. DOI: https://doi.org/10.21917/ijsc.2014.0106

Guo, Y., Pang, Z., Du, J., Jiang, F., & Hu, Q. (2020). An improved alexnet for power edge transmission line anomaly detection. *IEEE Access, 8*, 97830-97838. DOI: https://doi.org/10.1109/access.2020.2995910

Han, S.-J., & Cho, S.-B. (2003). Detecting intrusion with rule-based integration of multiple models. *Computers & Security, 22*(7), 613-623. DOI: https://doi.org/10.1016/S0167-4048(03)00711-9

Kasongo, S. M., & Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. Journal of Big Data*, 7*, 105. DOI: https://doi.org/10.1186/s40537-020-00379-6

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity, 2*, 20. DOI: https://doi.org/10.1186/s42400-019-0038-7

Koc, L., Mazzuchi, T. A., & Sarkani, S. (2012). A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. *Expert Systems with Applications, 39*(18), 13492-13500. DOI: https://doi.org/10.1016/j.eswa.2012.07.009

Kumar, V., Sinha, D., Das, A. K., Pandey, S. C., & Goswami, R. T. (2020). An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset. *Cluster Computing, 23*, 1397-1418. DOI: https://doi.org/10.1007/s10586-019-03008-x

Li, Y., Ma, R., & Jiao, R. (2015). A hybrid malicious code detection method based on deep learning. *International Journal of Security and Its Applications, 9*(5), 205-216. DOI: http://dx.doi.org/10.14257/ijsia.2015.9.5.21

Li, Z., Qin, Z., Huang, K., Yang, X., & Ye, S. (2017). Intrusion detection using convolutional neural networks for representation learning. In D. Liu, S. Xie, Y. Li, D. Zhao, & E. S. El-Alfy (Eds.)*, Neural information processing* (p. 858-866). Cham, DE: Springer.

Moustafa, N., & Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*. Canberra, AU: IEEE.

Olamantanmi, J. M., Alowolodu, O. D., Mebawondu, J. O., & Adetunmbi, A. O. (2020). Network intrusion detection system using supervised learning paradigm, *Scientific African*, *9*, e00497. DOI: https://doi.org/10.1016/j.sciaf.2020.e00497

Pfahringer, B. (2000). Winning the KDD99 classification cup: bagged boosting. *Association for Computing Machinery, 1*(2), 65-66. DOI: https://doi.org/10.1145/846183.846200

Puliafito, C., Mingozzi, E., Longo, F., Puliafito, A., & Rana, O. (2019). Fog computing for the internet of things. *ACM Transactions on Internet Technology, 19*(2), 1-41. DOI: https://doi.org/10.1145/3301443

Ray, P. P. (2018). A survey on Internet of Things architectures. *Journal of King Saud University - Computer and Information Sciences, 30*(3), 291-319. DOI: https://doi.org/10.1016/j.jksuci.2016.10.003

Syarif, I., Prugel-Bennett, A., & Wills, G. (2012). Unsupervised clustering approach for network anomaly detection. In *Fourth International Conference on Networked Digital Technologies [NDT 2012]* (p. 135-145). Heidelberg, DE: Springer.

Tajbakhsh, A., Rahmati, M., & Mirzaei, A. (2009). Intrusion detection using fuzzy association rules. *Applied Soft Computing, 9*(2), 462-469. DOI: https://doi.org/10.1016/j.asoc.2008.06.001

Tchakoucht, T. A., & Ezziyyani, M. (2018). Multilayered echo-state machine: a novel architecture for efficient intrusion detection. *IEEE Access, 6*, 72458-72468. DOI: https://doi.org/10.1109/access.2018.2867345

Tu, S., Waqas, M., Rehman, S. U., Aamir, M., Rehman, O. U., Jianbiao, Z., & Chang, C.-C. (2018). Security in fog computing: a novel technique to tackle an impersonation attack. *IEEE Access, 6*, 74993-75001. DOI: https://doi.org/10.1109/access.2018.2884672

Ustebay, S., Turgut, Z., & Aydin, M. A. (2018). Intrusion detection system with recursive feature elimination by using random Forest and deep learning classifier. In *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism [IBIGDELFT]* (p. 71-76). Ankara, TR: IEEE.

Wahab, O. A., Bentahar, J., Otrok, H., & Mourad, A. (2019). Resource-aware detection and defense system against multi-type attacks in the cloud: repeated bayesian stackelberg game. *IEEE Transactions on Dependable and Secure Computing, 18*(2), 605-622. DOI: https://doi.org/10.1109/tdsc.2019.2907946

Ye, N., Emran, S. M., Chen, Q., & Vilbert, S. (2002). Multivariate statistical analysis of audit trails for host-based intrusion detection. *IEEE Transactions on Computers, 51*(7), 810-820. DOI: https://doi.org/10.1109/TC.2002.1017701

Yi, S., Qin, Z., & Li, Q. (2015). Security and privacy issues of fog computing: a survey. In K. Xu, & H. Zhu, (Eds.)*, Wireless algorithms, systems, and applications* (p. 685-695). Cham, DE: Springer.

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access, 5*, 21954-21961. DOI: https://doi.org/10.1109/ACCESS.2017.2762418

Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., ... Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms: a complete survey. *Journal of Systems Architecture, 98*, 289-330. DOI: https://doi.org/10.1016/j.sysarc.2019.02.009

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion, 42*, 146-157. DOI: https://doi.org/10.1016/j.inffus.2017.10.006