2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC) | 978-1-6654-7971-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICESC54411.2022.9885569

Network Security with Network Intrusion Detection System using Machine Learning Deployed in a Cloud Infrastructure

C.Kaushik
Assistant Professor, ECE Dept.
VNRVJIET
Hyderabad, India
kaushik_c@vnrvjiet.in

T.Ram
UG Student, ECE Dept.
VNRVJIET
Hyderabad, India
ramterli03@gmail.com

Ritvik.C
UG Student, ECE Dept.
VNRVJIET
Hyderabad, India
ritvikchalla@gmail.com

T.Lakshman
UG Student, ECE Dept.
VNRVJIET
Hyderabad, India
terlilakshman@gmail.com

***Abstract* - A software/device which monitors the network or the internet for any cyber-attack or malicious activity is called Network Intrusion Detection System (NIDS). The conventional detection systems are limited to slow detection and have to be monitored by a network engineer. This research study has attempted to improve the conventional detection systems by including powerful machine learning algorithm to the software thereby increasing its accuracy and preventing the need of a network engineer. The proposed study has used NSLKDD dataset, which has many independent features for different cyber-attacks. The data is pre-processed in order to make the data fit for the algorithm. The most important features are selected using feature selection. The model is trained using different machine learning algorithms. The accuracies of these algorithms are evaluated, and the validation metrics are calculated for the models. The trained model with best accuracy is containerized using Docker to make it machine/operating-system independent. This containerized model is stored in Docker repository. It is deployed into a cloud service provider which has many machine learning dependencies assisting the detection system. In the cloud, a virtual machine is created with required memory and processing unit. The detection system which is in Docker repository as a container is deployed into the cloud. After successful deployment, the user is provided with an Application Programming Interface (API). The user can connect to the internet and use the Network Intrusion Detection System via API to detect any attacks present in the network and secure the devices/software in the network.**

Keywords - Network Security, Intrusion Detection System, Containerization, Virtual Machine, Cloud Service, Deployment, Machine Learning Application Programming Interface.

I. INTRODUCTION

In modern digital world of internet, majority of the electronic devices are connected to the internet. They have existence in the cyber world. Lots of information is shared across the devices connected to the internet which is used for societal development. But this huge network is susceptible to cyber threats like stealing confidential data. In order to counter these threats, intrusion detection systems are developed which help the organizations or the end user in detecting malicious activity or cyber-attacks before these attacks make significant damage. A Network IDS is a software or a device which monitors the network or the internet for any cyber-attacks or malicious activity [3]. It is placed at a particular point in the network to analyse traffic in the network and alerts the nodes/devices residing in the network in case of an attack so that appropriate action can be taken before the damage becomes irreversible and cause huge loss [2],[3].

Network IDS monitors outgoing and incoming traffic which helps it to detect any suspicious activity. It observes the data packets in the network and compares them with a set of attack specific data. If there is any abnormality in the data, then the data is analysed for the type of attack and alerts the device administrator [2]. NIDS uses anomaly-based detection using machine learning. The system is trained with the help of machine learning algorithm and it is used to create a machine learning model and incoming data is analysed by that model and it is declared suspicious based on the trained weights of model.

The objective of this paper is to develop a Network IDS powered with machine learning features deployed in cloud to detect suspicious activity or cyber-attacks in the network and help the user/organization protect the device/software. By containerizing the model, it has been made machine independent by deploying cloud in order to provide the user with an API service [5]. NIDS powered with ML algorithm assists the end user to detect malicious activity in the network accurately and the containerized cloud-based API service makes it cost efficient and provides high availability [6].

