Journal of Theoretical and Applied Information Technology

15th November 2023. Vol.101. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PERFORMANCE ANALYSIS OF INTRUSION DETECTION APPROACHES IN AGRICULTURAL IOT NETWORKS: A MACHINE LEARNING BASED APPROACH

DR. S.B. DEOSARKAR[1], SANTOSH KONDE[2]

[1] (Professor) Departments of Electronics and Telecommunication Engineering, Dr. Babasaheb Ambedkar Technological University, Lonere Dist. Raigad (M.S.) India.

[2](Research Scholar) Department of Electronics and Telecommunication Engineering, Dr. Babasaheb Ambedkar Technological University, Lonere Dist. Raigad (M.S.) India

E-mail : [1]sbdeosarkar@dbatu.ac.in, [2]skonde76@gmail.com

ABSTRACT

In the last decade there is a tremendous growth in the field of IOT ,which is related with Agriculture. The continuous development in the Agriculture IOT, there is number of connected devices have large amount complex network data and huge amount of complexity. In the Agriculture IOT Network have connected the low power CPU and minimum memory capacity, Live streaming devices. These devices are critical decision dependent which are unable to run security software in the existing environment. These Agriculture IOT devices create the inherent risk in the Networks. The attackers have more concentrated on this Agriculture IOT devices. The number of possibility on the Network attacks are increasing due to these draw backs. The current intrusion detection system (IDS) is not working effectively. For the analyzing and dealing with improvement study in the field of IDS and prevention techniques will be identify to the normal and abnormal activities in the field of IOT Agriculture Networks. Thus there is a requirement to design the effective IDS using machine learning for the field of Agriculture IOT Networks.

In this paper we have represent the survey and comparative study with the analysis of the machine learning methods, to threat detection in the field of Agricultural IOT networks environment. Machine learning methods performed on NSL – KDD and BOT – IOT data set. In this method we analyze machine learning (ML) models that include support vector machine (SVM), decision tree(DT), Naïve Bays(NV), Random forest(RF), and K – nearest Neighbors (KNN). Also we used the most performance Indicators namely as accuracy, precision, recall and F1 score, to test the effectiveness of several methods.

Keywords: *Internet of things (IOT), Machine learning (ML), Intrusion detection system (IDS), Dataset, Algorithm.*

1. INTRODUCTION

IOT in Agriculture have number of connected devices are interconnected with Light processor, web services and network cards. These devices are interfacing with different Networks. It also connected with light memory capacity and low power sensors, devices can be used for collecting the Information. In the field of Agriculture, retail, healthcare and manufacturing industries are used devices to collect the information related with automatic patient monitoring, supply chaining warehouses as well as tracking the purchased items. The agricultural IOT is the advanced technologies in the current agricultural activities to improve the quality of the product and productivity of foods. The number of open IOT devices in the field, there are many new threat in the Agriculture networks. When the new technologies adopted by the public demand in the Agriculture field, attracts the interest of attackers to disturb the IOT networks by using complex hacking techniques such as botnet. This is a lack of standardization in IOT system as well as the 1ow power, cheap and light devices is the additional benefit for the attackers to deficit these systems [1,2,3,4].

Any kind of unnatural behavior on the Network or system will be considered as intrusion. An intrusion is defined as: “any set of actions that attempt to compromise the integrity, confidentiality, or availability of resource.” an IDS is a set of tools to facilitate distinguishing, evaluating and describing Intrusion. IDS are a preventative and defense system that can eliminate abnormal activities [5] [6]. IDS is considered as a precaution wall for the security purpose, To secure the system in the field of IOT Agricultural networks IDS can be deployed with security measures such as access control ,authentication mechanism and encryption technique. Using pattern of undefined traffic or threat, IDS can be differentiate normal and malicious actions [7].

According to Deva and maglaras [8] data mining, which is used to describe discovery knowledge can help improvement and deploy IDS with most accurate and robust behavior as compared to traditional IDS, which may not be as effective against modern sophisticated attacks [9].

A requirement for IDS is "a low false positive rate and a high true positive rate." In the field of Network intruders can be divided into two types as

1.1 External intruders.

1.2 Internal intruders

1.1 External intruders: an unknown who uses various attacks methods to enter the networks.

1.2 Internal intruders : It is a network partner which is adjustable Node. IDS can detect both external and internal intruders, but internal intruders are difficult to detect easily.

The internal attackers have the authentication key resources to enter in any protected field by mechanisms. Intrusions are any type, such as attempted break, masquerade, penetration, leakage, Dos and malicious use. These types of seen attacks are partial detection solution by the IDS. The perfect IDS that will be detect and the types mentioned attacks or also detects all the intrusion. The IDS can be divided into two types: Host Intrusion Detection system (HIDS) and network intrusion detection system (NIDS). HIDS is used between the Host to detect Intrusion likes changes occurred in the system file, numerous attempts to access the host, unidentified memory allocation methods, abnormal CPU activity or I/O activity. HIDS can identify these activities by monitoring the host real time traffic or supervising log files on the host [10-12].

