IOSR Journal of Mobile Computing & Application (IOSR-JMCA)
e- ISSN: 2394-0050, P-ISSN: 2394-0042.Volume 7, Issue 1 (Jan. – Feb. 2020), PP 51-58
www.iosrjournals.org

Survey and performance analysis of Machine learning based Intrusion detection approaches in wireless sensor networks

Seema M.Shinde YeshwantMahavidyalaya.Nanded
Dr.Thorat S.B.
Director, ITM College, Nanded
Miss.Tazeen Khan
Software Engineer,Bengaluru,Karanataka
Dr.PravinTamsekar
ITM College, Nanded

Abstract
Nowadays, there is remarkable growth in technology and wireless sensor networks. These are primarily used for communication. The medium of communication between devices may be wired or wireless, hence, the chance of attacks through the networks is increasing daily. For secure communication, intrusion detection and prevention are primary concerns. Thus, study and analyses of intrusion detection and prevention techniques are the necessity to secure the network. With the assistance of intrusion detection and prevention systems, we can determine and then notify the normal and abnormal activities of the users. Thus, there is a requirement to design effective intrusion detection and prevention system by the use of machine learning for wireless sensor networks. In this paper, we present a survey and a comparative performance analysis of machine learning based approaches for intrusion detection in networks. The performance evaluation of these techniques is done by experiments conducted on the NSL-KDD dataset. In this work, we analyse machine learning models including Support Vector Machine (SVM), Decision Tree, Naive Bayes, Random Forest, and K-Nearest Neighbour. Besides, we used the most important performance indicators, namely, accuracy, precision, recall and f1 score for evaluating the efficiency of several methods.

I. Introduction

Any kind of illegitimate or unapproved behaviour in a network or a system will be considered as intrusions. An Intrusion Detection System (IDS) is a set of the tools, methods, and resources to facilitate distinguish, evaluate, and description intrusions [2]. Intrusion detection is a defence system that can detect abnormal activity. Intrusion is defined as: “any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource” [3]. IDSs are forever measured as a subsequent wall of defence from the security point of analysis. IDSs can be deployed along with other security measures, such as access control, authentication mechanisms, and encryption techniques to better secure the systems against attacks. Using patterns of benign traffic or normal behaviour or specific rules that describe a specific attack, IDSs can distinguish between normal and malicious actions [7]. According to Dewa and Maglaras [8], data mining which is used to describe knowledge discovery can help to implement and deploy IDSs with higher accuracy and robust behaviour as compared to traditional IDSs that may not be as effective against modern sophisticated attacks [9]. The necessity of IDSs is “low false-positive rate and high true positive rate”. Intruders to a network can be classified into two types: external intruder and internal intruder. (1) External intruder: An outsider using diverse means of attacks to arrive at the network. (2) Internal intruder: A compromised node that used to be an associate of the network. IDS can detect both external and internal intruders, but internal intruders are harder to detect. This is due to that internal intruders have the necessary keying resources to counteract any protection taken by the authentication mechanisms. Intrusion can be of any type such as attempted break, Masquerade, Penetration, Leakage, DoS and Malicious use. IDSs may provide partial detection solutions to those attacks. The perfect IDS that would able to detect all of the intrusions listed above [4], [5], [6]. Based on deployment, the IDS can be categorized into two types: host-based intrusion detection system (HIDS) and network-based intrusion detection system (NIDS). HIDS is disturbed among the measures on the host with the purpose of them are working and they are able of detecting intrusions like changes to important system files on the host, numerous breakdown access attempts to the host, abnormal method memory allocations, unusual CPU activity or I/O activity. By monitoring the real-time scheme usage of the host or by investigative log files on the host HIDS achieves this. NIDS can examine a whole packet; payload inside the packet, IP addresses or ports either passively or actively by listens to the network transmissions. Based on detection methodologies, IDS can be classified as anomaly-based detection, misuse based detection and specification-based detection: 1) Misuse detection: In this case, the patterns have to be defined and given to the system and act or behaviour of nodes is compared with well-known attack patterns. The disadvantages are that this method requests knowledge to build attack patterns and they are not able to sense novel attacks. The drawback of this approach is significantly to reduce the efficiency in terms of system management, as the administrator of the network always has to offer IDS agents with a current database. 2) Anomaly detection: The approach prime describes the real features of a 'normal behaviour', which are renowned by using automated training and this method does not search for exact attack patterns, but in its place, it checks whether the behaviour of the nodes can be measured as normal or anomalous. Then it flags any behaviour that diverges from these behaviours as intrusions. The IDS would have high confidence to decide that the node is malicious if a sensor node does not act according to the distinct specification of a particular protocol; the wrong decisions made by IDS in terms of false-positive and false-negative alarms influence the accuracy of detection. The disadvantage of this method is that the system can illustrate valid but hidden behaviour, which could show the way to a significant false alarm rate. 3) Specification-based detection: It is paying attention to discovering deviations from normal behaviours that are defined neither by machine learning techniques nor by training data as this method combines the aims of misuse and anomaly detection mechanisms. The specifications that describe what can be considered as normal behaviour are defined manually and any action is monitored concerning these specifications. The drawback of this approach is the manual development of all specifications, which is a time-consuming procedure for human beings and it cannot detect malicious behaviours that do not violate defined specifications of the IDS protocol. Sometimes misuse and anomaly-based detection techniques can be used that give birth to hybrid detection mechanisms [1].