II. ARCHITECTURE/DESIGN

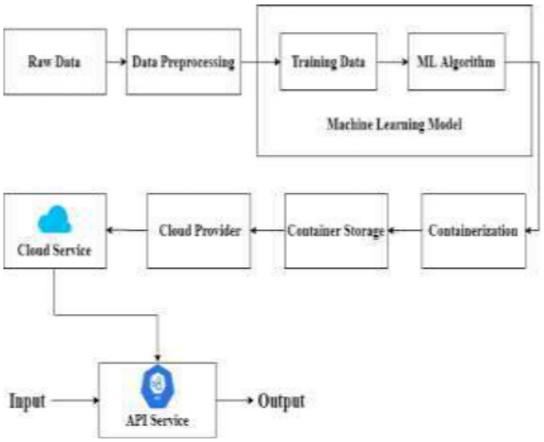


Figure 1. Block Diagram

III. INSTRUCTION DETECTION SYSTEM

IDS is an acronym for Intrusion Detection System which is a device/software which is capable of detecting and alerting the system administrator in case of threat and helps guide the organization or individual user in securing confidential information in the public network [2]. IDS detects and alert the system admin when initial input is drawn from conventional detection systems and it is fed to the ML Model as features to predict the type of cyber-attack and perform the next steps by eliminating the need of network engineer. Firewalls are useful in preventing unauthorized access to the systems but has the limitation of not being able to monitor and analyse network traffic where there is a high probability of cyber threat. Random Forest stands out when compared to others in achieving better accuracy even at peak network traffic. Firewall and authentication systems are important, but their use is limited to the entry point in the network. The threats in the network are not taken care by firewall systems. So, in addition to firewall systems, an organization's individual user's security must include Intrusion Detection System (IDS) [3].

a) Types of IDS

IDS can be categorized into two types namely Host-based IDS (HIDS) and Network-based IDS (NIDS) based on the proximity of network they operate in. Host-based IDS is operated in a host system, it will record important file attributes including hashes of the files and detects any suspicious activity within the host system. Network-based IDS is operated at network level where many individual systems are connected to each other in a huge network. It analyses network traffic patterns and formulates attack signatures for securing the network from attacks. We can analyse the traffic network patterns by using the steps: Identify Your Data Sources, Determine the Best Way to Collect from Data Sources, Determine Any Collection Restrictions, start a Small and Diverse Data Collection, Determine the Data Collection Destination, Enable Continuous Monitoring, View and Search Collected Data, Set Up Alerts.

b) Detection methods in IDS

There are two methods of detection in IDS:

1) Signature-based method:

This method finds threats based on special patterns like number of bytes or number of 1's /0's in the data packets. It also detects based on already known suspicious pattern that is used by a cyber-threat in the network [2]. These detected patterns in the IDS are called signatures. The system detects threats whose signatures are already present in the system, but it becomes complex to detect new type of attacks as their signatures are not recorded in the system [2],[4].

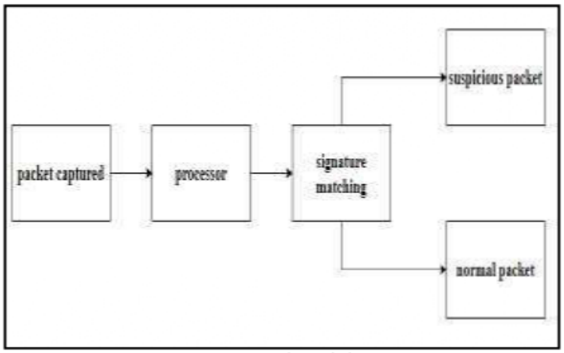


Figure 2. Signature-based detection

2) Anomaly-based method:

This method helps in detecting unknown cyber-attacks as new threats are developed quickly [4]. In this method, machine learning is used to create a model. Incoming data is analysed by that model and it is declared suspicious if it does not find matching data pattern in the model. ML based strategy has a better generalized approach compared to signature-based IDS as these machine learning powered models can be trained according to the type of device and network.

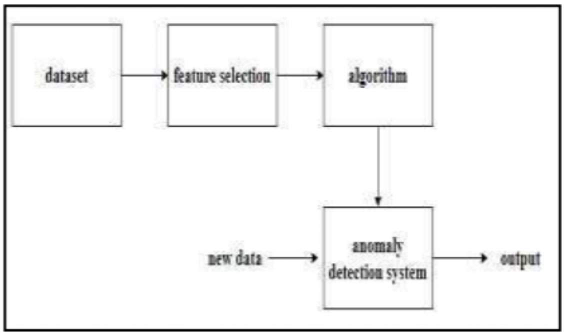


Figure 3. Anomaly-based detection

IV. MACHINE LEARNING

Machine Learning is the field of study which provides computers/machines the power to learn without being programmed explicitly. The difference between traditional computers and computers powered with machine learning is the way they carry out their process. Traditional computers take in data and logic and generate output whereas machine learning powered computers take data and previous output and generate their own algorithm specific to the data and output.

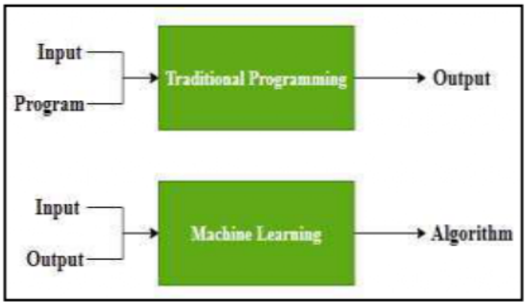


Figure 4. Traditional programming approach vs machine learning

a) Types of machine learning approaches

- Supervised learning
- Unsupervised learning
- Semi-supervised learning
- Reinforcement learning

Supervised learning: Supervised Learning aims at learning the rules of mapping between inputs and outputs [4]. The output of a supervised learning model could be a category from a finite set i.e. {yes, no} or {low, intermediate, high}. This is called classification. When the output of supervised learning model is a scalar (number), then it is called regression.

Unsupervised learning: In unsupervised learning, only previous inputs are given to the model. There will be no reference labelled output to be predicted [4]. It finds complex patterns hidden in the data.

Semi-supervised learning: The combined features of supervised and unsupervised learning give rise to semi-supervised learning. Here, output label is not given to each and every input but also there will not be enough freedom for the model to extract characteristics on its own.

Reinforcement learning: The most complex learning is reinforcement learning. It uses feedback networks to make rules. It uses positive and negative feedback to extract characteristics from the given data.

b) Machine learning in network security

The number of new cyber-attacks is increasing day by day and there are not enough cyber security analysts to handle these new attacks [4]. So, there is huge demand for machine learning in network security. Machine learning provides the system with learning capability, making the system learn from previously available data and detect cyber-attacks. Another important feature that machine learning provides to detection systems is the ability to learn from previous wrong output and adjust weights according to the feedback received. This helps the systems to adapt to any kind of attack and provide strong security to the systems in the network [6]. It also reduces dependency of a dedicated network engineer by analysing the data packets in the network automatically and updating itself continuously.

V. VIRTUALIZATION, CONTAINERIZATION AND CLOUD SERVICES

In some scenarios, some applications perform better on one operating system and some other applications perform better on another operating system (OS). After the applications are separately developed in their respective operating systems, they need to run simultaneously in one device. This was a big problem as the applications were developed in different operating systems and they do not run on the same operating system simultaneously. To overcome this problem, virtualization was developed. Virtualization is the process of dividing the hardware resources of the device and allocating them to different operating systems hosted on a single hardware device [5].

a) Importance of containerization over virtualization

Containerization is a process of combining dependencies/libraries and the source code of an application into an isolated file and making it platform/software/machine independent. The containers are independent of each other and do not interfere in the functioning of each other. Containerization was developed to deal with the limitation of virtual machines. Containers are a lightweight alternative to virtual machines [5]. Containers consume very less resources than that of virtual machine for the same task. In a virtual machine infrastructure, the resources of the host operating system are divided according to the requirement of different operating systems. Each virtual machine contains separate operating system required for the specific application [5]. A hypervisor is a software that manages the allocation of resources to different operating systems. So, to run more than one application with different requirements one has to install more than one operating system on a single device which is not resource efficient. Containers contain a minimized version of the operating system required for the application to run instead of having the whole operating system as in the case of virtual machine [5].

b) Cloud service

A cloud represents a large database which is present at a remote place. Storing data in the local system has many disadvantages like loss of data when there is system crash, insufficient memory to store large amount of data etc. Instead of storing data in a local system or in local memory, the users/organizations can store data at a remote database to have a permanent record of their data and get access to more memory for storing large files [7]. There are many cloud service providers in the market. Cloud services have a huge database with large amounts of data storage devices with large memory capacity [6],[7]. Apart from storage they also provide many other services like machine learning services, app services, website hosting services, virtual machine services, container support services, data science services, blockchain services etc., [7]. Virtual machines provide the necessary heavy computing power and infrastructure to churn the incoming bulk data and then process it to the model.

