Snehal Devidas Wahane, 2021, 9:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Different Layers of Computing that's Protect from Cyber Attack Data Using Machine Learning Algorithms

Ms. Snehal Devidas Wahane
Department of Computer Science & Engineering,
HVPM COET Amravati,
India.

Associate Prof. Dr. R. R. Keole (HOD)
Department of Information Technology & Engineering,
HVPM COET Amravati,
India.

Abstract- Attack and Anomaly detection in Internet of Things (IoT) Instruction is raising concern in the domain of Internet of Things (IoT), The now days daily network traffic in a smart city via IoT systems is increasing new cyber security challenges to connection IoT devices are being connected to the sensors or actuator that are directly connected to massive cloud servers. This paper gives a detail analysis of various applications based on Internet of Thing (IoTs). This explains about how internet of things evolved from mobile computing and ubiquitous computing. It emphasizes the fact that objects are connected over the internet rather than people. The properties of Internet of Things (IOT) are information, electronic tag, standard expressed and uploading information on to the could server. IOT applications are used in domains such as healthcare, supply chain management, defence and agriculture. Various research areas are there to make the cities, villages, and campuses IoT enabled and smart to provide a quality culture to address the IoT cyber security threats, An Anomaly Detection IoT (AD-IoT) system is used, which is intelligent anomaly detection based on machine learning algorithm.

Keywords: - Cyber security, IOT, Anomaly Detection Etc.

I. INTRODUCTION

With the continuous evolution of the IoT applications, attacks on those IoT applications continue to grow day by day rapidly. In this systematic literature review (SLR) paper, our goal is to provide a research asset to researchers on recent research trends in IoT cyber attacks security. We proposed six research questions related to IoT security and machine learning algorithm mostly used for security basis.

This extensive literature survey on the most recent publications in IoT security identified a few key research trends that will drive future research in this AD-IOT field. With the fastly more growth of large scale IoT attacks, it is important to develop models that can integrate state of the art techniques and technologies from big data and machine learning. Accuracy and efficiency are key quality factors in finding the best and more algorithms used and models to detect IoT attacks in real or near real-time
Internet of Things (IoT) consists of billions millions of connected devices that can share and receive data over the internet from source to destination addresses to be access. Today these devices can be mostly found everywhere in homes, transportation, healthcare, telecommunication, offices, agriculture, etc.

IoT devices day by day are growing rapidly; making a big difference in our daily lives, and helping industries like transportation and healthcare make critical decisions. [104] reported in the findings of Business Insider's 2020 IoT report that the IoT market is expected to grow over $2.4 trillion annually by 2027. This includes the growth of IoT devices from 8 billion in 2019 IoT has brought significant benefits to our daily lives, society, and industries; however, the technology used has still not matured enough to provide secure devices and communication. An increase in connected devices gives adversaries more options to gain access to devices and use them to the launch large-scale attacks that shows the accuracy our data will be malicious data or not. The operations of IDS are divided into three stages.

The first stage is monitoring stage, the second stage is the analysis stage and the final stage is detection stage. The architecture is based on the advantage of fog computing to reduce the latency between cloud and IoT sensor. It comprises of three layers that include application layer, fog layer and IoT sensor layer. The Fog layer is a major component of the architecture, which ensures processing and aggregation of the data. The AD-IoT system is designed to monitor all IoT traffic in a distributed fog layer and alert the administrator or the service provide.

II. LAYERS OF COMPUTING

1. Fog Layer:

Fog layer consist of certain IoT services, like prediction, monitoring, planning, inferring, diagnosis, maintenance i.e. pre-processing, which is closer to edges so that it enables a faster local automation and decision making. It is a decentralized computing architecture where data, communications, storage, and applications are distributed between the data source and the cloud.

That is, it is a horizontal architecture that shares resources and services stored anywhere in the cloud to Internet of Things devices. In a very brief and simplified way, fog computing will be the fog layer below the cloud layer, managing the connections between the cloud and the network edge [3].The big difference between fog computing and cloud computing is that it is a centralized system while the former is a distributed decentralized infrastructure.

2. Cloud Layer:

Cloud computing allows a large number of computing tasks to share high-speed hardware resources through the establishment of large-scale computing center and virtualization technology, which can effectively reduce computing and hardware maintenance costs. The layer focuses more on the application of high latency data with a large number of data types and complex computational model. Cloud computing is actually a model for the availability and use of Information and Communication Technologies, which enables remote access via the Internet to a range of shared computing media in the form of services.