NIDS are identified the entire packet, payload inside the packet, IP address, or porting by active or passive Listening Network traffic according to method use in detection, IDS can be classified into following types as :

1.3 Misuse base detection

1.4 Anomaly based detection

1.5 Specification based detection

1.3 Misuse base detection:

In this detection approaches the pattern are well defined and given to the system. All the nodes are compared to given defined patterns and attackers can be identified with based defined pattern. This method has disadvantages that it will be up to date database. The new attacks patterns can be create by knowledge and it will require more update knowledge of patterns.

1.4 Anomaly detection:

This method is benefitted to identify the abnormal or normal behavior of networks by giving the proper training. It will be identified the normal behavior by using automated training. IDS will have high confidence in deciding that the particular Node is malicious, if the sensor Node is not working by given specific particular protocol specification. Sometimes this detection system can be works as valid if the works as invalid. This is disadvantage of the system.

1.5 Specification based detection:

This system given attention to the detecting deviations regarding not defined by ML technique or training data. It is developed by manually and describes the pattern as normal behavior. If any action create against these specification will be tracked. The disadvantage of this approach is that manual development of all the specifications is a time consuming process and cannot detect malicious behavior that does not violate the defined specifications of the IDS [13].

The main objective of this paper is to provide a comparative study and performance analysis by using different machine learning (ML) based IDS for agriculture IOT Networks. The ML Technique we are using here as Random forest (RE), KNN, decision Tree (DT), Naive Bayes (NB), and SVM (support vector machine). For analysis purpose we can use the NSL – KDD [7], [14] and BOT – IOT dataset and the programming language, We can used here as Python. The total paper organized as section 2 presents the literatures survey of the current study techniques used for IDS. Section 3 discussed the ML approaches. Section 4 discussed the details about the datasets implementation and the experimental result. The concluding remarks of the study presented in the section 5.

2. LITERATURE SURVEY OF ML BASED IDS

Diro et al. [16] presented a distributed IDS based on deep learning (DL) for IOT networks. The author's proposed to deploy this system on a fog computing layer for hosting IDS. Three intrusion datasets namely NSL-KDD, ISCX, and KDD-CUP-99 were used in the performance evaluation, in which the results show up to 97% accuracy.

Muna et al. [17] presented anomaly detection system (ADS) for detecting threats in the Industrial IOT. The ADS used an unsupervised deep auto encoder algorithm for pattern classification. NSL-KDD and UNSW-NB15 datasets were used in the evaluation. The experimental results show 99% accuracy.

Vinayakumar et al. [18] presented IDS against botnet threats based on scanning DNS services in smart city IOT applications. The presented mechanism uses a two-layer environment to monitor DNS logs and search the domain name generated by the domain generation algorithm using DL algorithms to increase accuracy.

Latif et al. [19] presented IDS that uses a lightweight random neural network to detect threats in the Industrial IOT.Compared with traditional ML approaches such as SVM, ANN, and DT, the presented system shows better accuracy. An open-source dataset DS2OS used in experimentation.

Parra et al. [20] presented a distributed architecture using two DL approaches, namely a distributed CNN and an LSTM network. DCNN is used in the IOT micro security plug-in, while LSTM is used by the back-end server. The N-BaIoT dataset is used in the performance evaluation, where the results show an accuracy of 98% and 94.30% during the training phase and the testing phase.

Haddad Pajouh et al. [21] presented IDS using a recurrent neural network. The presented mechanism uses three stages; namely data collection, feature extraction and deep threat classifier. Experimental results demonstrate the highest accuracy of 98.18% in 10-fold cross-validation analysis and performance compared to conventional ML classifiers such as Naive Bayes, KNN, RF, and DT.

Koroniotis et al. [22] presented a network forensic scheme called PDF to detect and monitor threat patterns in IOT networks. The PDF scheme is based on three stages, namely data extraction, adaptation of DL parameters, and identification of anomalous incidents. In the second phase, a particle swarm optimization (PSO) algorithm is used, while in the third phase, a deep neural model is used. Experimental results show an accuracy of 99.9% compared to 93.2% with the DT and 72.7% with naive bays.

Selvakumar et al. [23] presented an IDS based on vector convolution DL approach. The authors also presented that the calculations be processed in fog nodes. Experiments performed on the BoT-IOT dataset show accuracy up to 99.92% to detect threats in the Internet-of-Medical-Things (IMOT).