The main aim of this paper is to provide a comparative study and performance analysis using different machine learning based techniques for intrusion detection for networks. We analyse Machine learning techniques are Random Forest (RF), K-Nearest Neighbours (KNN), Decision Tree (DT), Naïve Bayes(NB), and Support Vector Machine (SVM). For analysis purposes, an NSL-KDD [7], [10] is used as a dataset and Python is used as a programming language.

The rest of this paper is organized as follows: Section 2 presents the literature study of recent techniques employed for network IDS. Section 3 discusses machine learning approaches, Section 4 discusses the particulars of data set, implementation, and experimental results. Finally, the concluding remarks of the study are provided in Section 5.

II. Literature survey of machine learning approaches-based intrusion detection systems

This section describes the Machine learning approaches-based intrusion detection systems.

Myint and Meesad have proposed a classifier known as an Incremental Learning Algorithm based on SVM [11]. In this, a prediction is done by using SVM and is going to reduce steps required for calculation and complexity of the algorithm, error set, and time is saved for repeatedly training the dataset. In this approach, the author used the KDD Cup99 dataset to evaluate the performance of the system. The proposed system can predict 41 features of the incoming dataset.

Nabila Farnaaz and M. A. Jabbar have proposed a model using a Random Forest classifier for intrusion detection [12]. In this approach, the author considered RF as an ensemble classifier and the model gives a better performance as compared to other traditional classifiers for classification of attacks. To evaluate the performance of the model, the author used the NSL-KDD dataset, and the proposed model is efficient with a low false alarm rate and high detection rate.

Majjed et al. have proposed an effective deep learning approach STL-IDS supported the self-taught learning framework [13]. For feature learning as well as to reduce the dimension, the proposed system can be used. In this approach, to achieve a greater prediction accuracy of SVM the training as well as testing time is reduced. The proposed approach provides an improvement in network intrusion detection.

Sandhya Peddabachigari et al. have evaluated the decision tree for intrusion detection [14]. Intrusion detection with the decision was tested with the 1998 DARPA dataset, and the system gives better performance as compared to traditional models in terms of accuracy. Again the results show that the training time and testing time is better as compared to Support Vector Machine.

Mrutyunjaya Panda and Manas Ranjan Patra have proposed a framework of NIDS based on Naïve Bayes [15]. For implementation KDD Cup 99 is used as a dataset and from the results, it is determined that the planned system offers higher performance in terms of false-positive rate, procedure time and price.

Wenchao Li et al. have proposed a new intrusion detection system based on the K-nearest classification algorithm in WSN [16]. The proposed system is used to separate normal and abnormal nodes by monitoring the unusual behaviour. In this, the parameter selection and an error rate of the intrusion detection system are analysed. The proposed model gives better efficiency with a high detection rate and speed.

Michael Riecker et al. [19] propose a lightweight, energy-efficient system, which makes use of mobile agents to detect intrusions based on the energy utilization of the sensor nodes as a metric. A linear regression model is applied to predict the energy consumption. Authors evaluate the proposed detection algorithm about detection accuracy in a scenario with flooding and a black hole attack. They also study the influence of the history size and the walking strategies on the detection time. They neither require nodes to monitor their environment and collaborate, nor do they need to transfer audit data to a central point. Instead, use a mobile agent that collects energy readings and raises an alert if sudden changes occur. The feasibility of mobile agents used for intrusion detection in wireless sensor networks has been verified. The authors further showed that energy consumption is a suitable metric to detect denial-of-service attacks. In simulations, they evaluated their proposed method for intrusion detection and were able to achieve high detection accuracy while maintaining a low false-positive rate.

