Toward a new IDS based on PV-DM (Paragraph Vector-Distributed Memory Approach)

Chadia El Asry (chadia.elasry@um5r.ac.ma)
FSR, University Mohammed V
Samira Douzi
FMPR, University Mohammed V
Bouabid El Ouahidi
FSR, University Mohammed V

Research Article

Keywords: Intrusion detection systems, Deep learning, Machine learning, PV-DM, Feature selection

Posted Date: April 20th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1559488/v1

RESEARCH

Toward a new IDS based on PV-DM (Paragraph Vector-Distributed Memory Approach)

Chadia El Asry[1*], Samira Douzi[2*] and Bouabid El Ouahidi[1*]

*Correspondence:
chadia.elasry@um5r.ac.ma;
s.douzi@um5r.ac.ma;
b.elouahidi@um5r.ac.ma
[1]L.R.I, Faculty of Sciences, Mohammed V University, Rabat, Morocco
[2]FMPR, Mohammed V University, Rabat, Morocco
Full list of author information is available at the end of the article

Abstract

Intrusion detection systems (IDS) have become increasingly important in recent years as a means of assuring network security and eliminating undesirable conduct. Intrusion detection systems (IDSs) are either software applications or hardware devices that monitor network traffic for indications of malicious activity or policy violations. The study offers a detection technique based on feature selection, the Distributed Memory Paragraph Vector (PV-DM), and machine learning classifiers. We evaluated our proposed system using the NSL-KDD and UNSW-NB15 datasets. Experiments with multiclass classification demonstrate the efficacy of our approach, which achieves 98.92 percent accuracy, 98.92 percent precision, 95.44 percent recall, and 96.77 percent F1-score with only four features in NSL-KDD and 82.86 percent accuracy, 84.07 percent precision, 77.70 percent recall, and 80.20 percent F1-score with only six features in UNSW-NB15. Additionally, our proposed approach outperforms the standard LSTM model across the board.

Keywords: Intrusion detection systems; Deep learning; Machine learning; PV-DM; Feature selection

1 Introduction

The COV19 outbreak and extensive usage of teleworking in all government, military, and commercial organizations have resulted in a fourfold increase in cyber-attacks in 2020 [1]. As a result, having an effective intrusion detection system [2] capable of detecting continually changing attack programs is critical for assuring the security of data transferred between users.

IDSs are classified into two types: signature and anomaly intrusion detection systems. To address privacy concerns and security threats, signature-based IDSs frequently search a pre-defined database of security attack signatures for events and traffic that fit specific attack patterns [3]. However, signature-based IDS approaches are incapable of identifying unexpected patterns and signatures in new assaults [4]. On the other side, abnormality-based IDS approaches strive to understand normal behavior and classify everything else as an anomaly or intrusion [4]. They do, however, have a problem with false positives, which severely restricts their utility.

Numerous IDS approaches have been described in the literature that employ various deep learning models to automatically identify between normal and abnormal events in systems and networks [5]. Deep learning can model complex data structures using a variety of neural network designs in order to do various non-linear

data transformations and recognize certain patterns in data.

Deep learning is based on artificial neural networks (ANNs), which have a higher learning power and utilise several hidden layers for data transformation. Deep Belief Networks (DBNs), Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs), and Recurrent Neural Networks (RNNs) are different kinds of deep learning networks that have been widely used in applications such as security and intrusion detection [6].

Natural language processing (NLP) is a subfield of machine learning concerned with the use of a variety of computer algorithms to process naturally occurring texts. Modern NLP applications include information retrieval and extraction, question answering, text summarization, machine translation, and dialogue systems. Over the last decade, numerous research initiatives have begun to analyze and detect harmful patterns using natural language processing (NLP) methods [7]. PVDM is an unsupervised neural network model that learns context information from texts of varying lengths to construct fixed-length vectors. This architecture was developed by Quoc Le and Tomas Mikolov [8] for usage in the textual domain, although it is not confined to this domain and can be utilized in non-text domains as well.

Feature selection is the process of selecting a subset of relevant attributes from a dataset in order to construct robust IDS. Even for a small network, the amount of features that an IDS must explore from raw network data is typically huge. As a result, feature selection can be viewed as a valuable technique for developing classification models, as a large number of meaningless features can obstruct the classification process in a complex domain. Additionally, omitting non-essential features improves detection accuracy while increasing computation performance.

In this research, we propose an IDS based on a feature selection algorithm, the PV-DM technique, and machine learning models. To do this, we used two reduction techniques: Mutual Information (MI) and SHAP values to minimize superfluous features that impair classification performance. Additionally, we tested a variety of machine learning techniques to determine the most effective classifier and feature selection model. We compared our model to a widely used deep learning model in intrusion detection systems, LSTM; this is the first time, to our knowledge, that LSTM and PV-DM have been compared in the intrusion detection domain. Our approach outperforms the LSTM model in this comparison.

This paper is divided into the following sections: The next section discusses related works. Section 3 outlines the several notions that guided our search, Section 4 details the proposed approach in detail, including implementation and experimental results, and Section 5 closes the article.

2 Related Works

Multiple studies have been conducted on this subject, and numerous papers cover the various deep learning techniques used to develop successful IDS capable of defending against all types of attacks.

Mamoru [9] used a generic detection method (DOC2VEC) to conduct a timeline analysis and cross-dataset validation on D3M and MTA containing captured exploit Kit (EK) traffic; the effective result was the detection of an up-to-date EK traffic as indicated by an F-measure of 0.98 on the timeline analysis and 0.97 on the other dataset.

San [10] testing is carried out using skip-gram modeling, a variation of the word2vec technique. Indeed, the intrusion detection algorithm achieved precisions of 99.20 percent, recalls of 82.07 percent, and accuracy of 91.02 percent, with a false positive rate of 0.61 percent, which is far lower than the bulk of results, obtained using state-of-the-art approaches. This model is validated using a single dataset, the UNSW-NB15.

In [11], the authors propose a system that combines two deep learning models, LSTM and CNN. The CNN-LSTM model is evaluated on a single KDD99 dataset and with a single performance parameter, accuracy, which hits 99.78 percent.

[12] Proposed an artificial intelligence (AI) intrusion detection system based on a deep neural network (DNN) that was analyzed and validated using the KDD Cup 99 dataset. They employed four hidden layers to develop the neural network. The classification type was binary, and the accuracy was 99.08 percent (normal or attack).

In their study [13], the authors introduce a novel Deep Learning paradigm that blends Auto-Encoders and Deep Belief Networks (DBN). The Auto-Encoder was tasked with lowering the dimension of the data and determining its important characteristics, whilst the DBN was tasked with recognizing the suspect code. The new model recommendation was evaluated on the KDD Cup 99 dataset, with the outcomes compared using a single DBN. According to the achievement, the new procedure is significantly more accurate and efficient. However, the authors did not elaborate on why they combined DBN with Auto-Encoder to develop this hybrid system.

Kim et al. [14] discovered that the LSTM-RNN network beat all other methods on the KDD99 dataset, including the Generalized Regression Neural Network (GRNN), Product-based Neural Network (PNN), k-Nearest Neighbours (KNN), SVM, Bayesian, and others.