All the sensor data is stored on cloud hosted servers, which store and process data for analysis and decision making.

AD-IoT system promises to intelligently detect zero-attacks and IoT bot nets in distributing detection in the fog layer. This AD-IoT system design model is supposed to consist of several components involving a massive amount of IoT devices connected to distributed fog network privately or publicly. Detecting from this intelligent model distributed at each fog node, it should detect the new attacks to alert the cloud server management. AD-IoT Security Gateway IDS System is placed on a master fog node that can intelligently monitor the communication among the network traffic data.

AD-IoT system is based on the ensemble methods, which are used to improve the performance of algorithm in system model. Cyber attacks can find the vulnerable IoT devices either in private or public networks.

NIDS can utilize machine learning algorithm to classify and detect malicious behavior in the IoT fog network. This can be done by applying the NIDS system through use of the anomaly detection method based on the machine learning algorithms, which uses statistical analysis to clean and prepare data for an intelligently predictive model.

Thus, AD-IoT approach can enhance the performance for effectively detecting the cyber attacks in fog node, rather than detecting in the cloud layer, which guarantee a lightweight feature, less latency, and lower consumption than the cloud layer, which has a massive amount of data in a big infrastructure [4].

Although cloud computing has transformed the world of business in a dramatic way, its detection model cannot satisfy the requirements of fog-to-things computing. This is because the remote cloud architecture suffers from high response time and scalability issues to implement attack detection schemes for IoT devices. For this reason, it seems that fog architecture plays a substantial role in offloading cyber security breach detection functions from smart things and the cloud.

The distributed architecture of fog computing has a double role in that it reduces the storage space and computing power of security functions from IoT devices, and decreases latency issues associated with the cloud. Fog nodes are the most efficient spot where attacks can be detected in IoT due to their distribution and resource limitations [5].

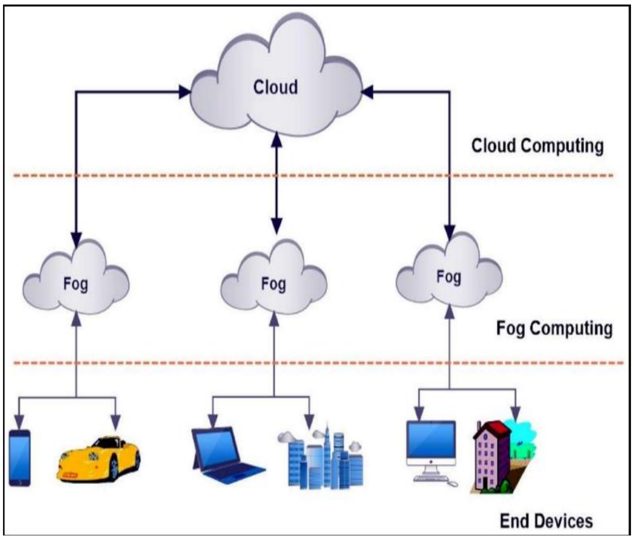

Fig 1. Architecture: level of communication.

3. Edge Layer:
Edge computing can be defined as the processing of sensor data away from the centralized nodes and close to the logical edge of the network, toward individual sources of data. It effectively pushes the computational functions to the edge of the network. In other words, rather than pumping all the data back up to the cloud for analysis and action, this process takes place much closer to the data's source. Edge computing triages the data locally, reducing the backhaul traffic to the central repository. It simplifies fog's communication chain and reduces potential points of failure.

Edge devices can be anything with sufficient compute capacity and capability such as routers, switches and even the IoT sensors collecting the data, in security purpose in edge layer design many machine learning algorithm that compare with many other algorithm mostly survey used random forest algorithm in our research we also used random forest algorithm with the compare with mal algorithm and the result its prove the random forest algorithm is best to anomaly detection.

III. WORKING OF RANDOM FOREST ALGORITHM

To understand and use the various options, further information about how they are computed is useful. Most of the options depend on two data objects generated by random forests. When the training set for the current tree is drawn by sampling with replacement, about one-third of the cases are left out of the sample. This oob (out-of-bag) data is used to get a running unbiased estimate of the classification error as trees are added to the forest. It is also used to get estimates of variable importance.

After each tree is built, all of the data are run down the tree, and proximities are computed for each pair of cases. If two cases occupy the same terminal node, their proximity is increased by one. At the end of the run, the proximities are normalized by dividing by the number of trees. Proximities are used in replacing missing data, locating outliers, and producing illuminating low-dimensional views of the data.