Mohammad Wazid et al. [20] proposed a robust and efficient secure intrusion detection approach which uses the K-means clustering to extend the lifetime of a Wireless Sensor Networks (WSN). Authors proposed a new intrusion detection technique for a hybrid anomaly; K-means built patterns of attacks automatically over training data for the detection purpose. After that intrusion is detected by matching network activities against the detection patterns. The authors assess the approach over a WSN dataset that is created using the Opnet modeler, which contains a range of attributes, such as end- to- end delay, traffic sent and traffic received. The training dataset contains the normal values of the network parameters. The testing dataset is created in an actual working model consists of normal and abnormal values of the network parameters. Authors claim that the proposed scheme achieves a 98.6 % detection rate and 1.2 % false-positive rate, which is better than the existing related schemes and the proposed technique can detect two types of malicious nodes: black hole and misdirection nodes.

Yassine Maleh et al. [22] propose a hybrid, lightweight intrusion detection system for sensor networks. The proposed Hybrid intrusion detection system (HIDS) takes advantage of cluster-based architecture to reduce energy consumption and this model uses anomaly detection based on the support vector machine (SVM) algorithm and a set of signature rules to detect malicious behaviours and provide global lightweight IDS. The detection approach is integrated into a cluster-based topology to increase the network lifetime. This is achieved by designating one known node as a leader of the group (cluster-head) that forwards nodes packets (data aggregated) to the base station (BS) instead of sending their (nodes) collected data to a remote location (base station). Cluster head acts as a local base station sensor, and then clusters elect themselves to be a CH at any given time with a certain probability. They propose a cluster-based architecture that divides the array of sensors into a plurality of groups, each of which comprises a cluster-head (CH). In this architecture, every node belongs to only one of the clusters which are distributed geographically across the whole network. The cluster head is used to reduce network energy consumption and to increase its lifetime. Simulation results show that the proposed model can detect abnormal events efficiently and has a high detection rate with a lower false alarm. The combination of anomaly detection based on SVM and detection based on attack signatures allows the intrusion detection model to achieve a high rate of intrusion detection (almost 98%) with a number very reduces false alarms (near 2%). The performance of the proposed intrusion detection model is evaluated using the KDDcup'99 database.

HichemSedjelmaci et al. [23] propose a hybrid intrusion detection system for clustered WSN. Their intrusion framework uses a combination of the Anomaly Detection based on the support vector machine (SVM) and Misuse Detection. Anomaly detection uses a distributed learning algorithm for the training of an SVM to solve the two-class problem (distinguish between normal and anomalous activities). Also, they use a hierarchical topology that divides the sensor network into clusters, each one having a cluster head (CH). The objective of this architecture is to save the energy that allows the network lifetime prolongation. In experiments, they used the KDDcup’99 dataset as the sample to verify the efficiency of the distributed anomaly detection algorithm and valid it by comparing it with a centralized SVM-based classifier, which achieves a high level of accurate detection. The proposed distributed learning algorithm for the training of SVM in WSN reaches high accuracy for detecting the normal and anomalous behaviour (accuracy rate over 98%). Also, a combination between the SVM classifier and Signature Based Detection achieve a high detection rate with a low false-positive rate and their approach reduces energy consumption.

III. Machine learning approaches [24]

3.1 Naïve Bayes algorithm

The Naïve Bayes algorithm is used to perform classification, which is based on Bayes theorem. This algorithm works on assumption that all input attributes are conditionally independent.

The steps of Naïve Bayes algorithm are as follows:

Step 1: Given a training set S, Calculate the probability of each class $p(v_j)$.

Step 2: Given a training set S, For each attribute value a_i of each attribute a, calculate conditional probability $p(a_{i|}v_j)$.

Step 3: Given an unknown instance X’, Classify X’ according to the best probability.

3.2 Decision Tree algorithm

Decision tree learning is a method for approximating discrete-valued target functions, in which the learned function is represented by a decision tree.

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example. This process is then repeated for the sub tree rooted at the new node.

The working steps of Decision Tree algorithm are given below.