The authors of [15] develop a deep learning-based intrusion detection system (DL-IDS) that utilizes a hybrid network of Convolutional Neural Networks (CNN) and Long Short-Term Memory Networks (LSTM). The DL-IDS is tested in multiclassification on a single dataset, CICIDS2017, and the experimental result is 98.67 percent overall accuracy and above 99.50 percent accuracy for each attack type.

Yakubu and colleagues suggested a bidirectional Long-Short-Term Memory (BiDLSTM)-based intrusion detection system in [16] for training and evaluating their model's performance using the NSL-KDD dataset. The experimental results demonstrate that the BiDLSTM model outperforms LSTM and other state-of-the-art models in terms of accuracy, precision, recall, and F-score values for U2R and R2L attacks.

Ramadan [17] described a hybrid IDS approach that incorporates a pre-processing step to speed up the process required. They employed the Enhanced Shuffled Frog Leaping (ESFL) technique to minimize the number of features, and then categorised the remaining features using the Light Convolutional Neural Network with Gated

Recurrent Neural Network (LCNN-GRNN) algorithm. They attain an accuracy of 92.56 percent with KDD TEST-21 and 97.89 percent with KDD TEST +. They did not provide the number of features used in the experiment in the publication.
Shen [18] suggested an ensemble methodology that coupled the extreme learning machine (ELM) as a primary classifier with a pruning method based on the Bat Algorithm (BA) as an optimizer to achieve 98.94 percent accuracy. The suggested technique is evaluated using the KDD99, NSL-KDD, and Kyoto datasets.
Dong [19] introduced the AE AlexJNet, an intrusion detection model based on deep learning and the Auto-Encoder AlexNet neural network. The accuracy of the AE-AlexNet model is 94.32 percent when the KDD99 dataset is used.
The authors presented DL-MAFID in [20], which solves multi-class intrusion detection challenges by utilizing multi-agent systems and an auto-encoder. The KDD-Cup'99 dataset is used to assess this technique's capacity for dimension reduction. Additionally, shallow classifiers such as MLP and KNN are used to recognize the dataset's five classifications. Experiments have demonstrated that DL-MAFID is capable of achieving a 99.95 percent detection accuracy while simultaneously decreasing detection time.
Another approach to deep learning is provided in [21], where the authors offer three IDS for deep learning based on ANN, DNN, and RNN. The UNSW-NB15 is used to evaluate the suggested approaches' performance in binary and multiclass classification.
According to the related works stated above, the majority of study does not apply data feature engineering models, which is a critical step in identifying significant features, and while some studies do employ a reduction strategy, we discover that they are content with employing a single method. Second, the majority of studies validated their strategy using a single dataset and a binary condition. Finally, we note that recent research suggests IDS based on deep learning architectures, which are extremely complicated and comprised of multiple layers, significantly increasing execution and reaction times. However, IDS users require a system that is simple to use, quick to respond, and capable of identifying anomalies in a matter of seconds. Where did the concept for employing the PV-DM architecture originate. To our knowledge, no one has combined it with Mutual Information and SHAP value dimension reduction techniques for binary and multiclass classification.

3 Background

This section discusses the approaches of deep learning, the techniques of feature selection, and the metric used in the experimental study.

3.1 Deep Learning

Deep learning methods based on neural networks, such as LSTM and PV-DM, can simulate the activities of the human brain; the following subsections summarize these two methodologies:

3.1.1 ***Paragraph Vector-Distributed Memory (PV-DM):***

The PV-DM methodology is an unsupervised learning approach inspired by neural network-based approaches for learning continuous vector representations of words,

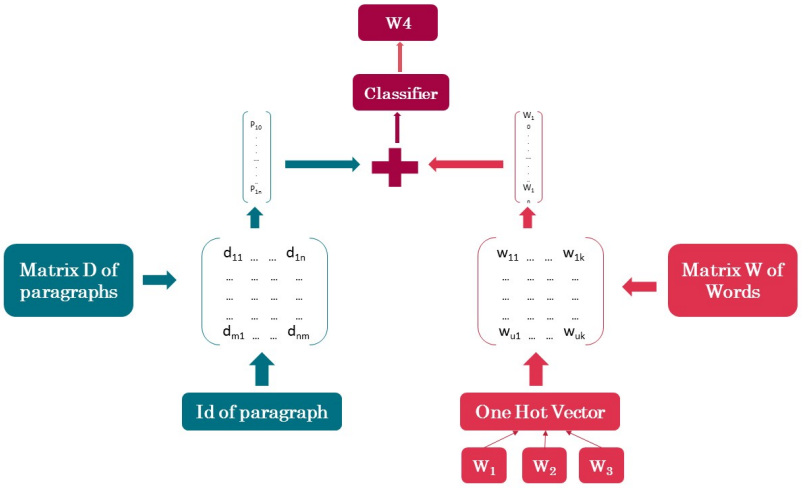


Figure 1: A framework for learning paragraph vector

specifically word2vec [22]. PV-DM's fundamental premise is that a paragraph P can be represented as an additional vector that aids in predicting the next word in a phrase [8] [8].

In the PV-DM model, each paragraph is translated into a single vector represented by a column in matrix D, and each word is mapped to a single vector represented by a column in matrix W Fig.1 [23]. As a result, a classifier (such as Softmax) that predicts the next word in a context averages and concatenates the paragraph and word vectors. We have the following:

$$p(w_t|w_{t-k},...,w_{t+k};d,W,D) = \frac{e^{y_{wt}}}{\sum_i e^{y_i}} \tag{1}$$

$$y = b + Uh(w_t|w_{t-k},...,w_{t+k};d,W,D) \tag{2}$$

Where: b and U are the classifier parameters, h is derived from W and D, and d is the vector of the paragraph from which the words $w_{t-k},\ldots,w_{t+k}$ be issue.

3.1.2 ***Long short-term memory (LSTM):***

LSTM (Hochreiter & Schmidhuber, 1997) is a special case of RNN [24] due to its memory. By adding three gates, an input gate, an output gate, and a forget gate, the disappearance and explosion gradient problems associated with conventional RNN training can be avoided. At time t, the input gate, output gate, and forget gate are denoted respectively by it, ot, and ft Fig.2 [25].