- **Step 1:** First, start with the selection of random samples from a given dataset.
- **Step 2:** Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- **Step 3:** In this step, voting will be performed for every predicted result.
- **Step 4:** At last, select the most voted prediction result as the final prediction result.

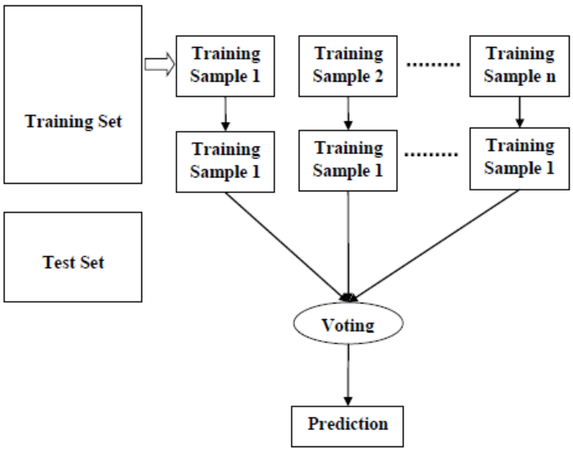

Fig 2. Working of Random Forest.

IV. WORKING OF MULTI-LAYER PERCEPTRON

An MLP is a network of simple neurons called perceptrons. The perceptron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting the output through some nonlinear activation function. Mathematically this can be written as;

$$y = \varphi(\sum_{i=1}^{n} w_i x_i + b) = \varphi(\mathbf{w}^T \mathbf{x} + b)$$

Where denotes the vector of weights, is the vector of inputs, is the bias and is the activation function.

Nowadays, and especially in multilayer networks, the activation function is often chosen to be the logistic sigmoid $\mathbf{1/(1+e^{-\infty})}$ or the hyperbolic tangent $\mathbf{\tanh(x)}$.

They are related by $(\mathbf{\tanh}(x)+\mathbf{1})/\mathbf{2} = \mathbf{1/(1+e^{-\infty})}$ these functions are used because they are mathematically convenient and are close to linear near origin while saturating rather quickly when getting away from the origin. This allows MLP networks to model well both strongly and mildly nonlinear mappings.

A typical multilayer perceptron (MLP) network consists of a set of source nodes forming the input layer, one or more hidden layers of computation nodes, and an output layer of nodes. The input signal propagates through the network layer-by-layer. The signal-flow of such a network with one hidden layer.

The computations performed by such a feed forward network with a single hidden layer with nonlinear activation functions and a linear output layer can be written mathematically as

$$\mathbf{x} = \mathbf{f}(\mathbf{s}) = \boldsymbol{B\varphi(As+a)+b}$$

Where $\mathbf{s}$ is a vector of inputs and $\mathbf{x}$ a vector of outputs. $\mathbf{A}$ is the matrix of weights of the first layer, $\mathbf{a}$ is the bias vector of the first layer. $\mathbf{B}$ and $\mathbf{b}$ are, respectively, the weight matrix and the bias vector of the second layer.

The function φ denotes an element wise nonlinearity. The generalization of the model to more hidden layers is obvious.

The supervised learning problem of the MLP can be solved with the back-propagation algorithm. The algorithm consists of two steps. In the forward pass, the predicted outputs corresponding to the given inputs are evaluated as in Equation. In the backward pass, partial derivatives of the cost function with respect to the different parameters are propagated back through the network.

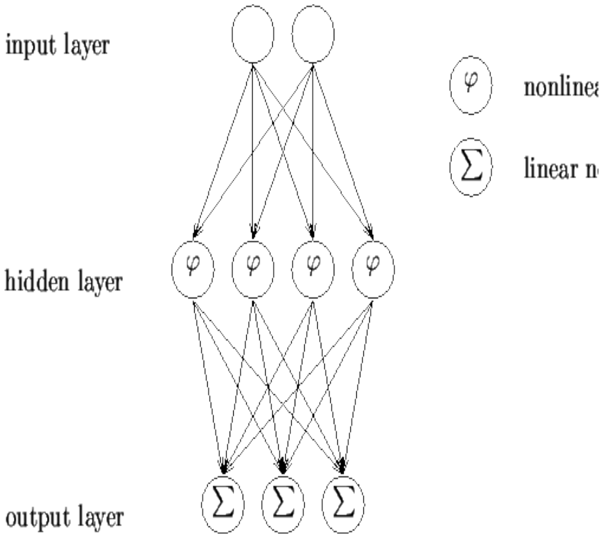

Fig 3. Signal-flow graph of an MLP.