Step 1: First, To place the best attribute from the dataset at the root of the tree some mathematical measure like information gain is used.

Step 2: Second, Divide train dataset into subsets. While dividing, we should consider each subset should contain data with the same value for an attribute.

Step 3: Lastly, just repeat Sep 1 and Step 2 on each subset until we find leaf nodes in all the branches of the tree.

3.3 Random forests algorithm

Random forests are an ensemble learning method for classification or regression that operate by constructing a multiple decision trees by picking “K” number of data points point from the dataset and then merges them together to get a more accurate and stable prediction. For each “K” data point’s decision tree, we have many predictions and then we take the average of all the predictions.

The steps for Random Forest algorithm are as follows:

Step 1: Select randomly “i” features from the entire “j” features with one condition i << j.

Step 2: Using the concept of best split point, calculate node “n” from the “i” features.

Step 3: Again using the concept of best split, we need to split node “n” into daughter node.

Step 4: Repeat Step 1–Step 3 until “1” number of node has been reached.

Step 5: Build forest by repeating Step 1–Step 4 for “k” number of times to create “k” number of trees.

Step 6: To predict target, take test features and use the rules of each randomly created decision tree and store the predicted target.

Step 7: Then simply find out votes for each predicted target.

Step 8: At last, consider the high voted prediction target as a final prediction.

3.4 K-Nearest Neighbour algorithm

K-nearest neighbours (KNN) algorithm classifies new objects based on a similarity measures. To measure similarity between different objects mathematical measure Euclidean Distance is used. In KNN algorithm, for each test data point, we would be looking at the K-nearest training data points and take the most frequently occurring classes and assign that class to the test data. Therefore, K represents the number of training data points lying in proximity to the test data point which we are going to use to find the class.

The steps of K-Nearest Neighbours algorithm are given below.

Step 1: Decide value of K.

Step 2: Calculate distance between query instance and all the training samples.

Step 3: Sort the distance in ascending order and confirm nearest neighbours supported the Kth minimum distance.

Step 4: On the basis of majority of class of nearest neighbours, assign the prediction value of the query instance.

3.5 Support Vector Machine (SVM)

The SVM classifier is used for classification and regression. In SVM, data is spat into the data point by using hyper plane and it is used to determine the class of data point [28]. The distance from the boundary to the nearest data point is called as margin and the data point that lies closest to the classification boundary is called a support vector. When we deal with SVM, then we have to assume two things: 1) The margin should be as large as possible, and 2) The support vectors are the most useful data points because they are the ones most likely to be incorrectly classified.

The working steps for SVM are as follows:

Step 1: Define optimal hyper plane: maximize margin.

Step 2: Extend the definition mentioned in Step 1 for nonlinearly separable problems: have a penalty term for misclassifications.

Step 3: Map data to high-dimensional space where it is easier to classify with linear decision surfaces: reformulate problem so that data is mapped implicitly to this space.

IV. Experimentation

We use NSL-KDD to test the performance of Machine learning based Intrusion detection approaches. Tables 1 summarize a collection of downloadable files at the disposal for the researchers. The experiment is performed on Google Colaboratory under python 3 using TensorFlow and Graphics Processing Unit (GPU).

Table 1: List of nsl-kdd dataset files and their description

Sr. No.	Name of the file	Description
1	KDDTrain+.ARFF	The full NSL-KDD train set with binary labels in ARFF format
2	KDDTrain+.TXT	The full NSL-KDD train set including attack-type labels and difficulty level in CSV format
3	KDDTrain+_20Percent.ARFF	A 20% subset of the KDDTrain+.arff file
4	KDDTrain+_20Percent.TXT	A 20% subset of the KDDTrain+.txt file
5	KDDTest+.ARFF	The full NSL-KDD test set with binary labels in ARFF format
6	KDDTest+.TXT	The full NSL-KDD test set including attack-type labels and difficulty level in CSV format
7	KDDTest-21.ARFF	A subset of the KDDTest+.arff file which does not include records with difficulty level of 21 out of 21
8	KDDTest-21.TXT	A subset of the KDDTest+.txt file which does not include records with difficulty level of 21 out of 21

4.1 Dataset Description

NSL-KDD dataset is proposed by Tavallaee et al. [27] and is recommended to solve some of the inherent problems of the KDD'99 dataset. The inherent drawbacks in the KDD cup 99 datasets have been revealed by various statistical analyses that have affected the detection accuracy of many IDS modelled by researchers. NSL-KDD data set [28] is a refined version of its predecessor. It contains essential records of the complete KDD data set. Compared to the original KDD dataset, the NSL-KDD dataset has the following improvements: (1)Redundant records are removed to enable the classifiers to produce an unbiased result, (2) Duplicate records are removed, (3) the number of selected records is organized as the percentage of records (e.g. DDTrain+_20Percent.ARFF), and (4) Sufficient number of records is available in the train and test data sets, which is reasonably rational and enables to execute experiments on the complete set, (5)The number of selected records from each difficult level group is inversely proportional to the percentage of records in the original KDD data set [29]. In each record 41 attributes are unfolding different features of the flow and The 42nd attribute is a label assigned to each either as an attack-type (DoS, Probe, R2L, and U2R) or as normal [27][30]. The specific types of attacks are classified into four major categories. Table 2 shows this detail.

Table 2: Mapping of attack class with attack type

Attack Class	Attack Type
DoS	Back, Land, Neptune, Pod, Smurf,Teardrop,Apache2, Udpstorm, Processtable, Worm
Probe	Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint
R2L	Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps

Table 3 shows the distribution of the normal and attack records available in the various NSL-KDD datasets. [29]

Table 3: Details of normal and attack data in different types of nsl-kdd data set

Dataset Type	Total No. of					
	Records	Normal Class	DoS Class	Probe Class	U2R Class	R2L Class
KDD Train+ 20%	25192	13449	9234	2289	11	209
		53.39%	36.65%	9.09%	0.04%	0.83%
KDD Train+	125973	67343	45927	11656	52	995
		53.46%	36.46%	9.25%	0.04%	0.79%
KDD Test+	22544	9711	7458	2421	200	2754
		43.08%	33.08%	10.74%	0.89%	12.22%

4.2 IDS methodology used in experimentation

The details of the IDS methodology used in experimentation are illustrated in Fig. 1. Specifically, the method consists of four stages: (1) datasets stage, (2) pre-processing stage, (3) training stage and (4) testing stage.

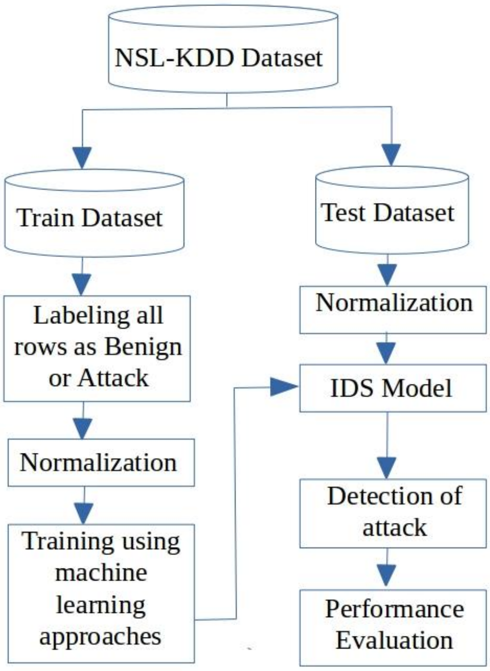

Fig. 1. Flowchart of the IDS methodology

4.3 Performance Metrics

We use the most important performance indicators, including, detection rate (DR), false alarm rate (FAR) and accuracy (ACC). We can calculate the performance metrics using the following

Accuracy (ACC): It is a metric that is used to indicate the proportion of correct classifications of the total records in the testing set.

Accuracy = (TP+ TN)/ (TP+ FN+ TN+ FP)

Precision (P): It is a metric that measures the actual performance within the required answer space, i.e., among the positions.

P =TP/(TP + FP)

Recall (R): It is the metric by which we measure how much of the predicted answers are discarded or for every correct label, how many other true labels have we discarded.

R =TP/(TP + FN)

F1 Score (F): It is the harmonic mean of the two matrices P and R.

F =(2 ∗ P ∗ R)/(P + R)

Where,

True positive (TP): It can be outlined as anomaly instances properly categorized as an anomaly.

False positive (FP): It can be outlined as normal situations wrongly categorized as an anomaly.

True negative (TN): It can be outlined as normal situations properly categorized as normal.

False negative (FN): It can be outlined as anomaly instances wrongly categorized as normal. [7]

4.4 Results and Discussion

Table 4: Comparison of machine learning based IDS

Algorithm	Accuracy (overall)	Precision		Recall		F1 Score	
		Attack	Normal	Attack	Normal	Attack	Normal
Decision Tree	80.57	0.97	0.70	0.68	0.97	0.80	0.81
K nearest neighbour	76.58	0.97	0.65	0.61	0.97	0.75	0.78
Random forest	77.64	0.97	0.66	0.63	0.97	0.76	0.79
Naive Bayes	45.03	0.94	0.44	0.04	1.00	0.07	0.61
Support Vector Machine	43.08	0.67	0.43	0.00	1.00	0.00	0.60

For comparison, five algorithms of machine learning were considered, namely Support Vector Machine, Naive Bayes, Random Forest, Decision Tree, and K nearest neighbor. For comparison purpose, precision, recall, and F1 score were considered, and their comparison results are shown in the Table 4, we can say that the accuracy of Support Vector Machine algorithm is lowest and accuracy of Decision Tree algorithm is highest.

V. Conclusion

This paper presents a comparative study and performance analysis on intrusion detection and prevention system for WSN using machine learning-based IDS. In this paper, the results of various machine learning techniques for attack detection are presented. Through the literature survey, we understand that there is a need to develop a scalable and attack resistance system for intrusion prevention using deep packet inspection in a WSN. A system is proposed to detect and prevent intrusion from using deep learning for the WSN.

References:

[1]. [1] Rakesh Sharma, and Vijay Anant Athavale, “A Survey of Intrusion Detection Techniques and Architectures in Wireless Sensor Networks”, Int. J. Advanced Networking and Applications, Volume: 10 Issue: 04 Pages: 3925-3937, 2019.

[2]. [2] M. Ngadi, A.H. Abdullah, and S. Mandala, “A survey on MANET intrusion detection”, International J. Computer Science and Security, volume 2, number 1, pages 1-11, 2008.

[3]. [3] Y. Zhang, W. Lee, and Y.A. Huang, “Intrusion detection techniques for mobile wireless networks”, J. Wireless Networks, vol. 9, num. 5, pp.545-556, 2003.

[4]. [4] T.S. Sobh, “Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art”, Elsevier J. Computer Standards and Interfaces, volume 28, number 6, pages 670-694, 2006.

[5]. [5] T. Anantvalee and J. Wu, “A survey on intrusion detection in mobile ad hoc networks”, Springer J. Wireless Network Security, pages 159-180, 2007.

[6]. [6] P. Albers, O. Camp, J. Percher, B. Jouga, L. M, and R. Puttini, “Security in Ad Hoc Networks: A General Intrusion Detection Architecture Enhancing Trust Based Approaches,” Proc. 1st International Workshop on Wireless Information Systems (WIS-2002), pp. 1-12, April 2002.

[7]. [7] Mohamed Amine Ferrag, LeandrosMaglaras, Sotiris Moschoyiannis, Helge Janicke, “Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study”, Journal of Information Security and Applications, 50 (2020) 102419.

[8]. [8] Dewa Z, Maglaras L A, “Data mining and intrusion detection systems”, Int. J. Adv. Comput. Sci. Appl. 2016; 7(1):62–71.

[9]. [9] Stewart B , Rosa L , Maglaras LA , Cruz TJ , Ferrag MA , Simões P , et al., “A novel intrusion detection mechanism for scada systems which automatically adapts to network topology changes”, EAI Endorsed Trans. Ind. Netw. Intell. Syst. 2017; 4(10):e4.

[10]. [10]Niyaz, Q., Sun, W., Javaid, A. Y., &Alam, M. (2015, December 03–05), “A deep learning approach for network intrusion detection system”, In BICT 2015, New York City, United States.

[11]. [11]Myint, H. O., &Meesad, P. (2009), “Incremental Learning Algorithm based on Support Vector

[12]. Machine with Mahalanobis distance (ISVMM) for Intrusion Prevention”, 978-1-4244-3388

[13]. 9/09/$25.00 ©2009 IEEE.

[14]. [12]Farnaaz, N., & Jabbar, M. A. (2016). Random forest modelling for network intrusion detection system. Procedia Computer Science, 89, 213–217 (Elsevier).

[15]. [13] Al-Qatf, M., Lasheng, Y., Alhabib, M., & Al-Sabahi, K. (2018), “Deep learning approach combining sparse auto encoder with SVM for network intrusion detection”, IEEE Access. https:// doi.org/10.1109/ACCESS.2018.2869577.

[16]. [14] Peddabachigari, S., Abraham, A., & Thomas, J. (2016), “Intrusion detection systems using decision trees and support vector machines”, International Journal of Advanced Networking and Applications, 07(04), 2828–2834. ISSN: 0975-0290.

[17]. [15] Panda, M., & Patra, M. R. (2007, December). “Network intrusion detection using Naïve Bayes”, IJCSNS International Journal of Computer Science and Network Security, 7(12).

[18]. [16] Li, W., Yi, P., Wu, Y., Pan, L., & Li, J. (2014), “A new intrusion detection system based on KNN classification algorithm in WSN”, Journal of Electrical and Computer Engineering, 2014 (Hindawi Publishing Corporation).

[19]. [17] Van, N. T., Thinh, T. N., &Sach, L. T. (2017), “An anomaly-based network intrusion detection system using deep learning”, In 2017 International Conference on System Science and Engineering (ICSSE).

[20]. [18]Yang, Y., Zheng, K., Wu, C., Niu, X., Yang, Y., “Building an Effective Intrusion Detection System Using the Modified Density Peak Clustering Algorithm and Deep Belief Networks”, Appl. Sci. 9, 238 (2019).

[21]. [19] Michael Riecker, Sebastian Biedermann, Rachid El Bansarkhani and Matthias Hollick, “Lightweight energy consumption-based intrusion detection system for wireless sensor networks”, International Journal of Information Security, vol. 14, no. 2, pp. 155-167, 2015.

[22]. [20] Mohammad Wazid and Ashok Kumar Das, “An Efficient Hybrid Anomaly Detection Scheme Using K- Means Clustering for Wireless Sensor Networks”, Wireless Personal Communications, vol. 90, no. 4, pp. 1971-2000, October 2016.

[23]. [21]Ahmed Saeed, Ali Ahmadinia, Abbas Javed and Hadi Larijani, “Random Neural Network based Intelligent Intrusion Detection for Wireless Sensor Networks”, In proceedings of International Conference on Computational Science, vol. 80, pp. 2372-2376, 2016.

[24]. [22]Yassine Maleh, AbdellahEzzati, Youssef Qasmaoui and Mohamed Mbida, “A Global Hybrid Intrusion Detection System for Wireless Sensor Networks”, The fifth International Symposium on Frontiers in Ambient and Mobile Systems, vol. 52, pp. 1047-1052, 2015.

[25]. [23]HichemSedjelmaci and Mohamed Feham, “Novel Hybrid Intrusion Detection System for Clustered Wireless Sensor Network”, International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.4 July 2011.

[26]. [24]P. R. Chandre, P. N. Mahalle, and G. R. Shinde, “Deep Learning and Machine Learning Techniques for Intrusion Detection and Prevention in Wireless Sensor Networks: Comparative Study and Performance Analysis”, Lecture Notes in Networks and Systems 82, https://doi.org/10.1007/978-981-13-9574-1_5

[27]. [25]Nicholas Lee, Shih Yin Ooi and Ying Han Pang, “ A Sequential Approach to Network Intrusion Detection”, Lecture Notes in Electrical Engineering 603, https://doi.org/10.1007/978-981-15-0058-9_2

[28]. [26]Kishor Kumar Gulla, P. Viswanath, Suresh BabuVeluru, and R. Raja Kumar, " Machine Learning Based Intrusion Detection Techniques", Handbook of Computer Networks and Cyber Security, https://doi.org/10.1007/978-3-030-22277-2_35

[29]. [27]Tavallaee M, Bagheri E, Lu W, Ghorbani A. A. "A detailed analysis of the kdd cup 99 data set", In: 2009 IEEE Symposium on Computational Intelligence for Security and Defence Applications. IEEE; 2009. p. 1–6.

[30]. [28]Nslkdd. https://www.unb.ca/cic/datasets/nsl.html

[31]. [29]L. Dhanabal, and Dr. S.P. Shantharajah, "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms", international Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2015.

[32]. [30]Sapna S. Kaushik, Dr. Prof. P. R. Deshmukh," Detection of Attacks in an Intrusion Detection System", International Journal of Computer Science and Information Technologies, Vol. 2 (3), 2011, 982-986