VI. DATASET DESCRIPTION

NSL-KDD dataset is used to train the model [1]. It is a standard dataset to train intrusion detection system and make it ready for serving its purpose in network security. The dataset consists of most of the attacks over the internet and has a detailed information about each type of attack. Training dataset consists of 125973 rows (records) and 42 columns (features). Testing dataset consists of 22544 rows (records) and 42 columns (features). The records are information related to data packets/traffic in the network. It has 42 features out of which 41 features describe different parameters of the data traffic and the last feature is the label which specifies whether it is normal data packet or malicious data packet (cyber-attack). The attacks specified in the dataset fall under 4 categories:

- Denial of Service (DoS)
- User to Root (U2R)
- Remote to Local (R2L)
- Probe

1) Denial of Service (DoS): Denial of service is not being able to provide response to the requests made by the clients. It is the most common type of attack over the internet [4].

2) User to Root (U2R): User to Root attack deals with system permissions in which the attacker tries to infiltrate the system and tries to gain root or administrator access [4]. When the attacker gains access to root permissions he/she can modify the system's software configurations, steal confidential information, make the system look malicious in the network in which it resides, alter the encryption algorithms in the system at root level etc.

3) Remote to Local (R2L): In this type of attack, access to host computer is gained by the attacker. The attacker tries to locate a remote device in the network with less secured encryption mechanism. Then the attacker introduces malicious threat into the target device that helps the attacker gain local access of a remote system [4]. The attacker then tries to steal information of the target device.

4) Probe: Probe attack is also known as site scanning attack. This attack is mostly related to websites and web applications in that it tries to gain as much information as possible from the web applications. The attacker tries to know the type of operating system in the server and finds out vulnerabilities of the target system in order to initiate the attack.

VII. PROCESS

By considering the work done in [1] as a reference, the raw data is pre-processed. Some features may not contribute much in predicting the output so they can be eliminated using feature selection algorithm [1]. This data is given to multiple machine learning algorithms for training and after training these models are tested and validation metrics like confusion matrix and accuracies are compared. The algorithm with best accuracy is selected and the model is containerized and deployed in cloud. The cloud provides the user with an API with which the user can access the model in a web browser.

a) Data pre-processing

The process of converting raw data into useful and efficient format is called pre-processing [1]. It is very essential in machine learning because the quality of data plays an important role in predicting the correct output. So, the raw data should be pre-processed to be compatible with the model to get best results.

i) Mapping: The different types of attacks present in the dataset are categorized to their respective attack class type [1]. The percentage of attack class categories in the train and test datasets are plotted visually in the below figure,

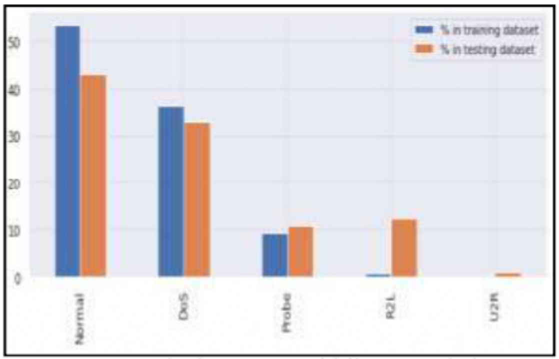


Figure 5. Bar graph of percentage of different attack classes

The no. of records and their percentage in both train and test datasets are given in the table below:

Table 1. No. of records and percentage in train and test datasets

	No. of records in train data	% in train data	No. of records in test data	% in test data
Normal	67343	53.46	9711	43.08
DoS	45927	36.46	7458	33.08
Probe	11656	9.25	2421	10.74
R2L	995	0.79	2754	12.22
U2R	52	0.04	200	0.89

ii) Standardization: Standardization is an important step in data pre-processing. It makes the different features in the dataset to be treated equally important without being biasing by the model. We standardize the dataset in a way so that the standard deviation of the values is 1 and the mean of the values is 0.

iii) Feature Selection: In a dataset with many features, it becomes very difficult to perform data processing for classification type of problems. For feature selection, random forest classifier is used to build the model, then Recursive Feature Elimination (RFE) is used to remove unwanted features that may not contribute much for training the model [1]. Here, we are selecting 10 important features that contribute more to accurate output prediction.

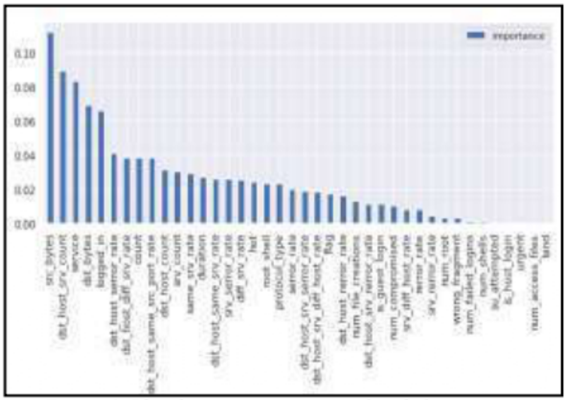


Figure 6. Plot showing importance of each feature

Figure 6. shows the importance of all the features in the dataset. By using feature selection 10 features are selected. The selected features are src_bytes, dst_bytes, logged in, count, srv_count, dst_host_srv_count,dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_serror_rate, and service. The dataset is partitioned into four sets based on attack categories (Denial of Service, User to Root, Remote to Local, Probe) [1].

iv) Encoding: Encoding is the process of converting each data point in a way such that it is understood by the computer. Encoding helps to process the data faster and the model can predict the output very fast. Here, one-hot encoding is used which overcomes the biasing problem with label encoding. In one-hot encoding instead of assigning a number to the target label and then increasing its value for the other target variable, the target variables or labels are encoded in a binary fashion [1]. Encoding is done on four datasets

b) Training and testing the model

After the data set is pre-processed, redundant features are eliminated, and important features are selected with the help of feature selection algorithm, the new dataset containing selected features is given to the model for training. The algorithms used for training the model are as follows:

- Logistic Regression (LR)
- Decision Tree (DT)
- K Nearest Neighbour (KNN)
- Random Forest Classifier (RFC)
- Naive Bayes (NB)

After the model is trained with different machine learning algorithms [4], it is tested with the testing data for validation metrics. Validation metrics like confusion matrix and accuracy are calculated for each model [1]. The metrics for different algorithms are compared and the algorithm with best accuracy is selected. K-Nearest Neighbours with accuracy approximately 98% with a random seed value when N-neighbour = 5.

c) Containerization and deployment

Once the best algorithm is selected, the model is made into a container using containerization software. This makes the model machine/software independent. The container is then stored in container repository which is a storage for containers and the user can download the corresponding container and run the machine learning model in any operating system. But in order for this approach to work smoothly, the user's system must contain containerization software. To overcome this problem the container is deployed into cloud. The container is taken from container repository/storage and deployed in a virtual machine created in cloud infrastructure. This virtual machine is supported by machine learning services of the cloud. The user can use the model via internet and API provided by the cloud service.