The chain rule of differentiation gives very similar computational rules for the backward pass as the ones in the forward pass. The network weights can then be adapted using any gradient-based optimization algorithm. The whole process is iterated until the weights have converged.

The MLP network can also be used for unsupervised learning by using the so called auto-associative structure. This is done by setting the same values for both the inputs and the outputs of the network. The extracted sources emerge from the values of the hidden neurons. This approach is computationally rather intensive. The MLP network has to have at least three hidden layers for any reasonable representation and training such a network is a time consuming process.

A multilayer perceptron (MLP) is a perceptron that teams up with additional perceptrons, stacked in several layers, to solve complex problems. The diagram below shows an MLP with three layers. Each perceptron in the first layer on the left (the input layer), sends outputs to all the perceptrons in the second layer (the hidden layer), and all perceptrons in the second layer send outputs to the final layer on the right (the output layer).

A perceptron follows these steps:

1. Takes the inputs, multiplies them by their weights, and computes their sum. The weights allow the perceptron to evaluate the relative importance of each of the outputs. Neural network algorithms learn by discovering better and better weights that result in a more accurate prediction. There are several algorithms used to fine tune the weights, the most common is called back propagation.

2. Adds a bias factor, the number 1 multiplied by a weight. This is a technical step that makes it possible to move the activation function curve up and down, or left and right on the number graph. It makes it possible to fine tune the numeric output of the perceptron.

3. Feeds the sum through the activation function. The activation function maps the input values to the required output values. For example, input values could be between 1 and 100, and outputs can be 0 or 1. The activation function also helps the perceptron to learn, when it is part of a multilayer perceptron (MLP). Certain properties of the activation function, especially its non-linear nature, make it possible to train complex neural networks.

4. The result is the perceptron output; the perceptron output is a classification decision. In a multilayer perceptron, the output of one layer's perceptron is the input of the next layer. The output of the final perceptron, in the "output layer", is the final prediction of the perceptron learning model shows the comparison between both the classifiers i.e. RF and MLP.

The results of RF are almost perfect and the results of MLP are below the expectations. The Accuracy of RF is 98% and MLP consist only 53% which is very poor percentage. The accuracy and false alarm rate of the techniques are assessed, and the results revealed the superiority of the RF compared with MLP, which shows a huge difference and prove the RF as most efficient algorithm with binary classification as well as with multi- classification.

Table 1. Comparison between RF and MLP.

Parameter Name	Average Performance	
	Rf	Mlp
Accuracy	0.983	0.536
Precision	0.956	0.442
Recall	0.865	0.404
F1-Score	0.897	0.312
Hit Rate	86.47	20.01
Miss Rate	35.52	78.84
Positive Predictive Value	95.85	2.191
Negative Predictive Value	99	94.7
False Discovery Rate	2.65	57.7
False Omission Rate	0.128	0.465
Mcc	0.9795	0.40736
Kappa Score	0.9794	0.36177

V. SUMMARY

Cloud, fog and edge have created an abundance of new opportunities for professionals and companies. If your business relies on extensive digital communications, transportation and process management, you are going to need to better understand how these approaches and how they can improve your data security

REFERENCES

[1] Abeshu and N. Chilarnkurti, "Deep learning: the frontier for distributed attack detection in fog-to-things computing," IEEE Communications Magazine, vol. 56, no. 2, pp. 169-175, 2018.

[2] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in Wireless Networks and Mobile Communications (WINCOM), 2016 International Conference on. IEEE, 2016, pp. 258-263.

[3] B. Zarpelao, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in internet of things;' Journal of Networkand Computer Applications, vol. 84, pp. 25-37, 2017.

[4] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the internet of things," in Services (SERVICES), 2015 IEEE World Congress on. IEEE, 2015, pp. 21-28.

[5] H. Habibzadeh, T. Soyata, B. Kantarci, A. Boukerche, and C. Kaptan, "Sensing,

communication and security planes: A new challenge for a smart city system design r," Computer Networks, vol. 144, pp. 163- 200, 2018.

[6] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015, pp. 1-6.

[7] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards developing network forensic mechanism for botnet activities in the iot based on machine learning techniques," in Mobile Networks and Management:9th International Conference, MONAMI 2017, Melbourne, Australia, December 13-15, 2017, