How to Cite:
Mhedi, E. A., Kashef, N. M., Elmorsy, S. A., & Aburawash, U. A. (2022). Detecting abnormal behaviors by using intelligent system. *International Journal of Health Sciences*, *6*(S6), 11044–11056. https://doi.org/10.53730/ijhs.v6nS6.13015

Detecting abnormal behaviors by using intelligent system

Ehab Anan Mhedi
Department of Mathematics and Computer Science/Faculty of Science, Alexandria University, Alexandria, Egypt
*Corresponding author email: Ehab.anan_PG@alexu.edu.eg

Nermeen Mahmoud Kashef
Department of Mathematics and Computer Science/Faculty of Science, Alexandria University, Alexandria, Egypt
Email: Nermeen.kashief@alexu.edu.eg

Shaimaa Ali Elmorsy
Department of Mathematics and Computer Science/Faculty of Science, Alexandria University, Alexandria, Egypt
Email: shaimaa.aly.comp@alexu.edu.eg

Usama Abdullah Aburawash
Department of Mathematics and Computer Science/Faculty of Science, Alexandria University, Alexandria, Egypt
Email: aburawash@alexu.edu.eg

Abstract---With the rising use of Internet technologies around the world, the number of network intruders and attackers has skyrocketed. For this reason, introducing systems for detecting attacks within the network security measures stops hackers from gaining access to data. In order to detect various forms of assaults, the development of intrusion detection systems is quite crucial. The election of features and elimination of the unrelated information can improve the classifier's accuracy execution because the network traffic dataset contains many useful and unhelpful features. Along these lines, in this work, three hybrid strategies for selecting the feature were implemented, which include the constant feature by standard deviation, the Quasi constant by variance threshold, and the information gain. These approaches were used to order and rank features, after which the best higher ranking was selected for classification and intrusion detection. These features were tested on three classifiers long short-term memory, convolutional neural network, and CNN-LSTM. From the acquired results, a high level of attacks was detected, and classification accuracy was achieved by

cross-breeding the selection of the different best features. Furthermore, the convolutional neural network classifier achieved the highest accuracy rate, which exceeded the value of 99.5%. The proposed model was applied to the Knowledge Discovery and Data KDD CUP99 dataset.

Keywords---convolutional neural network (CNN), LSTM, intrusion detection system, KDD99, machine learning.

Introduction

Due to the increasing occurrence of network intrusion events and leaking of confidential information by hackers because of systems' bugs and gaps, the enhancement of the systems through the development of intrusion detection systems has become an inevitable contemporary concern. Cyber-attacks nowadays have become such an important concern that sophisticated actions even from ruling authorities in different countries have been made. Interestingly, the latter have become so excited to celebrate the creation of cyber-armies, which are expert personnel who are trained to protect confidential information. Intrusion detection systems IDS is regarded as a sufficiently influential notion that is applied under different titles to preserve networks from any probable attack via appraising a load of a system's traffic to make sure of their availability, confidentiality, and integrity. Even though a wide range of research and tests are being implemented in this field, IDSs continue to face blockages and barricades for detecting destructive network intrusions. Recently, IDS systems were broadly designed and developed based on DL and ML to play a more influential and efficient role in intrusion detection[1]. Currently, there are two types of IDS systems, namely the network-based intrusion detection system (NIDS) and host-based intrusion detection system (HIDS). [2].

Related work

In 2020, Prasanna Kottapalle [3], used the variant size 10% of the KDD99. In the proposed work, the algorithms of SVM, DBN, CNN, and CNN-LSTM were applied. From the calculated results, it was proven that this model could raise the accuracy of intrusion detection and foster the efficiency of the attack detection method, which was as follows: CNN attained an accuracy of99.23%, and CNN-LSTM got 99.78%. In 2020, Sarika Choudhary and Nishtha Kesswani[4], proposed a deep neural network (DNN) to detect attacks by using the KDD99 dataset, which obtained the highest accuracy of 96.3%, which is comparable to the experiment conducted on the NSL-KDD dataset and with the same DNN model, where the percentage of the accuracy was91.5%.

In 2021, deep learning was also used by Laghrissi, Fatima Ezzahra and others [5],for the detection of attacks by using long-term memory (LSTM).For decreasing dimensionality and feature selection, the principal component analysis (PCA algorithm and mutual information (MI) methods were used. A binary classification with multiple classifications (3-categories) was also applied. The authors combined the results in both classification groups and tested them on the KDD99

dataset. For the LSTM-PCA algorithm in binary, an accuracy of 99.44% was derived, while in multiclass the value of99.39% was attained. In striking contrast, for the LSTM-MI algorithm in binary the acquired accuracy was 96.99%, while multiclass got 96.57% of testing accuracy.

In 2021 [6],Ruizhe Yao et al. suggested a model that combines the two algorithms of CNN and LSTM in the form of a cross-layer. More specifically, the CNN component identifies provincial features to produce universal features, while the LSTM component uses the function of memory to generate periodic features. Two types of features were integrated to generate complete features with characteristics from several domains that can more reliably detect intrusion input in advanced metering infrastructure (AMI). In one of the experiments, the CNN-LSTM algorithm scored an accuracy of 99.95% by using the KDD Cup 99 dataset, whereas for the NSL-KDD dataset an accuracy of 99.79% was attained.

Data reduction techniques used in IDS
Dimension Reduction

In general, dimension reduction is regarded as a significant subfield of Artificial Intelligence. In light of the importance of dealing with big data correctly there is a possibility of information damage. The singled-out attributes are sometimes high dimensional to the data and difficult to work with. As a result, such data are difficult to be categorized. Because of the high dimensionality of the data, some redundant features result in data loss. Since the normal database management cannot handle this type of high dimensional data (HDD)[7], the seminal purpose of the feature reduction issue is to decrease the size of the primary dataset preserving the function accuracy. Feature construction and feature selection are all involved in the feature reduction process. The construction process or attribute extraction builds a new set of features from the original datasets, whereas the feature selection process selects the relevant features from the origin dataset[8,9].This work deals also with the attribute selection process, accordingly, whereas duplicate data were removed, and important features were selected in the KDD99 database.

Feature Selection

The collected data from the network pockets were used to figure out intrusions. For that reason, the manual classification of a broad range of network data that was selected by the determined system could not be as time-economic as it primarily seems to be. During this process, data analysis is a totally different level of activity and is considered a difficult task since data include a huge amount of attributes and behavioral patterns. Additionally, for real-time intrusion detection, the system should be secured against any intruding factor. This could be implemented by the identification of the important attributes within the available dataset. However, the reduction of the number of attributes would eye-catchingly optimize the ratio of intrusion detection. Currently, various feature selection methods are available. For instance, the application of algorithms for filtering the data does not contribute to the attack identification, as well as arranging the data into the same-feature chain of clusters for the classification of the hidden patterns to eliminate redundant attributes. Feature selection is defined as a

process of identifying a subset of attributes from the available dataset. The selection of features of certain values optimizes the visualization and understandability of the intrusion detection learning algorithms. The feature selection process, in general, eliminates undesirable noises, irrelevant attributes, and redundancy from a dataset [10].

Proposed methodology

The main process steps used in this work are summarized in Fig.1.

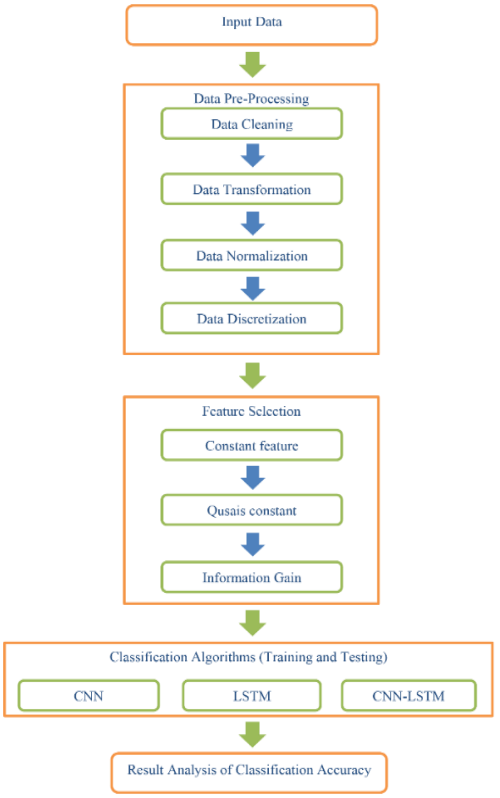

Fig.1. Depiction of the proposed system block diagram.

Data preprocessing

Data pre-processing helps to improve the ability and effectiveness of the algorithms to properly classify data. In the data preprocessing step the data will pass in the following stages:

- Data cleaning: During this stage, any records with inconsistent or missing values are removed. The repeated records of the respective dataset are also omitted. In this phase, 70% of the dataset is reduced by the method of dropping the features in python.
- Data transformation: The next step in the preprocessing process is to convert text-based attributes to numeric values so that classification

algorithms can use them as input. The conversion was carried out by using the sklearn. preprocessing library in Python.

- Data normalization: As a step-in data preprocessing, feature normalization is quite critical for the detection performance when datasets are too huge. The feature ranges were normalized by scaling their values to fit within a small range of values (from -1 to 1).Normalization was performed by using the Min-Max approach. The technique of applying a linear change to the original data values is known as the Min-Max Normalization. By assuming that min and max are the upper and lower values of feature x. (min_x,max_x)into a new interval to [new_{max_A} , new_{min_A}], every value $v$from the original interval will be mapped into valuenew_v byapplying the following equation:

$$v` = \frac{v - min_A}{max_A - min_A}\left(new_{max_A} - new_{min_A}\right) + new_{min_A} \quad (1)$$

- Data Discretization: When a wide range of dataset ranges exists, the classification of the data so that it can be presented in the form of categorical attributes becomes difficult since a continuous trait must be converted into a categorical trait. As a result, converting a continuous attribute to a categorical attribute necessitates the determination of the number of the required classes and the way of values assigned to these classes .KDD datasets were divided into two phases for the classification technique, namely training and testing datasets. The training phase was used to construct a classification model, which is the process of learning something from examples to predict the type of class. During the testing phase, the dataset was also used to evaluate and test the algorithm's learning [11,12].

Feature selection

The effective stage of feature selection was applied to improve the performance of classification algorithms. Three different types of feature selection methods were employed (Constant feature, Qusais constant, and Information Gain) and then top-ranking features from the three methods were combined. Moreover, three different types of classification models (CNN, LSTM, and CNN - LSTM) were used to demonstrate that integrated feature selection has a beneficial impaction the accuracy of categorization attacks category. There are two sections to the dataset: training and testing. The classifier was trained by using the dataset for training. The dataset for testing was utilized to evaluate the performance of the classifiers to acquire the recognition rates.

Feature selection techniques

A huge number of features were obtained through a lengthy feature engineering process, while some features may not be employed. The most important thing is to build a model with only the most important features and to exclude those that don't have any predictive potential. Under this direction, a hybrid strategy was proposed in this work to pick the significant and useful aspects of the detection process, as follows:

- Constant feature by the standard deviation:

The fixed features that are equal to zero were removed by using the drop const features by Python. The dropped features will have no impaction the system performance because they are identical.

- Quasi constant by variance threshold:
  Semi-constant characteristics, i.e. values with more than 99% comparable output values were removed we used sklearn. feature_ selection library and we imported Variance Threshold for this method by Python.
- Information Gain (IG):
  This method was used to rate the importance of a feature from the most to the least significant. The IG metric measures information provided by the feature offers about the target class. IG can detect the features with the highest information based on the target class. Features with a high IG are highly relevant to the target class and thus, are often used to achieve the best classification results. IG, on the other hand, is unable to reduce unnecessary features. As a result, unnecessary features should be further removed. As can be observed from the following equations, IG is generated from Entropy. In the (IG) method, the entropy meter was used to measure impurities in collecting data for training. A feature with greater entropy contains more information. The A ratio ofSrefers to the relationship to class i. For multi-classes, the maximum IG value was1[13].

$$\text{Entropy} = \sum_{K=0}^{n} - p_i log^2 p_i \qquad (2)$$

The (IG) of an attribute A is defined as follows:

$$Gain(S.A) = Entropy(\text{S}) - \sum_{v \in value.(A)} \frac{|sv|}{|s|} Entropy(S_v) \qquad (3)$$

$A:$ represents the collection of all attributes A's possible values.
$SV:$ stands for the subset of S that has a value for A.

Deep learning models for IDS

Some deep learning (DL) approaches were also used to propose deep learning-based IDS solutions. These techniques are more efficient than machine learning because of their deep structure and ability to learn the important features from the dataset on its own and generate an output. The following describes the classifiers that were applied to the (21) features extracted from the methods of selecting features out of the (42) features.

Long Short-Term Memory (LSTM)

LSTM is a developed version of RNN, which was proposed by Schmidhuber and Hochreiter in 1997 as a solution to RNN difficulties (training vanishing-exploding gradients and lengthy clusters of data).LSTM is used as the basis for a broad framework of surprising deep learning implementations, performing over 4 billion neural operations per day[14].More specifically, memory blocks are used to replace the hidden layer nodes of the LSTM network to save information for long periods. There is a memory cell with three different gates in the memory block. A gate is also a component that is capable of handling and modification of the data

that passes through it, whereas it can be described as follows, knowing that initialize the weight matrices of the input connections W_k , recurrent connections R_k, and bias connections b_k. Where the subscript k can either be the input gate i, output gate o, or forget gate f of the memory cell c.[15]:

- **Forget gatef_t:** From this gate is determined whether the memory cell result is useful in calculating the running memory cell. The (f_t)function takes the current state value (input x_t) and the previously hidden state value (h_{t-1}) and uses the sigmoid function (σ) to determine whether to keep or discard the result.

$$f_t = \sigma\left(W_f x_t + R_f h_{t-1} + b_f\right) \quad (4)$$

- **Input gate i_t**: This gate employs x_t and h_{t-1} and then employs the sigmoid function (σ) to choose whether or not to employ them in the subsequent steps.

$$i_t = \sigma(W_i x_t + R_i h_{t-1} + b_i) \quad (5)$$

- **Output gate o_t**:This gate controls whether or not state values are passed to the next hidden layer.

$$O_t = \sigma(W_O x_t + R_O h_{t-1} + b_O) \quad (4)$$

Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is one of the important algorithms in deep learning and on a large scale is used to analyze data in depth. CNN enables a high-accuracy classification of large data after the analysis process [16].CNN is mostly used for image classification and speech recognition. The feature categorization procedure in this work was performed by using CNN. For this purpose, the KDD 99 Cup dataset was employed, which is a common benchmark dataset that is used as an input dataset for the CNN training process. A specified amount of test data was also used for the performing classification procedure after the training process was completed. The input datasets for the CNN are the most contributing features selected by the suggested feature selection algorithm. Different dense layers, as well as various filters and the rectified linear unit (ReLU) activation function, were used here. The CNN model was trained by using the training data and then tested with the test dataset, with the outcome features of one layer being fed into the input of other layers.

The training dataset was utilized to lower the classification error rate, as well as to identify the attacks. In addition, convolutional layers use various filters to handle input throughout the mapping process, as well as to filter features based on the mapping results. Furthermore, the ReLU function was utilized to keep the volume size of each layer constant. Furthermore, the output of the convolutional layer was transmitted to the fully connected layers for additional processing with parameters, such as hidden units, bias value, and ReLU activation function. The completely connected layer's features score was used to identify attack types such as Probe, DoS, R2L, and U2R. Eq gives the convolutional function [16]:

$$H_j = f(h_{j-1} \otimes w_j + b_j) \quad (7)$$

where ⊗ denotes a function of convolution,$f(y)$ stands for the ReLU activation function, h_j is a set of features, $w_j$representsthe bias value of a convolutional kernel layer, and b_j is the bias value of a layer. After using some convolutional layers and feature maps with filters, h_j should be converted to a vector. During the training and testing phases, the ReLU function was utilized to reach decisions on each input record. The conclusion of this function, which was included in CNN's soft-max layer, affects classification performance [16].

CNN-LSTM Model

A hybrid use of a convolutional neural network (CNN) and a long-term memory network (LSTM) for intrusion detection by extracting features for network traffic data was proposed in this work. With the presence of some unbalanced samples in most types of attacks within the training, the goal of the hybridization method is to reduce the impact of those samples. The LSTM layers were chosen due to their capacity to elicit serial style information and short and long-term subordination. The CNN layers, on the other hand, were used because of the potential to extract valuable characteristics based on data from a time series. Furthermore, CNN's layer said in the noise removal from the insertion data. Consequently, a hybrid model that uses both CNN and LSTM is foreseeable to be developed for improving IDS precision in predicting. [17,18].The Convolutional layers convert the input into a CNN and output to the following line. Different stages including convolution, maximum grouping, and flattening, were also frequently included in this network. The features derived from the input are represented by the convolution layer, while the grouping layer reduces the dimensionality of each feature map by retaining the most relevant information. What's more, the flatten layer converts the data into a one-dimensional matrix for entry into the next layer. The output of the layers of convolution was settled to produce a single long feature vector. Finally, the completely connected layers link each neuron in one layer to the next layer's neuron. [3].

KDD99 dataset

The KDD99 dataset is widely used to assess the performance of intrusion detection systems. More specifically, this dataset contains several redundant records, resulting in an unbalanced distribution of traffic types. It is made up of approximately 5 million records, including both abnormal and normal events [19]. The total number of KDD99 dataset data was(4,848,430), whereas in this work,10% of the dataset was taken, which amounted to (494,020) with 42 features. This data was subjected to a pre-processing process, as will be clarified later. After the preprocessing process, the number of the extracted features will be 23 types, one normal type and 22 types of attacks. The attacks can be divided into four main categories:

- Denial of Service Attack (DoS): Mail-bomb, land, ping of death or syn flood are instances of intrusions in which a memory resource or computing intensively occupied or overflowing are created by the offender to use legitimate requests, or denies lawful users have access to a system[20].
- Probing Attack (Prob):

This kind of intrusion, being known as a probing attack, is most probable to occur when a hacker scans and analyzes a system's ports via IP to figure out its weaknesses and gaps for exploiting the stored system data [21].

- User to Root (U2R) Attack:
  Such intrusions occur when a local user or machine exploits a system's vulnerable gap to access certain advantages fundamentally preserved for root users. For instance, an attack on the buffer overflow[22].
- Remote to Local (R2L) Attack:
  This kind of system intrusion takes place when an intruder is competent in transmitting data packets across a network but does not have access to the prey's computer system. A system's vulnerable gap is utilized to achieve advantages of the domestic access users to the device of a computer and manipulate data, for instance, by predicting a pass code[22]. Table (I) illustrates the data set and the categories applied.

Table 1
The KDD CUP99 data set is divided into five categories

NO.	Class Name	Class cont.	Instance Sum	Attacks Sum
1	NORMAL	normal	87831	1
2	DOS	Pod, teardrop, smurf, neptune, land, back	54572	6
3	R2L	Warezmaster, waresclient, spy, phf, multihop, imap, guess_passwd, ftp_write	999	8
4	U2R	Rootkit, perl, loadmodule, butter_overflow	52	4
5	PROBE	Satan, portsweep, nmap, ipsweep	2131	4
TOTAL	5 Class		145585	23

Discussions and results of the proposed models

The best feature-selection approaches were thoroughly investigated through their hybridization, and then the chosen attributes in algorithms were examined to obtain a more accurate functionality through classifying cyber-attack. Henceforth, an assumption of the incorporating attributes will be provided, which are more eligible and suitable for the classification process. Various classifiers were applied with the best ranked (21) attributes out of (42) features. CNN had the highest detection rate with an average rate of 99.6% of correctly classified instances. The detection rate of the LSTM classifier was99.1%, while the CNN_LSTM hybrid classifier got a detection rate of 99.5%. In general, the results show that through hybridization between feature selection methods and the use of deep learning classifiers, the classification detection rate is most probable. Therefore, in this part of the work, a comparison of the experimental results of the suggested approach between the classifiers was obtained by the used measures, while the acquitted outcomes were also compared with previous works (Table IV).The accuracy was calculated by the confusion matrix, which definesthe repetition of the classified intrusions correctly or incorrectly [23].Six performance metrics were utilized to assess the performance of the deep learning algorithms [24,25], whereas Table II presents the results and metrics.

Table 2
Results and metrics

Models	Accuracy	Precision	Recall	F1 Score	Cohens Kappa Score	Test time/m
CNN	0.9966	0.9964	0.9966	0.9965	0.9932	3.32
LSTM	0.9919	0.9917	0.9919	0.9918	0.9837	5.53
CNN_LSTM	0.9957	0.9956	0.9957	0.9956	0.9913	5.58

Table 3
Comparison with previous work

No. Ref.	year	Dataset	No. of sample for Training	No. of sample for Testing	Classification Algorithms	%DR *Accuracy*	*Precision*	*Recall*	*F1 Score*
[26]	2018	KDD99			DNN	92.9	9.98	0.954	
					Ada Boost	92.5	9.95	0.951	
					Decision Tree	92.8	9.99	0.953	
					K Nearest	92.9	9.98	0.954	
					Linear Regression	84.8	9.89	0.897	
					Naïve Bayes	92.9	9.88	0.955	
					Random Forest	92.7	9.99	0.953	
					SVM * - Linear	81.1	9.94	0.868	
					SVM * - rbf	81.1	9.92	0.868	
[11]	2019	10% of KDD99	70%= ~ 101906.7	30%= ~ 43674.3	K-NN	98.9	98.9		
					NAÏVE BAIS	93.3	93.3		
					MLP	96.5	96.5		
[27]	2019	KDD99	1,074,992	311,029	Logistic Regression	79.7			
					Decision tree	81.05			
					KNN	94.17			
					SVM	83.09			
					Random Forest	99.0			
					Adaboost	90.73			
					Multi-Layer Perceptron	80.5			
					Naïve Bayes	92.4			
[28]	2019	NSL-KDD (KDDTest+)	125973	22544	LSTM	89.23			
					CNN-LSTM	94.12			
					RNN	83.28			
		NSL-KDD (KDDTest-21)	125973	11850	LSTM	74.77			
					CNN-LSTM	79.37			
					RNN	68.55			
[3]	2020	KDD99	494020	311028	SVM	98.20			
					DBN	98.59			
					CNN	99.23			
					CNN-LSTM	99.78			
[4]	2020	KDD99	0.10953 at epoch 72.ROC		DNN	96.3			
		NSL-KDD	0.15506 at epoch 43.ROC		DNN	91.5			
[29]	2022	NSL-KDD	93.32%	88.26%	Decision Tree	98	96	0.99	0.97
					Logistic Regression	94	91	0.96	0.93

					Random Forest	98	96	1.00	1.00
					XGBoost	98	97	0.99	0.98
Proposed model	2022	10% of KDD99	70%= ~ 101906.7	30%= ~ 43674.3	CNN	99.66	9.96	0.996	0.99
					LSTM	99.19	9.91	0.991	0.99
					CNN_LSTM	99.57	9.95	0.995	0.99

Conclusion

In this work, the accuracy of the classification models according to the chosen attributes was systematically examined since. From the acquired outcomes, it was demonstrated that a higher detection rate could be obtained through crossbreeding between a selection of features and classification models according to the subset of features. A hybrid feature reduction method was also introduced to eliminate the number of features using stander division, quasi-constant and information gain from 42 to 21 features. These features were implemented by using different deep learning models to distinguish better results. The LSTM algorithm provides a lower accuracy rate than other algorithms but it required more time to build a detection model. On the other hand, CNN gets high classification detection accuracy and is the fastest test model for training algorithms. Finally, CNN_LSTM favorable outcomes were provided by the CNN_LSTM for classification as well.

Acknowledgments

Ehab A. M sincerely thanks to supervisor or Co-Authors for their insightful comments and suggestions that helped significantly improve this research work.

References

1. A.A. Salih, M.B. Abdulrazaq, Combining Best Features Selection Using Three Classifiers in Intrusion Detection System, 2019 Int. Conf. Adv. Sci. Eng. (2019) 94–99. https://doi.org/10.1109/ICOASE.2019.8723671.
2. A.A.S. Dr. Maiwan Bahjat Abdulrazaq, Combination of Multi Classification Algorithms for Intrusion Detection System.pdf, IJSER. 6 (2015). https://www.researchgate.net/profile/Maiwan-Abdulrazzaq-2/publication/339659233_Combination_of_multi_classification_algorithms_for_intrusion_detection_system/links/5ece01b3299bf1c67d203a97/Combination-of-multi-classification-algorithms-for-intrusion-detec.
3. Alhussein, K. Aurangzeb, S.I. Haider, Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting, IEEE Access. 8 (2020) 180544–180557. https://doi.org/10.1109/ACCESS.2020.3028281.
4. B. Riyaz, S. Ganapathy, A deep learning approach for effective intrusion detection in wireless networks using CNN, Soft Comput. 24 (2020) 17265–17278. https://doi.org/10.1007/s00500-020-05017-0.
5. C. Yin, Y. Zhu, J. Fei, X. He, A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks, IEEE Access. 5 (2017) 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418.
6. C.M. Hsu, H.Y. Hsieh, S.W. Prakosa, M.Z. Azhari, J.S. Leu, Using long-short-term memory based convolutional neural networks for network intrusion

detection, Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST. 264 (2019) 86–94. https://doi.org/10.1007/978-3-030-06158-6_9.
7. E. Sugawara, H. Nikaido, Properties of AdeABC and AdeIJK Efflux Systems of Acinetobacter baumannii Compared with Those of the AcrAB-TolC System of Escherichia coli, 2014. https://doi.org/10.1128/AAC.03728-14.
8. F.A. Khan, A. Gumaei, A. Derhab, A. Hussain, TSDL: A Two-Stage Deep Learning Model for Efficient Network Intrusion Detection, IEEE Access. 7 (2019) 30373–30385. https://doi.org/10.1109/ACCESS.2019.2899721.
9. F.E. Laghrissi, S. Douzi, K. Douzi, B. Hssina, Intrusion detection systems using long short-term memory (LSTM), J. Big Data. 8 (2021) 16. https://doi.org/10.1186/s40537-021-00448-4.
10. G. Chao, Y. Luo, W. Ding, Recent Advances in Supervised Dimension Reduction: A Survey, Mach. Learn. Knowl. Extr. 1 (2019) 341–358. https://doi.org/10.3390/make1010020.
11. J.M. Luca Massaron, Deep Learning For Dummies, 2019. https://g.co/kgs/9JztFF.
12. J.V. V. Silva, N.R. de Oliveira, D.S. V. Medeiros, M.A. Lopez, D.M.F. Mattos, A statistical analysis of intrinsic bias of network security datasets for training machine learning mechanisms, Ann. Telecommun. (2022) 1–21. https://doi.org/10.1007/s12243-021-00904-5.
13. M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set in Computational Intelligence for Security and Defense Applications, Comput. Intell. Secur. Def. Appl. (2009) 1–6.
14. O. Al-Jarrah, A. Arafat, Network intrusion detection system using attack behavior classification, 2014 5th Int. Conf. Inf. Commun. Syst. ICICS 2014. (2014). https://doi.org/10.1109/IACS.2014.6841978.
15. O.S.A. Aboosh, O.A.I. Aldabbagh, Android Adware Detection Model Based on Machine Learning Techniques, in: 2021 Int. Conf. Comput. Commun. Appl. Technol., IEEE, 2021: pp. 98–104. https://doi.org/10.1109/I3CAT53310.2021.9629400.
16. P. Agrawal, H.F. Abutarboush, T. Ganesh, A.W. Mohamed, Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019), IEEE Access. 9 (2021) 26766–26791. https://doi.org/10.1109/ACCESS.2021.3056407.
17. P. Kottapalle, A CNN-LSTM Model for Intrusion Detection System from High Dimensional Data, J. Inf. Comput. Sci. 10 (2020) 1362–1370.
18. P. Ray, S.S. Reddy, T. Banerjee, Various dimension reduction techniques for high dimensional data analysis: a review, Artif. Intell. Rev. 54 (2021) 3473–3515. https://doi.org/10.1007/s10462-020-09928-0.
19. P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao, J. Chen, DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system, Secur. Commun. Networks. 2020 (2020). https://doi.org/10.1155/2020/8890306.
20. R. Datti, S. Lakhina, Performance Comparison of Features Reduction Techniques for Intrusion Detection System, 8491 (2012) 332–335.
21. R. Delgado, X.A. Tibau, Why Cohen's Kappa should be avoided as performance measure in classification, PLoS One. 14 (2019). https://doi.org/10.1371/journal.pone.0222916.
22. R. Rama Devi, M. Abualkibash, Intrusion Detection System Classification Using Different Machine Learning Algorithms on KDD-99 and NSL-KDD

Datasets - A Review Paper, Int. J. Comput. Sci. Inf. Technol. 11 (2019) 65–80. https://doi.org/10.5121/ijcsit.2019.11306.
23. R. Yao, N. Wang, Z. Liu, P. Chen, X. Sheng, Intrusion detection system in the advanced metering infrastructure: A cross-layer feature-fusion CNN-LSTM-based approach, Sensors (Switzerland). 21 (2021) 1–17. https://doi.org/10.3390/s21020626.
24. S. Choudhary, N. Kesswani, Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT, Procedia Comput. Sci. 167 (2020) 1561–1573. https://doi.org/10.1016/j.procs.2020.03.367.
25. S. Elhag, A. Fernández, A. Bawakid, S. Alshomrani, F. Herrera, On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems, Expert Syst. Appl. 42 (2015) 193–202. https://doi.org/10.1016/j.eswa.2014.08.002.
26. S. Paliwal, R. Gupta, Denial-of-Service, Probing & Remote to User (R2L) Attack Detection using Genetic Algorithm, Int. J. Comput. Appl. 60 (2012) 57–62.
27. Thakkar, R. Lohiya, Attack classification using feature selection techniques: a comparative study, J. Ambient Intell. Humaniz. Comput. 12 (2021) 1249–1266. https://doi.org/10.1007/s12652-020-02167-9.
28. V.K. Rahul, R. Vinayakumar, K. Soman, P. Poornachandran, Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security, 2018 9th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2018. (2018). https://doi.org/10.1109/ICCCNT.2018.8494096.
29. Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, F. Ahmad, Network intrusion detection system: A systematic study of machine learning and deep learning approaches, Trans. Emerg. Telecommun. Technol. 32 (2021) 1of29. https://doi.org/10.1002/ett.4150.