-Input/Update gate layer: decides if new values derived from incoming values can transit or not using two functions: "sigmoid" (zero prohibits modification, while one

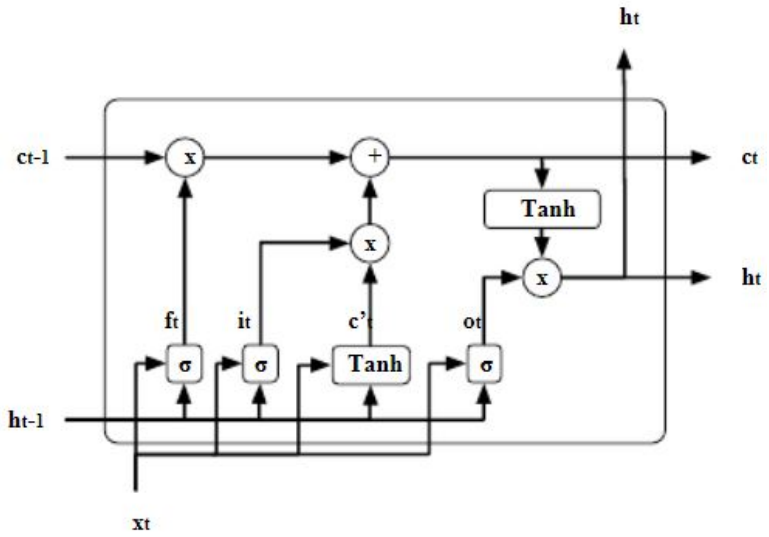


Figure 2: Long short-term memory (LSTM) architecture

permits modification) and the "tanh" function to generate a new vector of values to be appended to the cell state. It is calculated as follows:

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \, and \, \tilde{c_t} = tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{3}$$

- By applying a sigmoid function (keep (1) or forget (0)), the forget gate layer determines which values are allowed to pass between the current input and prior cell state output. It is calculated as follows:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{4}$$

- The Output Layer specifies what information should be output from the cell state. Prior to running the result through the "sigmoid," it is passed to the "tanh" function to guarantee that the values are between -1 and +1. It is calculated as follows:

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \, and \, h_t = o_t * tanh(c_t) \tag{5}$$

3.2 **Features selection:**

Feature selection is the process of picking a subset of significant features; it typically aids in reducing the computational cost of modeling, improving the model's performance, and making models easier for researchers to interpret.
To minimize the dimensionality of the datasets in our study, we compare two algorithms: Mutual Information (MI) and SHAP values.

3.2.1 ***Mutual Information (MI):***

Mutual information has been successfully used in filter feature selection algorithms to evaluate both the predictive ability of a subset of features and their redundancy with other variables [26] . MI calculated the amount of information that can be gleaned from observing another random variable and scoring each feature. When two variables are statistically independent, the MI value is 0, and it grows as the degree of addiction increases.
The MI is defined in the following manner:

$$I(X;Y) = \sum P_{XY}(x,y) log \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)} = E_{P_{XY}} log \frac{P_{XY}}{P_X P_Y} \tag{6}$$

Which X and Y are two discrete variable and $P_X(x)$ and $P_Y(y)$ are the marginals: $P_X(x) = \sum_y P_{XY}(x,y)$.

3.2.2 ***SHAP values:***

SHAP (SHapley Additive exPlanations), established by Lundberg and Lee (2017) [27], utilizes Shapley's classic values from game theory [28] to explain the model at the individual data point level. Shapley's value can be expressed in the following manner:

$$\varphi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M-|S|-1)!}{M!} (f_x(S \cup i) - f_x(s)) \tag{7}$$

Where M is the number of variables, S denotes a collection of variables, f_x denotes the prediction function at time x, and $f_x(S) = E[f(x)|x_s]$, where i denotes the i^{th} variable.
SHAP values are calculated by comparing model predictions with and without the feature using combinatorial calculus and retraining the model over all conceivable combinations of attributes that contain the one of interest. SHAP values enable the importance of each feature to be calculated for each data point by averaging the absolute values of the Shapley values computed on a particular dataset in order to generate "global" values for each variable.

3.2.3 ***Evaluation metrics:***

To more precisely quantify the effectiveness of our suggested intrusion detection system, we use the confusion matrix, which enables us to calculate true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) (FN). Accuracy, Sensitivity (or Recall), Specificity, and Precision are extracted from this matrix. These metrics are calculated in the following manner:The accuracy is the ratio of correctly predicted observations. Where:

- The accuracy is the ratio of correctly predicted observations:

$$Accuracy = \frac{TP+TN}{ALLPredictions} \tag{8}$$

• The recall is a proportion of correctly predicted positive events:

$$Recall = \frac{TP}{TP+FN} \tag{9}$$

• The precision means a ratio of correct positive observations:

$$Precision = \frac{TP}{FP+TP} \tag{10}$$

• The F1 score signifies the weighted average of precision and recall:

$$F1Score = 2 * (\frac{Presion*Sensitivity}{Precision+Sensitivity}) \tag{11}$$

4 Our approach

labelOA This section describes dataset preprocessing, feature selection strategies for calculating relevant features, the suggested model's implementation settings, experimental results, and discussion.

4.1 Datasets:

This study analyzed two datasets: the NSL-KDD Dataset [29] and the UNSW NB15 [30].

4.1.1 ***NSL-KDD dataset:***

NSL-KDD is an updated version of the KDD-cup 99 dataset [31]. This dataset is more dependable to use because it does not contain redundant or duplicate records. At the commencement of the NSLKDD dataset, there were 4 898 430 records, including 972 780 normal records and 3 925 650 attack records. The 38 threats include smurf, neptune, satan, ipsweep, and portsweep. NSL-KDD contains 43 features, which are classified into four groups in table 1.

Table 1: The four NSL KDD feature groups.

Features Type	Feature numbers	Total number
Categorical	2, 3, 4, 42	4
Binary	7, 12, 14, 20, 21, 22	6
Discrete	8, 9, 15, 23 to 41, 43	23
Continuous	1, 5, 6, 10, 11, 13, 16, 17, 18, 19	10

4.1.2 ***UNSW-NB15: (University of New South Wales –NB 2015):***

The UNSW-NB15 dataset [30] was created six years ago (2015) at the Australian Centre for Cyber Security (ACCS) for intrusion detection research purposes, by utilizing an IXIA tool to extract a mix of normal and attack activity. The dataset contains 49 features classified as Binary, Float, Integer, Nominal, and Timestamp, in addition to a variety of normal and attacked behaviors. Fuzzers, Analysis, Backdoors, Denial of Service, Exploits, Generic, Reconnaissance, Shell code, and Worms

are only a few of the nine attack types covered in UNSW-NB15.
Numerous academics do intrusion detection system testing using the NSLKDD and UNSW-NB15 datasets. We discover that the number of attack records surpasses the number of normal records, and that some assaults have a very low record count in comparison to other attacks, resulting in a data imbalance problem.

4.1.3 ***Implementation:***

Python 3.7.1 was used as the programming language and Google Collab as the integrated development environment, respectively. Several libraries were consulted (Table.2).

Table 2: Some of the libraries that were used in the implementation.

Libraries	**Description**
NumPy	It is one of the libraries offered in the Python language that handles scientific computing for machine learning.
Sklearn(also known as Scikit-Learn)	Contains various functions that help with model selection, dimensionality reduction, pre-processing, clustering, regression and classification. Sklearn is very simple to use and most necessary efficient as it is built on NumPy, SciPy and matplotlib.
Pandas	Pandas is a popular software built on top of the Python programming language. It is easy data analysis toolkit, which can be imported.
Gensim	this is one of the Python library for document indexing, similarity retrieval and topic modelling,using modern statistical machine learning
Keras	Keras is a neural network API library built in Python and interfaceable with frameworks like Theano and TensorFlow.

4.2 **Datasets preprocessing:**

4.2.1 - Grouping of attacks:

We classified the attacks in the NSL-KDD dataset into four groups in this phase, as shown in table 3:

The distribution of attack and normal records in the UNSW-NB15 and NSL-KDD dataset is shown in Fig.2a and Fig.2b.

Due to the fact that both datasets contain a relatively small number of records for some types of attacks, as described previously and as illustrated in Fig.2a and Fig.2b, we chose to focus our research on the first four categories of attacks in each dataset Table.4.

4.2.2 - Label conversion and feature numericalization:

In this phase, we assigned a numerical value to each attack type; for NSL-KDD, we assigned a value of 0 to Normal, 1 to DoS, 2 to Probe, 3 to R2L, and 4 to U2L. We set Normal to 0 for UNSW-NB15, Generic to 2, Exploits to 3, and Fuzzers to 4.
Due to the fact that the majority of machine learning algorithms do not allow categorical data, it must be converted to numerical data before it can be used by the machine. We changed the non-numeric features ('protocol-type, service, and flag' for NSL-KDD and 'proto, service, and state' for UNSW-NB15) to numbers using the pandas. factorize() method .

Table 3: Attacks categories in NSLKDD dataset.

Attack Category	Description	Attack type
DoS	Denial of service is a malicious attack aimed at preventing normal users from using a service by making it inaccessible.	neptune, back, land, pod , smurf, teardrop, udpstorm, mailbomb, apache2, processtable, worm
Probe	Probe or surveillance is an attack that her purpose is to obtain important information about the security of networks in order to change with the security settings.	ipsweep, nmap, portsweep, satan, mscan, saint
R2L	This class of attacks tries to gain local unauthorized access to a remote machine by sends packets to the network.	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, sendmail, named, snmpgetattack, snmpguess, xlock, xsnoop, httptunnel
U2R	The primary purpose of this attack is to illegally explore or steal data, install viruses, or cause the sufferer to suffer harm by introduction a normal user account.	buffer_overflow , Loadmodule, perl, rootkit, ps, sqlattack, xterm

Table 4: The four NSL KDD feature groups.

Datasets	Categories of attacks
NSL-KDD	Normal, DoS, Probe, R2L
UNSW-NB15	Normal, Generic, Exploits, Fuzzers

4.2.3 - Standardization:

Standardization is a term that refers to a scaling approach in which data are centered on the mean and have a unit standard deviation. Standardization is essential since the maximum and minimum values for numerous characteristics in the feature space, such as time, src bytes, and dst bytes, are extremely different. This is accomplished by the use of the StandardScaler [33] algorithm, which is calculated as:

$$z = \frac{(x_i - mean(x))}{stdev(x)} \tag{12}$$

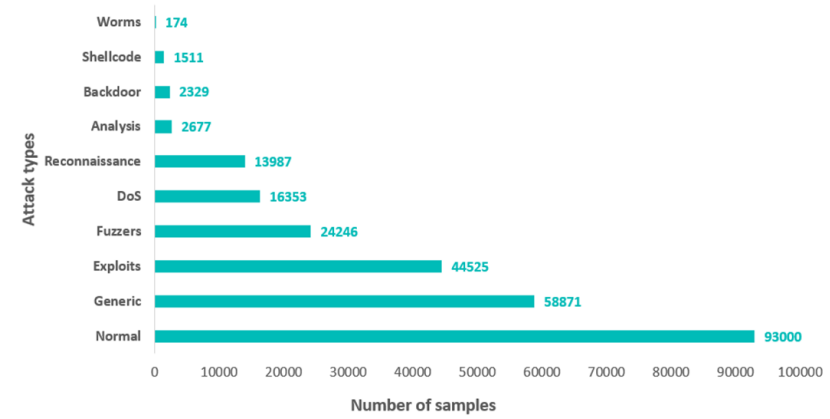
Where mean(x) is the mean of the training sample, stdev(x) is the standard deviation of the training sample, and x_i is the feature value.

4.3 MI and SHAP values:

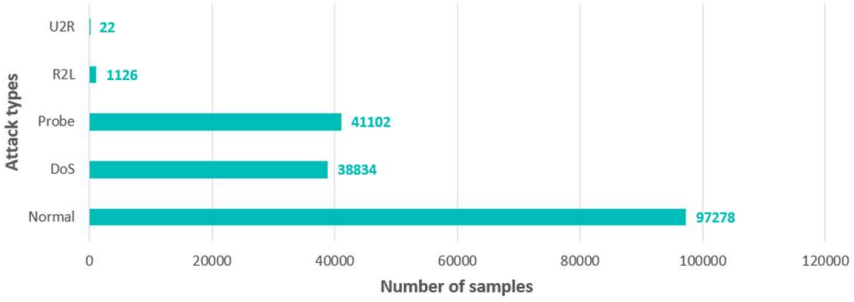
4.3.1 ***Mutual Information (MI) Algorithm:***

As previously stated, mutual information determines the degree of reliance between two variables. In this study, we use Mutual Information to determine the magnitude of each feature's effect on the goal (in both data sets).

As shown in Fig.3b, the mutual information scores for features src_bytes, service, count, dst_bytes, srv_count,logged_in, dst_host_diff_srv, rate, dst_host_srv_count, dst_host_same_src_port_rate, and dst_host_srv_diff_host_rate are the highest in the NSL-KDD dataset. On the other hand, Mutual Information assigns the highest scores to the UNSW-NB15 dataset's attribute (Fig.3a) sbytes, sload, smean, dbytes, rate, dur, ct_dst_sport_ltm, dmean, ct_state_ttl, and service. While the other at-



(a) Attack and normal records in the UNSW-NB15



(b) Attack and normal records in the NSL-KDD

Figure 3: Attack and normal records in UNSW-NB15 and NSL-KDD datasets.

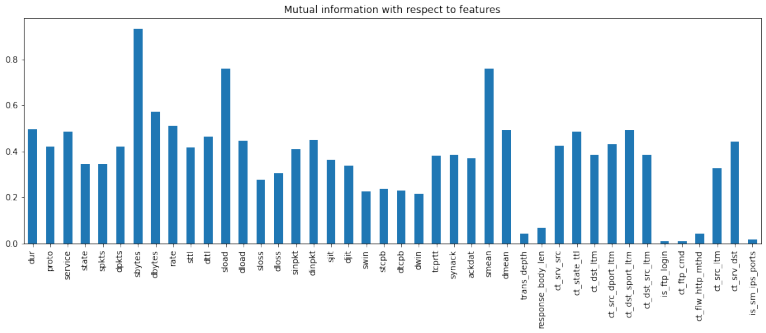
tributes in both datasets scored low, indicating that they had no discernable effect on the target.

4.3.2 ***SHAP values Algorithm :***

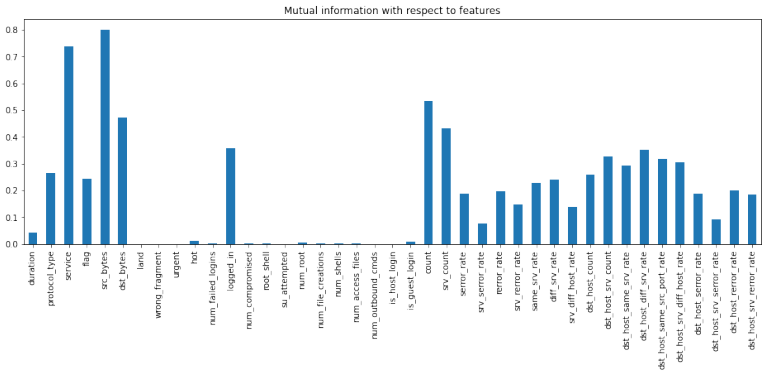
As previously stated, SHAP values indicate the contribution of each feature to target prediction. To determine the relevance of a characteristic, the outcome of any possible combination of features is considered (Fig.5). As illustrated in Fig.5a and Fig.5b, the following NSL-KDD characteristics have the highest SHAP values: "src_bytes, dst_bytes, service, dst_host_diff_srv_rate, dst_host_serror_rate, rerror_rate, dst_host_rerror_rate, dst_host_same_src_port_rate, count, and hot". Whereas the 10 most valuable features for UNSW-NB15 are ,"sttl, sbytes, ct_dst_src_ltm, proto, dur, ct_srv_dst, ct_state_ttl, response_body_len, sloss, and dbytes".

Additionally, as indicated in Figures Fig.3b and Fig.5b, Mutual Information for NSL-KDD identified features srv_count, logged_in, dst_host_srv_count, and dst_host_srv_diff_host_rate as being among the top ten features with a high score; nevertheless, SHAP values did not rank them among these top ten features, but did rank other features such as dst_host_serror_rate, rerror_rate, dst_host_serror_rate and hot.

For UNSW-NB15, we discovered that mutual information and SHAP values share



(a) UNSW-NB15 feature Mutual Information scores.



(b) NSL-KDD feature Mutual Information scores.

Figure 4: Mutual information calculus.

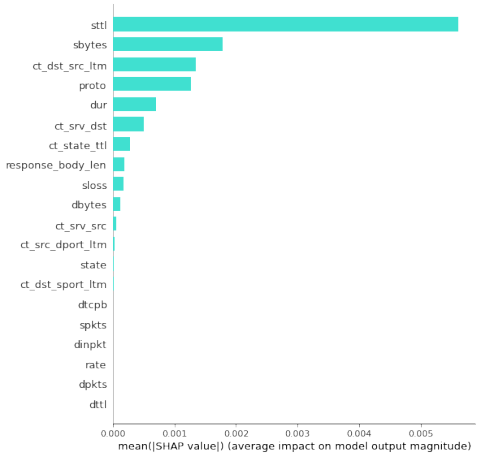
Table 5: Features of reduced Datasets.

Reduced Dataset	**Features Included**
Red_NSL_MI	5, 3, 23, 6, 24, 12, 35, 33, 36, 37
Red_NSL_ShV	5, 6, 3, 35, 38, 27, 40, 36, 23, 10
Red_UNSW_MI	4, 9, 24, 5, 6, 1, 32, 25, 29, 41
Red_UNSW_ShV	7, 4, 33, 40, 1, 38, 29, 27, 11, 5

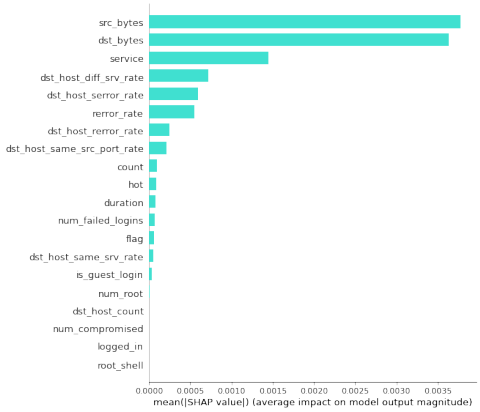
the feature sbytes, dbytes, dur, and ct_state_ttl but have unique scores.
After reducing both datasets to their ten highest features using Mutual Information and SHAP values techniques, as shown in Table.5, we obtain four reduced datasets. We acquire four reduced datasets after reducing both datasets to simply the 10 highest features picked using Mutual Information and SHAP values algorithms, as shown in Table.5.

4.4 **PV-DM model:**

Following the feature reduction procedure, we employ the PV-DM model to generate embedding vectors for use as inputs to the machine learning models. This is accomplished through the employment of six sets, four of which are listed in Table.5, as well as the original sets (NSL KDD and UNSW15), which contain all features. The PVDM model was constructed using the parameters given forth in Table.6. We strive to optimize our approach by identifying the most accurate feature selection method and classifier in terms of accuracy, precision, recall, F1-score,



(a) The SHAP Value indicates the significance of features in UNSW-NB15.



(b) The SHAP Value indicates the significance of features in NSL-KDD.

Figure 5: The SHAP Value indicates the significance of features.

Table 6: Table of parameters of the PV-DM.

Parameters of PVDM Model	**Values**
Number of features	All,10
min_count	5
window	8
vector_size	100
sample	1e-4
negative	5
workers	4
alpha	0.025
min_alpha	0.025
epochs	100

and AUC (Fig.6). As demonstrated in Table.7, we divided each set's embedding PV-DM vectors into training and testing vectors and classified them using five classifiers: SVM, RANDOM-FOREST, LOGISTIC REGRESSION, GAUSSIAN-NB, and XGBOOST.

Table 7: The size of training set and testing set for each set.

Embedding vectors of	Train size	Test size
Red_NSL_MI, Red_NSL_ShV , NSL-KDD	125061	53279
Red_UNSW_MI , Red_UNSW_ShV , UNSW-NB15	176514	44128

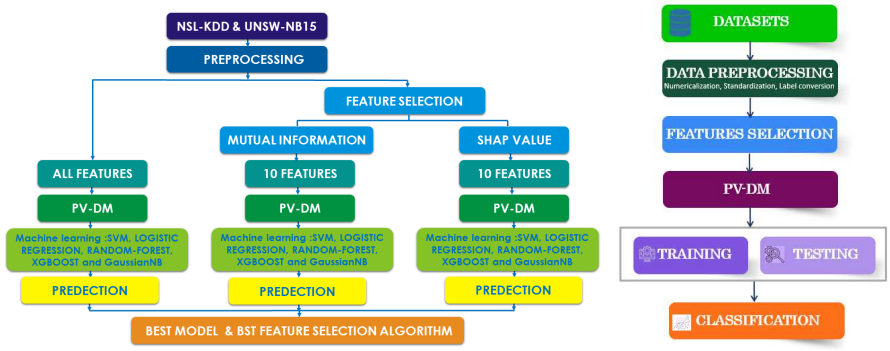


Figure 6: Architecture of experiment of choose the best classifier and feature selection algorithm with PV-DM

4.4.1 ***Experimental results and discussion :***

The accuracy, recall, precision, F1-score, and AUC of each model are depicted in Tables.8,9,10,11,12,13. Additionally, Fig.7 presents the Roc plot for XGBOOST classifier.

Table 8: Classification performance of PV-DM with All features, with five classifiers for UNSW-NB15

	SVM%	RANDOM-FOREST%	LOGISTIC REGRESSION%	GAUSSIAN-NB%	XGBOOST%
ACCURACY%	55.99	71.19	56.96	32.77	**77.44**
PRECISION%	48.61	89.84	72.48	42.88	**83.68**
RECALL%	73.35	60.10	51.08	63.00	**71.34**
F1-SCORE %	54.39	66.20	57.06	50.16	**75.54**
AUC %	71	79	72	68	**84**

The best result is in bold blue color

Table 9: Classification performance of PV-DM with MI using 10 features, with five classifiers for UNSW-NB15

	SVM%	RANDOM-FOREST%	LOGISTIC REGRESSION%	GAUSSIAN-NB%	XGBOOST%
ACCURACY%	68.31	79.36	68.02	55.15	**80.97**
PRECISION%	73.97	86.25	74.80	59.32	**83.30**
RECALL%	58.02	69.14	59.45	72.54	**74.74**
F1-SCORE %	61.35	73.22	63.51	64.29	**77.74**
AUC %	76	83	77	79	**85**

The best result is in bold blue color

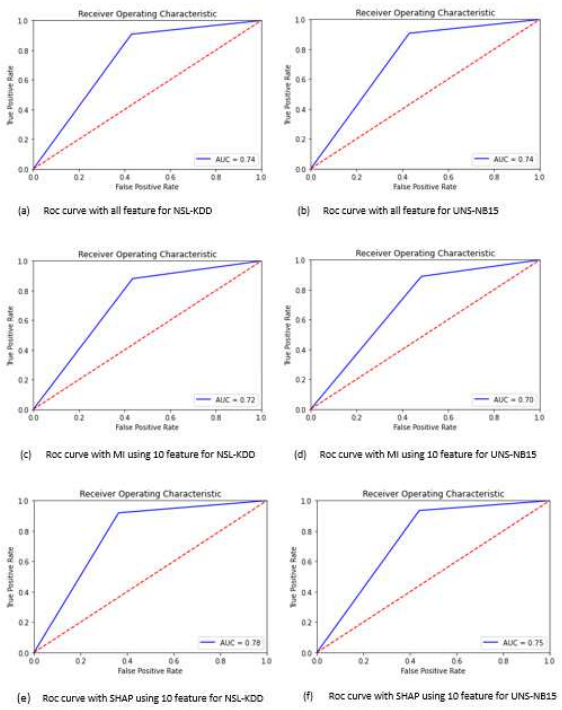


Figure 7: The ROC curve of the XGBOOST classifier

Table 10: Classification performance of PV-DM with SHAP values using 10 features, with five classifiers for UNSW-NB15

	SVM%	RANDOM-FOREST%	LOGISTIC REGRESSION%	GAUSSIAN-NB%	XGBOOST%
ACCURACY%	73.53	81.81	73.24	60.36	**82.84**
PRECISION%	78.93	88.94	78.19	62.27	**84.66**
RECALL%	64.74	72.89	66.55	76.41	**78.11**
F1-SCORE %	68.24	77.54	70.42	67.93	**80.72**
AUC %	80	85	81	81	**87**

The best result is in bold blue color

Table 11: Classification performance of PV-DM with All features for NSL-KDD, with five classifiers for UNSW-NB15

	SVM%	RANDOM-FOREST%	LOGISTIC REGRESSION%	GAUSSIAN-NB%	XGBOOST%
ACCURACY%	79.56	90.73	79.60	37.19	**93.79**
PRECISION%	70.57	94.25	71.68	40.06	**93.95**
RECALL%	63.67	69.16	68.82	68.20	**82.80**
F1-SCORE %	65.09	74.13	70.19	45.10	**86.89**
AUC %	79	83	81	72	**91**

The best result is in bold blue color

We discovered that the PV-DM with XGBOOST classifier consistently beat all other classifiers in terms of accuracy and recall, which are crucial for classification outcome. Additionally, while the PV-DM with Gaussian-NB has the lowest scores, whether with MI or SHAP values, its recall is greater than that of SVM, RANDOM-FOREST, and LOGISTIC REGRESSION, indicating that it successfully predicts attacks.

Table 12: Classification performance of PV-DM with MI using 10 features for NSL-KDD, with five classifiers for UNSW-NB15

	SVM%	RANDOM-FOREST%	LOGISTIC REGRESSION%	GAUSSIAN-NB%	XGBOOST%
ACCURACY%	86.40	93.07	86.64	43.74	**96.10**
PRECISION%	81.30	97.29	78.23	43.21	**95.45**
RECALL%	74.50	75.13	75.89	70.97	**86.94**
F1-SCORE %	77.15	81.11	77.03	49.40	**90.40**
AUC %	85	86	86	75	**93**

The best result is in bold blue color

Table 13: Classification performance of PV-DM with SHAP values using 10 features for NSL-KDD, with five classifiers for UNSW-NB15

	SVM%	RANDOM-FOREST%	LOGISTIC REGRESSION%	GAUSSIAN-NB%	XGBOOST%
ACCURACY%	83.40	95.27	83.67	44.28	**96.54**
PRECISION%	79.47	98.37	76.63	43.08	**97.42**
RECALL%	69.79	84.10	73.66	75.56	**92.94**
F1-SCORE %	72.93	89.41	75.09	47.52	**94.89**
AUC %	83	91	85	77	**96**

The best result is in bold blue color

On the other hand, we can see that, in comparison to MI, using SHAP values as a feature selection method consistently delivers the best performance results. Even if MI is less performant than SHAP values, it is clear that PV-DM with Mutual Information improves the performance of all measures with all classifiers when compared to PV-DM without Mutual Information.
In summary, based on the data above, we conclude that XGBoost and SHAP values are the optimal algorithms for combining with PV-DM.
Our proposed model is depicted in Fig.8, and in the following section, it will be compared against LSTM.

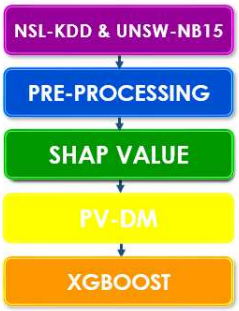


Figure 8: Our proposed model

4.4.2 ***Comparison of our method versus LSTM:***

We will compare our method to the most efficient and extensively used intrusion detection method, LSTM, in this part (Fig.9). The goal of this comparison is to assess our model's performance with varied numbers of features, with a focus on minimizing the false negative rate.

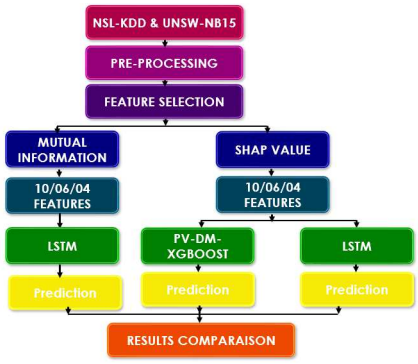


Figure 9: Architectures of comparison implemented in this research

Table 14: LSTM Model Parameters.

Parameters	**LSTM values**
Number of features	10,06,04
Activation function	Sparse categorical crossentropy
Loss function	8
Optimiser	Nadam
Learning rate	0.002
Epsilon	1e-08
Schedule decay	0.004
Epoch number	0.025
dropout	0.3

We build LSTM networks with 10, 6, and 4 features for pattern recognition. In this network, only one hidden layer was used. Table.14 shows the parameter values for the proposed LSTM model.

4.4.3 ***Result of comparison:***

Table.15 and Table.16 summarize each model's accuracy, sensitivity (recall), precision, and F1 scores.

Table 15: Performances of classification on NSL-KDD reduced sets.

MODEL	FEATURES	ACCURACY %	PRECISION%	RECALL%	F1-SCORE%
LSTM-MI	10	**97.77**	94.9	87.0	90.1
	06	91.85	74.1	77.7	75.7
	04	83.70	82.3	58.6	57.9
LSTM-SHAP VALUES	10	**99.19**	92.6	85.1	87.9
	06	97.80	98.1	76.8	79.2
	04	89.14	81.4	68.9	67.9
OUR PROPOSED MODEL	10	96.54	**97.42**	**92.94**	**94.89**
	06	**98.76**	**98.86**	**95.55**	**97.10**
	04	**98,92**	**98,92**	**95,44**	**96,77**

The best results is in bold

When we employed ten MI-selected features for NSL-KDD and UNSW-NB15, we found that LSTM produced the most accuracy, but our model produced the highest precision, recall, and f1-score.

Whereas LSTM with MI (06 and 04 features) degrades in accuracy, precision, recall,

Table 16: Performances of classification on UNSW15 reduced sets.

MODEL	FEATURES	ACCURACY %	PRECISION%	RECALL%	F1-SCORE%
LSTM-MI	10	**84.81**	78.1	78.1	77.9
	06	76.22	71.1	62.5	61.5
	04	72.30	68.5	57.0	56.1
LSTM-SHAP VALUES	10	81.44	77.9	69.5	69.7
	06	75.00	70.0	72.8	70.5
	04	50.19	53.1	56.7	49.3
OUR PROPOSED MODEL	10	82.84	**84.66**	**78.11**	**80.72**
	06	**82.86**	**84.07**	**77.70**	**80.20**
	04	**83.27**	**83.55**	**77.70**	**79.81**

The best results is in bold

and f1-score, this suggests that LSTM is unable to distinguish attacks effectively. Our technique using 06 and 04 attributes, on the other hand, gave the best outcomes and boosted the performance of measures.

Using Shap value as a reduction dimensionality approach with LSTM, on the other hand, reveals that our proposed methodology offered the best accuracy, precision, recall, and f1-score for reduced UNSW-NB15 set, with 10, 06, and 04 features. We notice that the performances of NSL-KDD are comparable, except for the situation of 10 features, where the accuracy value for LSTM is much greater than that of our methodology.

It can be demonstrated that by removing extraneous characteristics and selecting only the relevant elements, LSTM performs badly in terms of performance measurements, indicating that LSTM does not correctly discriminate the assaults. We explain that the LSTM asks for a large quantity of data in order to learn about risks, and it takes each sequence on its own, which explains why its performance degrades when feature selection methods are used. Our PV-DM-based strategy, on the other hand, considers both the global and local context of each value, and even if the dataset is unbalanced, our model becomes more efficient in terms of recognizing attacks, with a recall of 95.44 utilizing only four features in the NSL-KDD dataset.

The global performance of our proposed architecture, particularly the recall value, is more efficient than that of the LSTM using selection features methods, which is a critical step toward more accurately recognizing various types of attacks, as well as shortening training and testing times and displaying an improved intrusion detection rate[34].

Furthermore, when utilizing the NSL-KDD dataset, the findings for all measures are superior to those for the UNSW-NB15 dataset; this is because the UNSW-NB15 dataset is more complex than the NSL-KDD dataset when viewed from various perspectives. To begin, the UNSW-NB15 data collection comprises a wide range of current attack and normal network traffic aspects. On the other hand, the attack on the NSL-KDD data set and usual behavior are no longer applicable. Furthermore, the similarities between normal and assault observations in the majority of the features in the UNSW-NB15 dataset add to the dataset's complexity [35].

In order to better the analysis of the experimental results, we compared the results of our suggested approach to those of previous studies Table.17. Because IDSs

differ in terms of execution environment, data pre-treatment techniques, and interpretation process, this comparison is just illustrative. Our model outperforms the comparison models, as shown in Table 11, suggesting that our approach is better suited to this type of problem.

Table 17: Performances comparaison.

MODEL	FEATURE SELECTION METHOD	Datasets	ACCURACY%	PRECISION%	RECALL%	F1-SCORE%
LCNN-GRNN [17]	Enhanced ShuffledFrog Leaping (ESFL)	NSLKDD	92.56	-	-	-
RNN-LSTM [21]	-	UNSW-NB15	85.38	-	-	-
DNN [36]	04 features	NSLKDD	78,1	-	-	-
	-	UNSW-NB15	64,5	61,4	64,5	58,6
Decision Tree with 05 features [37]	CPIO (Continuous Pigeon Inspired Optimizer) (05 features)	NSLKDD	88.3	-	-	88.2
two-stage classification algorithm (rotationforest and bagging) [38]	Hybrid (GA+PSO+ACO)(37 features)	NSLKDD	85,79	88,0	86,8	-
Proposed model	SHAP values (04 features)	NSLKDD	98,92	98,92	95,44	96,77
	SHAP values(06 features)	UNSW-NB15	82.86	84.07	77.70	80.20

5 Conclusion

In this work, we provide an effective intrusion detection system based on Paragraph Vector-Distributed Memory (PV-DM), the SHAP value feature selection technique, and the XGBoost Classifier.

We compared our strategy to Long-Short Term Memory (LSTM). Despite the fact that the two datasets employed in this study, NSL-KDD and UNSW-NB15, are unbalanced, the experimental results demonstrate that our suggested technique outperforms LSTM with feature selection algorithms in terms of performance and attack knowledge.

When we reduce the number of features specified, the performance indicators of LSTM deteriorate; yet, our model preserves its performance.

6 Future works

We will continue to explore many LSTM variants (such as Weighted LSTM, Peephole LSTM, and Multiplicative LSTM) as well as additional deep learning algorithms like as CNN and ANN in the future.

Additionally, we will attempt to address the issue of unbalanced data in the NSL-KDD and UNSW-NB15 datasets and validate our strategy using these datasets.

Acknowledgements
Not applicable.

Funding
Not applicable. This research received no specific grant from any funding agency.

Availability of data and materials
Not applicable. For any collaboration, please contact the authors.

Ethics approval and consent to participate
The author confirms the sole responsibility for this manuscript. The author read and approved the final manuscript

Competing interests
The authors declare that they have no competing interests.

Contributions
All authors read and approved the final manuscript.

Consent for publication

Author details
[1]L.R.I, Faculty of Sciences, Mohammed V University, Rabat, Morocco. [2]FMPR, Mohammed V University, Rabat, Morocco.

References
1. ANSI , https://www.ssi.gouv.fr/uploads/2020/12/anssi-communique_presse common_situational_picture_2020.pdf.
2. Heady, R. , Luger, George , Maccabe, Arthur and Servilla, Mark The architecture of a network level intrusion detection system. *arXiv preprint arXiv:1409.0473*,10.2172/425295,1990.
3. Rajasekaran, K. Classification and Importance of Intrusion Detection System. *International Journal of Computer Science and Information Security*, 10. 44, 2020.
4. Othman, Suad & Nabeel, Taha & Ba-Alwi, Fadl & Zahary, Ammar. Survey on Intrusion Detection System Types. *International Journal of Cyber-Security and Digital Forensics*, 7. 444-462, 2018.
5. Teresa F Lunt. A survey of intrusion detection techniques. *Computers and Security*, Volume 12, Issue 4,Pages: 405–418, ISSN 0167-4048, 1993.
6. L. Lipton, Zachary C., John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. In , arXiv preprint arXiv: 1506.00019, 2015.
7. Das, Saikat , Ashrafuzzaman, Mohammad, Sheldon, F.T. and Shiva, S. Network Intrusion Detection using Natural Language Processing and Ensemble Machine Learning *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 829-835, doi: 10.1109/SSCI47803.2020.9308268.
8. Le, Quoc & Mikolov, Tomas. Distributed Representations of Sentences and Documents. In *31st International Conference on Machine Learning*, ICML 2014.
9. Mimura, Mamoru & Tanaka, Hidema. Reading Network Packets as a Natural Language for Intrusion Detection. 10.1007/978-3-319-78556-1_19, 2018.
10. Rafael San Miguel Carrasco, Miguel-Angel Sicilia. Unsupervised intrusion detection through skip-gram models of network behavior. *Computers and Security* Volume 78,2018, Pages 187-197, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2018.07.003.
11. Kottapalle, Prasanna. A CNN-LSTM Model for Intrusion Detection System from High Dimensional Data. *Journal of Information and Computational Science*, 10. 1362-1370., 2020.
12. Jin Kim, Nara Shin, Seung Yeon Jo and Sang Hyun Kim. Method of Intrusion Detection using Deep Neural Network. In *IEEE*, 2017.
13. Li Y., Ma R. and Jiao R., A Hybrid Malicious Code Detection Method Based on Deep Learning. *International Journal of Security and Its Applications (IJSIA)*, vol. 9, no. 5, pp. 205–216, 2015.
14. J. Kim, J. Kim, H. L. T. Thu, and H. Kim, Long short term memory recurrent neural network classifier for intrusion detection. In *Proceedings of the 2016 International Conference on Platform Technology and Service (PlatCon), IEEE, Jeju, South Korea* , February 2016
15. Sun, Pengfei & Liu, Pengju & Li, Qi & Liu, Chenxi & Lu, Xiangling & Hao, Ruochen & Chen, Jinpeng. DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system networks. *Security and Communication Networks*, 2020. 1-11. 10.1155/2020/8890306.
16. Imrana, Yakubu & Xiang, Yanping & Ali, Liaqat & Abdul-Rauf, Zaharawu. A bidirectional LSTM deep learning approach for intrusion detection. *Expert Systems with Applications. 185. 115524. 10.1016/j.eswa.2021.115524*, 2021.
17. Ramadan RA, Yadav K. A novel hybrid intrusion detection system (IDS) for the detection of internet of things (IoT) network attacks. *Ann Emerg Technol Comput*, 2020. https:// doi. org/ 10. 33166/ aetic. 2020. 05. 004.
18. Shen Y, Zheng K, Wu C, Zhang M, Niu X, Yang Y. An ensemble method based on selection using bat algorithm for intrusion detection. *Comput J*, 2018;61(4):526–38.
19. Dong Y, Wang R, He J. Real-Time Network Intrusion Detection System Based on Deep Learning. *IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, 2019. https:// doi. org/ 10. 1109/ icsess47205. 2019. 90407 18.
20. F. Louati and F. B. Ktata, A deep learning-based multi-agent system for intrusion detection, *Social Netw. Appl. Sci.*, vol. 2, no. 4, pp. 1–13, Apr. 2020.
21. A. M. ALEESA, , MOHAMMED YOUNIS, AHMED A. MOHAMMED, NAN M. SAHAR, DEEP-INTRUSION DETECTION SYSTEM WITH ENHANCED UNSW-NB15 DATASET BASED ON DEEP LEARNING TECHNIQUES *Journal of Engineering Science and Technology*, 2021.
22. Mikolov, Tomas & Chen, Kai & Corrado, G.s & Dean, Jeffrey, Efficient Estimation of Word Representations in Vector Space. *Proceedings of Workshop at ICLR*, 2013.

23. Douzi, Samira & Amar, Meryem & Ouahidi, Bouabid & Laanaya, Hicham. Towards A new Spam Filter Based on PV-DM (Paragraph Vector-Distributed Memory Approach). *Procedia Computer Science*,2017, 110. 486-491. 10.1016/j.procs.2017.06.130.
24. Zhao, Z., Chen, W., Wu, X., Chen, P. C. and Liu, J. Lstm Network: A Deep Learning Approach for Short-Term Traffic Forecast. *IET Intelligent Transport Systems*,2017,11, 68-75. https://doi.org/10.1049/iet-its.2016.0208.
25. Pisa, Ivan & Vilanova, Ramon & Santín, Ignacio & Lopez Vicario, Jose & Morell, Antoni. Artificial Neural Networks Application to Support Plant Operation in the Wastewater Industry. ,2019,10.1007/978-3-030-17771-3_22.
26. Cover TM, Thomas JA. Information theory and statistics. *Elements of information theory. 2nd edn. Wiley* ,2005. https:// doi. org/ 10. 1002/ 04717 4882X. ch11.
27. Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. *In Advances in Neural Information Processing Systems 30. Curran Associates, Inc., 4768–4777.* ,2017.
28. L. S. Shapley, A value for n-person games, *Contributions to Theory Games, vol. 2, no. 28, pp. 307–317, 1953.*
29. NSL-KDDdataset, [online]Available:http://nsl.cs.unb.ca/nsl-kdd/.
30. Moustafa, Nour, and Jill Slay. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *Military Communications and Information Systems Conference (MilCIS)*,2015. IEEE, 2015.
31. M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, A Detailed Analysis of the KDD CUP 99 Data Set, *Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA),* ,2009.
32. Dhanabal L, Shantharajah S. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms, *Int J AdvRes Comput Commun Eng* ,2015;4(6):446–52.
33. StandardScaler algorithm, https://www.datacorner.fr/feature-scaling/
34. M. Alazab, S. Venkatraman, P. Watters and M. Alazab, Zero-day malware detection based on supervised learning algorithms of API call signatures, *Proc. 9th Australas. Data Mining Conf., vol. 121, pp. 171-182,*,2011.
35. Nour Moustafa & Jill Slay, The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 dataset. ,2016,1-14. 10.1080/19393555.2015.1125974.
36. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat and S. Venkatraman, Deep Learning Approach for Intelligent Intrusion Detection System, *in IEEE Access, vol. 7, pp. 41525-41550*,2019, doi: 10.1109/ACCESS.2019.2895334.
37. Wu, K.; Chen, Z.; Li, W. A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks, *IEEE Access* ,2018, 6, 50850–50859. [CrossRef]
38. Adhi Tama Bayu, Comuzzi Marco, Rhee Kyung Hyune, TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-based Intrusion Detection System, *IEEE Access 7. 10.1109/ACCESS.*,2019.2928048.