Manimurugan et al. [24] presented IDS using a deep belief approach. The presented mechanism is evaluated using the CICIDS 2017 dataset, which shows an accuracy of 97.71% and 96.37% for the PortScan threat and the infiltration threat, respectively.

Popoola et al. [25] presented a hybrid IDS, called LAE-BLSTM, to detect botnets in IOT networks. The LAE-BLSTM mechanism uses a deep bidirectional long-short-term memory (BLSTM) and a long-term memory auto encoder (LAE). LAE is used for feature dimensionality reduction, while BLSTM is used to identify botnet threat traffic in IOT networks. The Bot-IOT dataset used in the performance evaluation, showing that the LAE-BLSTM mechanism achieved a data size reduction ratio of 91.89%.

Basati et al [26] presented IDS called deep feature extraction. This model is based on CNN. The authors focused mainly on those devices that have low computing power. The authors used the UNSW-NB15, CICIDS2017 and KDD-Cup99 datasets for their experiments. The model was tested for both binary and multiclass classification.

Rashid et al [27] presented a tree-based stacking ensemble approach for intrusion detection in IOT. Two intrusion datasets, NSL-KDD and UNSW-NB15, were used to evaluate the effectiveness of the presented model. The authors also improved efficiency by integrating feature selection strategies to identify the most important features.

Fatani et al [28] presented feature engineering for an IDS system taking advantage of the swarm intelligence (SI) approach. Four popular public datasets, CIC2017, NSL-KDD, BoT-IOT and KDD99, were used for testing the accuracy of the presented IDS approach.

Alkahtani et al [29] presented three advanced and widely used DL models for intrusion detection. The authors conducted experiments with LSTM, CNN, and a hybrid CNN–LSTM model. To evaluate these DL models, the authors used the IoTID20 dataset.

Keserwani et al [30] presented a method for extracting significant IOT network features for intrusion detection. The presented method consists of a combination of grey wolf optimization and PSO. The authors used the KDDCup99, NSL-KDD and CICIDS-2017 datasets.

Qaddoura et al [31] presented single-layer feed forward neural network to detect threat IOT networks. The authors used data reduction with clustering and SMOTE oversampling approach.

Saba et al [32] presented a two-stage hybrid approach for detecting malicious threats in IOT networks. A genetic algorithm as well as well-known ML approaches such as SVM, ensemble classifier, and DT were used to select relevant features.

Majjed et al. [33] presented DL approach STL-IDS. For dimensionality reduction, the presented system can be used. In this approach, both training and testing time are reduced to achieve higher prediction accuracy of SVM.

SandhyaPeddabachigari et al. [34] evaluated a DT for intrusion detection. DT based intrusion detection was tested on a 1998 DARPA dataset, and the system outperforms traditional models in terms of accuracy. Again, the results show that the training and testing times are better compared to the SVM.

Mrutyunjaya Panda et al. [35] presented a NIDS framework based on Naïve Bayes. For the implementation, KDD Cup 99 is used as the dataset and from the results it is found that the planned system offers higher performance in terms of false positive rate, procedure time and cost.

3. MACHINE LEARNING APPROACHS [36]

3.1 Naïve Byes:

This algorithm is based on the probabilistic algorithms. It works on the probability of all the features vectors and their outcome. This is worked on the previous event occurring probability called as Posterior probability. On the basis of event we can evaluate the probability of event by comparison method. This algorithm is based on the bayes theorem. It is used to perform the classification. This algorithms works on the assumption that all the input attributes are conditionally independent. The steps of these Algorithms are as below,

Step 1: Given a training set S, compute the probability of each class P (Vj)

Step 2: Given training sets S, calculate the conditional probability P (ai/vj) for each attribute value ai and for each attribute a.

Step 3: Given an unknown instance of x! Classify x! According to the best likelihood.

3.2. Decision tree:

This algorithm is well known as the supervised learning algorithm. It is useful to preset a visual representation of the model. It is a hierarchal model which represents the flowchart which has several nodes. These nodes act as tests on the attribute in the dataset with a branch which leads to either another node or a decision on the data to be classified.

DT is a method for approximating discrete objective functions in which learned function is represented by D.T. Decision tree classify the instances by sequence in the form of a tree from roots to its some leaf node that provides the instances classification every nodes in the tree represents a test for some attribute of the instances and each branch descending from the node corresponds to one of the possible value for that attribute. an instance is classified from starting of the root node of the tree, testing the attribute specified by that node and then moving down the branch of the tree represents the value of attribute in the given instance. This process will repeated for the sub tree rooted at the new node. The working steps of the decision tree algorithm are mentioned here as.

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

Step 2: Divide the data set into the surest. When partitioning we should consider whenever each subset should contain data with same attribute values.

Step 3: Repeat the step 1 and 2 on each subset until we find leaf nodes in all branches of the tree.

3.3. Random forest (RF):

Random Forest (RF) is a supervised learning algorithm that is seen to be an improvement on the D.T. model. This model has two key concepts. The first in which when the training the model, each tree is given a random assortment of the data which can result in some trees. The same data can be using in multiple times; the reason is to lower the variance of the model, which lowers the difference in the predicted result scores. In the second concept only using small subset of the features when spitting the nodes in the trees. It is done to prevent over fitting when the model uses the training data to inflate the prediction is used to determine the overall class of the data. This method is called boot strap aggregating.

This is a learning method for classification and regression, which works by building multiple DT by selecting ‘k’ number of data points from the dataset and then joining them together to get a more accurate and stable prediction for each ‘k’ data point DT, we have many predictions and that take the average of all the predictions.

The steps for the Random forest algorithm are as follows.

Step 1: Randomly select elements ‘i’ from the entire elements ‘j’ with one condition i << j.

Step 2: Using the concept of best splitting point, calculate node ‘n’ from element ‘i’

Step 3: Again using the concept of best split, we need to spilt node ‘n’ into child node.

Step 4: Repeat step 1-3 until you reach node number 1.

Step 5: Create a forest by repeating step 1to step 4 for ‘k’ times to create ‘k’ number of trees.

Step 6: To predict the target, run the test functions and use the rules of each randomly generated decision tree and save the predicated target.

Step 7: Then simply find out the nodes for each projected target.

Step 8: Finally consider the target of the high vote prediction as the final prediction.

3.4: K-Nearest Neighbor (KNN)

KNN is available as a supervised learning model that is considered to be one of the simplest ML models. KNN have no more knowledge training done about the concept with KNN, instead the training data are used when making the prediction to classify the data. KNN will be operated on the assumption that similar data points will group and finds the closest data points use the K value which can be set to any number.

This algorithm classifies the new objects based on the degree of similarity. The mathematical measure of Euclidean distance is used to measure the similarity between different object. In this algorithms we take test data point, we would look at the K – nearest training data point and take the most frequently occurring classes and assign that class to the test data. K represent the number of training data points lying close to the test data point that we will use to find the class.

The steps of the KNN algorithms are given below

Step 1: choose the value of K.

Step 2: Complete the distance between the query instance and all training instances.

Step 3: Sort the distance in ascending order and confirm that the nearest neighbors support the Kth minimum distance.

Step 4: Based on the majority class of nearest neighbors, assign a predication value to the query instance.

3.5: Support vector machines (SVM)

Support vector machine (SVM) is classify future predictions, separating the training data which is using the hyper plane so it is also called as supervised algorithms.. The hyper plane divide a dataset into two classes which are decision boundaries that help classify the data point .A hyper plane can be represented as line or plane in multi dimensional space it is separate the data depends on their specific class. It finding the maximum margin space between the support vectors. This is used for classification and regression. In SVM the data point are separated by using a hyper plane. The distance from the boundary to the nearest data point is called as edge ,and the data point that present closest to the classification boundary is called support vector when implementation of SVM ,we have to assume two points.

3.5.1 The margin should be as large as possible.

3.5.2 Support vectors are the most useful point because they are the most likely to be misclassified.

The implementation steps for SVM are as below,

Step 1: Define the optimal hyper plane; maximize the edge.

Step 2: Extend the definition given in step 1 for nonlinear

Separable problems; have a penalty term for misclassification.

Step 3: Map the data into a high dimensional space where it is easier to classify using linear decision surfaces; reformulate the problem so that the data is implicitly mapped into this space.

4. IMPLEMENTATION OF DATA SET

Test the performance of ML based IDS we used to the NSL-KDD and BOT- IOT dataset. The experiment and test can be performed on the Google colaboratory IDE with the python 3.10 languages by using tensor processing unit (TPU).

4.1 Description of dataset

4.1.1 NSL-KDD Dataset:

The NSL – KDD dataset is proposed by Tavalee et.al. [37] And it is recommended to solve some of the internal problems of the KDD 99 dataset. The researchers can resolved the Inherent drawbacks of the KDD 99 dataset were revised by various statically analyses so they are increasing the efficiency of the detection accuracy of many IDS developed by the various researchers. The NSL KDD dataset [38] is an improved version of its predecessor. It stores basic records of the complete KDD dataset .we compared to one original KDD dataset, The NSL-KDD dataset have new improvement as.

- It redundant records are removed so that classifiers can shows the unbiased result.
- All the duplicate records are removed completely.
- The number of selected records is produced as percentage of records (e.g. DD Train ± 20% .ARFF)
- Sufficient number of records is available in the train and test dataset which will be enough to allow, the experiment performed on the complete dataset.
- Number of selected records from each group of difficulty levels is inversely proportional to the percentage records in the original KDD dataset [39]. In the records, 41 attributes have different properties of the flow, and the 42nd attribute is a label assigned to each either an attack type (DOS, probe, R2L and U2R) or normal [37] [40]. Specific types of attacks are divided into four main categories Table 1 show detail as.

Table 1: Attack type and Attack class

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

4.1.2 BoT-IOT Dataset:

The Bot-IOT dataset developing in the University Of New South Wales (UNSW) and it is published on 16 October 2019, website of the

IEEE. The dataset consists of ten CSV files, containing records for the following attacks on IOT networks: (i) Data exfiltration, (ii) Denial of Service HTTP, (iii) Denial of Service TCP, (iv) UDP Denial of Service, (v) HTTP Distributed Denial of Service, (vi) TCP Distributed Denial of Service, (vii) Distributed Denial of Service UDP, (viii) Key logging, (ix) Operating system scanning, and (x) Scanning Services. UNSW developed simulating a realistic network which has real attack data and simulated attack data in the data set. There are 35 features in the dataset. [15] Shown in Table No. 2.

Table 2 : Dataset feature and descriptions

Features	Description
Mean	Average duration of aggregated records
Sum	Total duration of aggregated records
Min	Minimum duration of aggregated records
Max	Maximum duration of aggregated records
Spkts	Source to destination packet count
Dpkts	Destination to source packet count
Sbytes	Source to destination byte count
Dbytes	Destination to source byte count
Rate	Total packets per second in transaction
Srate	Source to destination packets per second
Drate	Destination to source packets per second
Stime	Record start time
Sport	Port that data is being sent from
Dport	Port that data is being received from
Pkts	Total number of packets transferred
Bytes	Total number of bytes transferred
Ltime	Record last time
Seq	Sequence number
Dur	Record total duration

4.2 IDS methodology used in experimentation.

The details of IDS methodology and in the experimentation are shown in figure 1 specifically the method consist of three phases (1) Dataset and pre – processing (2) Training (3) Testing.

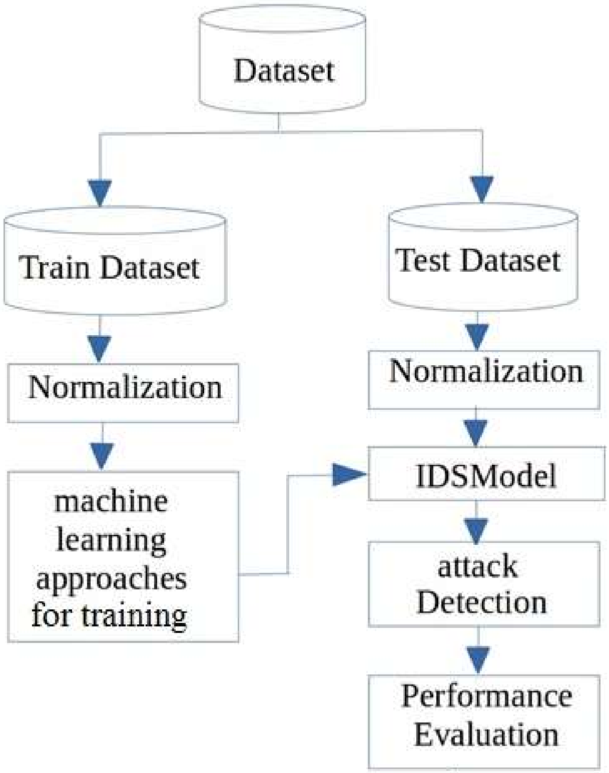

Fig.1. Flowchart of the IDS methodology used in experimentation

4.3 Performance metrics.

We can use key performance indicators including accuracy, precision, recall and F1 score.

4.3.1 Accuracy: It is a metric used to indicate the proportion of current classification on the total records in the test set. it can expressed as

$$\text{Accuracy} = \frac{\text{Number of current predications}}{\text{Total Number of Predications}}$$

This can be expanded as:

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

4.3.2 **Precision (P):** It is a metric that measures the actual performance within the desired response space between positions.

$$\text{Precesion} = \frac{\text{TP}}{\text{TP+FP}}$$

4.3.3 **Recall:** *(R)* it is a metric that measures how many predicated responses were discarded or for each cannot label how many other true labels we discarded.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

4.3.4 **F1 score (F):** F1 score is the weighted average of both precision and recall which produce a number between 0 and 1. It is the harmonic mean of two matrices precession and recall

$$\text{F1score} = \frac{2 \times (\text{record} \times \text{precession})}{\text{recall} + \text{preesion}}$$

As we can consider as F = (2 × P × R) / (P + R)

It is important noted that selection of F1 score or accuracy is dependent on the distributed data. Where

True positive (TP): cases of anomaly correctly categorized an anomaly.

False positive (FP): Case of a normal class miscategorised as a normally

True Negative: Cases of normal class and careful categorized as normal.

False Negative: (FN): Cases of abnormality misclassified as normal. [7]

4.4 Results and Discussion

Machine learning (ML) algorithm was considered as namely.

SVM (support vector machines), Naive Bays (NB), Random forest (RF),

Decision Tree (DT) also KNN. For the comparison we can use the accuracy, recall, precession and F1 score for performance metrics.

4.4.1 NSL-KDD dataset results

The comparison results for NSL-KDD dataset are shown in the Table 3, as shown below. It can be said that the accuracy of the Naive Bays algorithm is the lowest and the accuracy of the KNN algorithm is the highest.

4.4.2 BOT-IOT dataset results

The comparison results for BOT-IOT dataset are shown in the Table 4 in below for different types of attack categories.

5. CONCLUSIONS

This paper presents a comparative study and performance analysis of IDS for agriculture IOT networks based on ML. In this paper, the results of various ML techniques for attack detection are presented. Through a literature survey, we understood that there is a need to develop a scalable and attack- resistant system for intrusion detection using deep packet inspection in agriculture IOT networks.

Table 3: Comparison of machine learning based IDS for NSL-KDD dataset

Algorithm	Accuracy (overall)	Precision		Recall		F1 Score	
		Attack	Normal	Attack	Normal	Attack	Normal
KNN	77.63	0.63	0.97	0.96	0.66	0.76	0.79
SVM	76.55	0.66	0.9	0.9	0.67	0.76	0.77
Decision Tree	74.17	0.62	0.91	0.9	0.64	0.73	0.75
Random forest	73.24	0.55	0.98	0.97	0.62	0.7	0.76
Naive Bayes	51.17	0.90	0.00	0.54	0.02	0.68	0.00

Table 4: Comparison of machine learning based IDS for BOT-IOT Dataset

Algorithm	Accuracy (overall)	Precision		Recall		F1 Score	
		Attack	Normal	Attack	Normal	Attack	Normal
Data Ex filtration Attack Results							
Decision Tree	97.22	1.00	0.83	0.97	1.00	0.98	0.91
KNN	100	1.00	1.00	1.00	1.00	1.00	1.00
Random forest	91.66	0.96	0.75	0.93	0.86	0.95	0.80
Naive Bayes	94.44	0.93	1.00	1.00	0.80	0.96	0.89
SVM	100	1.00	1.00	1.00	1.00	1.00	1.00
Service Scan Attack Results							
Decision Tree	99.99	1.00	0.98	1.00	0.98	1.00	0.98
KNN	99.98	1.00	0.96	1.00	0.98	1.00	0.97
Random forest	99.99	0.99	1.00	1.00	0.99	1.00	0.99
Naive Bayes	97.98	0.98	1.00	1.00	0.09	0.99	0.16
Operating System Scan Attack Results							
Decision Tree	99.99	1.00	1.00	1.00	1.00	1.00	1.00
KNN	99.98	1.00	0.99	1.00	1.00	1.00	0.99
Random forest	99.98	1.00	1.00	1.00	1.00	1.00	1.00
Naïve Bayes	99.82	1.00	1.00	1.00	0.86	1.00	0.93
SVM	99.98	1.00	0.99	1.00	1.00	1.00	0.99

Key logging Attack Results							
Decision Tree	98.77	1.00	0.92	0.99	0.98	0.99	0.95
KNN	99.51	1.00	0.94	0.99	1.00	1.00	0.97
Random forest	99.75	1.00	0.98	1.00	1.00	1.00	0.99
Naive Bayes	98.53	1.00	0.88	0.99	0.97	0.99	0.92
SVM	99.02	1.00	0.90	0.99	1.00	0.99	0.95
Denial of Service HTTP Attack Results							
Algorithm	**Accuracy (overall)**	**Precision**		**Recall**		**F1 Score**	
		Attack	**Normal**	**Attack**	**Normal**	**Attack**	**Normal**
Decision Tree	100	1.00	1.00	1.00	1.00	1.00	1.00
KNN	99.95	1.00	1.00	1.00	0.79	1.00	0.88
Random forest	99.98	1.00	1.00	1.00	0.93	1.00	0.96
Naive Bayes	99.86	1.00	1.00	1.00	0.64	1.00	0.78
Denial of Service TCP Attack Results							
Decision Tree	99.99	1.00	0.99	1.00	0.98	1.00	0.98
KNN	99.99	1.00	0.98	1.00	1.00	1.00	0.99
Random forest	99.99	1.00	0.99	1.00	0.98	1.00	0.99
Naive Bayes	99.93	1.00	0.99	1.00	0.07	1.00	0.14
SVM	99.99	1.00	0.96	1.00	0.99	1.00	0.98
Denial of Service UDP Attack Results							
Decision Tree	99.99	1.00	0.96	1.00	0.98	1.00	0.97
KNN	99.99	1.00	0.90	1.00	1.00	1.00	0.95
Random forest	99.99	1.00	0.98	1.00	0.99	1.00	0.99
Naive Bayes	99.99	1.00	1.00	1.00	0.59	1.00	0.74
SVM	99.99	1.00	0.94	1.00	1.00	1.00	0.97

Distributed Denial of Service HTTP Attack Results							
Decision Tree	99.95	1.00	0.87	1.00	1.00	1.00	0.93
KNN	99.95	1.00	0.87	1.00	1.00	1.00	0.93
Random forest	99.86	1.00	1.00	1.00	0.95	1.00	0.98
Naive Bayes	98.36	0.98	1.00	1.00	0.13	0.99	0.23
SVM	99.97	1.00	0.87	1.00	1.00	1.00	0.93
Distributed Denial of Service TCP Attack Results							
Decision Tree	99.99	1.00	0.98	1.00	0.99	1.00	0.98
KNN	100	1.00	1.00	1.00	1.00	1.00	1.00
Random forest	99.99	1.00	0.99	1.00	1.00	1.00	1.00
Naive Bayes	99.99	1.00	1.00	1.00	0.53	1.00	0.70
SVM	100	1.00	1.00	1.00	1.00	1.00	1.00

Algorithm	**Accuracy (overall)**	**Precision**		**Recall**		**F1 Score**	
		Normal	**Attack**	**Normal**	**Normal**	**Attack**	**Normal**
Distributed Denial of Service UDP Attack Results							
Decision Tree	99.99	1.00	0.98	1.00	0.99	1.00	0.99
KNN	99.99	1.00	0.98	1.00	0.99	1.00	0.99
Random forest	99.99	1.00	0.99	1.00	0.99	1.00	0.99
Naive Bayes	99.99	1.00	1.00	1.00	0.60	1.00	0.75
SVM	99.99	1.00	0.95	1.00	1.00	1.00	0.98

REFERENCES

[1] Mohamed Amine Ferrag, Lei Shu, HamoudaDjallel and Kim-Kwang Raymond Choo, "Deep Learning-Based Intrusion Detection for Distributed Denial of Service Attack in Agriculture 4.0", Electronics 2021, 10, 1257. https://doi.org/10.3390/electronics10111257

[2] Safi Ullah, Jawad Ahmad, Muazzam A. Khan, Eman H. Alkhammash, MyriamHadjouni,

[3] YazeedYasinGhadi, Faisal Saeed and NikolaosPitropakis, "A New Intrusion Detection System for the Internet of Things via Deep Convolutional Neural Network and Feature Engineering", Sensors 2022, 22, 3607. https://doi.org/10.3390/s22103607.

[4] AnsamKhraisat and AmmarAlazab, "A critical review of intrusion detection systems in the

internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges", Khraisat and AlazabCybersecurity (2021) 4:18

[5] Eric Gyamfi and AncaJurcut, "Intrusion Detection in Internet of Things Systems: A Review on Design Approaches Leveraging Multi-Access Edge Computing, Machine Learning, and Datasets", Sensors 2022, 22, 3744. https://doi.org/10.3390/s22103744

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

[7] Y. Zhang, W. Lee and Y.A. Huang, "Intrusion detection techniques for mobile wireless networks", Journal on Wireless Networks, vol. 9, num. 5, pp.545-556, 2003.

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

[8] Dewa Z and Maglaras L A, "Data mining and intrusion detection systems", International Journal on Advanced Computer Science Applications 2016; 7(1):62–71.

[9] Pandey, V.C., Peddoju, S.K. &Deshpande, P.S. A statistical and distributed packet filter against DDoS attacks in Cloud environment. Sadhana, Vol. 43(32). (2018). https://doi.org/10.1007/s12046-018-0800-7

[10] Deshpande, P., Sharma, S.C., Peddoju, S.K. et al. HIDS: A host based intrusion detection system for cloud computing environment. Int J SystAssurEngManag 9, 567–576 (2018). https://doi.org/10.1007/s13198-014-0277-7

[11] Deshpande, P., Sharma, S.C., Peddoju, S.K. et al. Security and service assurance issues in Cloud environment. Int J SystAssurEngManag 9, 194–207 (2018). https://doi.org/10.1007/s13198-016-0525-0

[12] PrachiDeshpande; S. C. Sharma; P. SateeshKumar,"Security threats in cloud computing", International Conference on Computing, Communication &Automation,May 2015. doi:10.1109/CCAA.2015.7148450

[13] Rakesh Sharma and Vijay AnantAthavale, "A Survey of Intrusion Detection Techniques and Architectures in Wireless Sensor Networks", International Journal on Advanced Networking and Applications, Volume: 10 Issue: 04 Pages: 3925-3937, 2019.

[14] Niyaz Q, Sun W, Javaid A Y &Alam M, "A deep learning approach for network intrusion detection system", In BICT 2015, New York City, United States.

[15] NickolaosKoroniotis, NourMoustafa , Elena Sitnikova, Benjamin Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset", Future Generation Computer Systems 100 (2019) 779–796.

[16] Diro A A and Chilamkurti N, "Distributed attack detection scheme using deep learning approach for Internet of Things", Future Generation Computer Systems, 2018, 82, 761–768.

[17] Muna A H, Moustafa N and Sitnikova E, "Identification of malicious activities in industrial internet of things based on deep learning models", Journal of Information Security and Applications, 2018, 41, 1–11.

[18] Vinayakumar R, Alazab M, Srinivasan S, Pham QV, Padannayil SK and Simran K A, "visualized botnet detection system based deep learning for the Internet of Things networks of smart cities", IEEE Transactions on Industry Applications, 2020, 56, 4436–4456.

[19] Latif S, Zou Z, Idrees Z and Ahmad J A, "Novel Attack Detection Scheme for the Industrial Internet of Things Using a Lightweight Random Neural Network", IEEE Access, 2020, 8, 89337–89350.

[20] Parra G D L T, Rad P, Choo K K R and Beebe N, "Detecting Internet of Things attacks using distributed deep learning", Journal of Network and Computer Applications, 2020, 163, 102662.

[21] HaddadPajouh H, Dehghantanha A, Khayami R and Choo KKR, "A deep recurrent neural network based approach for internet of things malware threat hunting", Future Generation Computer Systems, 2018, 85, 88–96.

[22] Koroniotis N, Moustafa N and Sitnikova E, "A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework", Future Generation Computer Systems, 2020, 110, 91–106.

[23] BhuvaneswariAmma N G and Selvakumar S, "Anomaly detection framework for Internet of things traffic using vector convolutional deep learning approach in fog environment" Future

Generation Computer Systems, 2020, 113, 255–265.

[24] Manimurugan S, Al-Mutairi S, Aborokbah M M, Chilamkurti N, Ganesan S and Patan R, “Effective Attack Detection in Internet of Medical Things Smart Environment Using a Deep Belief Neural Network”, IEEE Access 2020, 8, 77396–77404.

[25] Popoola S I, Adebisi B, Hammoudeh M, Gui G and Gacanin H, “Hybrid Deep Learning for Botnet Attack Detection in the Internet of Things Networks”, IEEE Internet Things,2021, 8, 4944–4956.

[26] Basati A and Faghih M M, “DFE: Efficient IoT network intrusion detection using deep feature extraction”, Neural ComputAppl 2022,1–21.

[27] Rashid M, Kamruzzaman J, Imam T, Wibowo S, Gordon S, “A tree-based stacking ensemble technique with feature selection for network intrusion detection”, ApplIntell 2022, 1–14.

[28] Fatani A, Dahou A, Al-Qaness M A, Lu S, AbdElaziz M, “Advanced Feature Extraction and Selection Approach Using Deep Learning and Aquila Optimizer for IoT Intrusion Detection System”, Sensors 2022, 22, 140.

[29] Alkahtani H and Aldhyani T H, “Intrusion detection system to advance internet of things infrastructure-based deep learning algorithms”, Complexity 2021, 2021, 5579851.

[30] Keserwani P K, Govil M C, Pilli E S and Govil P, “A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model”, J ReliabIntell Environ 2021, 7, 3–21.

[31] Qaddoura R, Al-Zoubi A, Almomani I and Faris H, “A multi-stage classification approach for iot intrusion detection based on clustering with oversampling”, ApplSci 2021, 11, 3022.

[32] Saba T, Sadad T, Rehman A, Mehmood Z and Javaid Q, “Intrusion detection system through advance machine learning for the internet of things networks”, IT Prof 2021, 23, 58–64.

[33] 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

[34] 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.

[35] Mahbub M, Gazi M S A, Provar S A A, Islam M S, “Multi-Access Edge Computing-Aware Internet of Things: MEC-IoT”, In Proceedings of the 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), London, UK, 19–20, August 2020; pp. 1–6.

[36] 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

[37] 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.

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

[39] 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.

[40] 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