VII. RESULTS AND DISCUSSIONS

i) Validation metrics

The Confusion Matrix (CM) is used to calculate accuracy. It is shown in table 2. The five algorithms are trained with four datasets each and the confusion matrices and accuracies of each attack type by each of the five algorithms are given below:

Table 2. Confusion Matrix (CM)

	predicted (yes)	**predicted (no)**
actual (yes)	True Positive	False Negative
actual (no)	False Positive	True Negative

1) Denial of Service
a) Naive Bayes Classifier: Accuracy (Acc.) = 0.68

Table 3. CM for Naive Bayes (DoS)

Naive Bayes	**predicted (yes)**	**predicted (no)**
actual (yes)	3042	4416
actual (no)	1128	8583

b) Decision Tree Classifier: Acc. = 0.67

Table 4. CM for Decision Tree (DoS)

DT	**predicted (yes)**	**predicted (no)**
actual (yes)	1934	5524
actual (no)	38	9673

c) Random Forest Classifier: Acc. = 0.679

Table 5. CM for Random Forest (DoS)

RF	**predicted (yes)**	**predicted (no)**
actual (yes)	1988	5470
actual (no)	38	9673

d) K Nearest Neighbour: Acc. = 0.68

Table 6. CM for KNN (DoS)

KNN	**predicted (yes)**	**predicted (no)**
actual (yes)	2468	4990
actual (no)	414	9297

e) Logistic Regression: Acc. = 0.82

Table 7. CM for Logistic Regression (DoS)

LR	**predicted (yes)**	**predicted (no)**
actual (yes)	4549	2909
actual (no)	134	9577

2) User to Root
a) Naive Bayes Classifier: Acc. = 0.97

Table 8. CM for Naive Bayes (U2R)

Naive Bayes	**predicted (yes)**	**predicted (no)**
actual (yes)	9706	5
actual (no)	200	0

b) Decision Tree Classifier: Acc. = 0.98

Table 9. CM for Decision Tree (U2R)

DT	**predicted (yes)**	**predicted (no)**
actual (yes)	9708	3
actual (no)	200	0

c) Random Forest Classifier: Acc. = 0.98

Table 10. CM for Random Forest (U2R)

RF	**predicted (yes)**	**predicted (no)**
actual (yes)	9711	0
actual (no)	200	0

d) K Nearest Neighbour: Acc. = 0.98

Table 11. CM for KNN (U2R)

KNN	**predicted (yes)**	**predicted (no)**
actual (yes)	9709	2
actual (no)	200	0

e) Logistic Regression: Acc. = 0.12

Table 12. CM for Logistic Regression (U2R)

LR	predicted (yes)	predicted (no)
actual (yes)	1134	8577
actual (no)	130	70

3) Remote to Local

a) Naive Bayes Classifier: Acc. = 0.78

Table 13. CM for Naive Bayes (R2L)

Naive Bayes	predicted (yes)	predicted (no)
actual (yes)	9710	1
actual (no)	2754	0

b) Decision Tree Classifier: Acc. = 0.73

Table 14. CM for Decision Tree (R2L)

DT	predicted (yes)	predicted (no)
actual (yes)	9127	584
actual (no)	2745	9

c) Random Forest Classifier: Acc = 0.78

Table 15. CM for Naive Random Forest (R2L)

RF	predicted (yes)	predicted (no)
actual (yes)	9711	0
actual (no)	2752	2

d) K Nearest Neighbour: Acc. = 0.78

Table 16. CM for KNN (R2L)

KNN	predicted (yes)	predicted (no)
actual (yes)	9711	0
actual (no)	2754	0

e) Logistic Regression: Acc. = 0.31

Table 17. CM for Logistic Regression (R2L)

LR	predicted (yes)	predicted (no)
actual (yes)	1142	8569
actual (no)	3	2751

a) Naive Bayes Classifier: Acc. = 0.79

Table 18. CM for Naive Bayes (Probe)

Naive Bayes	predicted (yes)	predicted (no)
actual (yes)	7310	2401
actual (no)	134	2287

b) Decision Tree Classifier: Accuracy= 0.86

Table 19. CM for Decision Tree (Probe)

DT	predicted (yes)	predicted (no)
actual (yes)	9016	695
actual (no)	942	1479

c) Random Forest Classifier: Acc. = 0.87

Table 20. CM for Random Forest (Probe)

RF	predicted (yes)	predicted (no)
actual (yes)	9025	686
actual (no)	866	1555

d) K Nearest Neighbour: Acc. = 0.21

Table 21. CM for KNN (Probe)

KNN	predicted (yes)	predicted (no)
actual (yes)	81	9630
actual (no)	7	2414

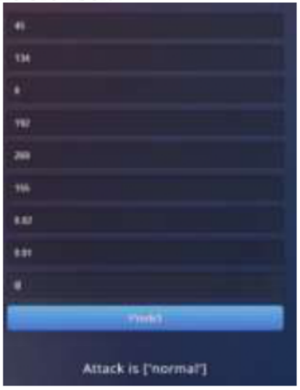
e) Logistic Regression: Acc. = 0.86

Table 22. CM for Logistic Regression (Probe)

LR	predicted (yes)	predicted (no)
actual (yes)	8098	1613
actual (no)	377	2044

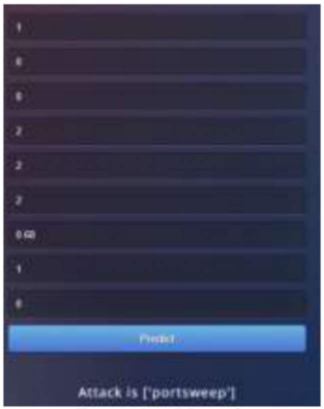
ii) API results

A Uniform Resource Locator (URL) is generated by cloud provider and it can be accessed in a web browser. In order to provide a good User interface (UI), HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets) along with a python web framework (Flask) are used. The selected features are given as input to the API which takes the input and transfers it to the cloud where the trained model is deployed. The model processes the input data and gives the output back to the API service. The API displays the predicted output in the browser.



Attack is ['normal']

Figure 7. API results



1

0

0

2

2

2

0.60

1

0

Attack is ['portsweep']

Figure 8. API results

IX. CONCLUSION

Cyber-attacks are increasing day by day and the need for efficient systems is necessary. The systems/software should be available for every individual/organization. The process discussed in this paper meets the above requirements by using machine learning to make the software efficient and accurate and deployment in cloud provides high availability and many individuals can access it with minimal cost. This can be further integrated within other software or applications thereby making it customizable and extend its functionalities.

REFERENCES

[1] R. Patgiri, U. Varshney, T. Akutota and R. Kunde, "An Investigation on Intrusion Detection System Using Machine Learning", 2018 IEEE Symposium Series on Computational Intelligence(SSCI), Bangalore, India, (2018), pp. 1684-1691.

[2] T. P. Gondaliya, H. D. Joshi, H. J. Joshi, "Different Tools and Types of Intrusion Detection System with Network Attacks", 2nd International Conference on Multidisciplinary Research & Practice, vol. 3, no. 1, (2015), pp. 290-296.

[3] P. Sadotra, Dr. C. Sharma, "A Review on Integrated Intrusion Detection System in Cyber Security[J]", International Journal of Computer Science and Mobile Computing (IJCSMC), vol. 5, no. 9, (2016), pp. 23-28.

[4] M. Almseidin, M. Alzubi, S. Kovacs, M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system[C]", IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), (2017).

[5] C. Pahl, "Containerization and the PaaS Cloud", IEEE Cloud Computing, vol. 2, no. 3, (2015), pp. 24-31.

[6] S. Miller, K. Curran, T. Lunney, "Cloud-based machine learning for the detection of anonymous web proxies[C]", 27th Irish Signals and Systems Conference (ISSC), (2016).

[7] S. Roschke, F. Cheng, C. Meinel, "Intrusion Detection in the Cloud", 8th IEEE International Conference on Dependable, Autonomic and Secure Computing, (2009).

[8] Baraneetharan, E. "Role of Machine Learning Algorithms Intrusion Detection in WSNs: A Survey." Journal of Information Technology 2, no. 03 (2020): 161-173.

[9] Shakya, Subarna. "Modified Gray Wolf Feature Selection and Machine Learning Classification for Wireless Sensor Network Intrusion Detection." IRO Journal on Sustainable Wireless Systems 3, no. 2 (2021): 118-127.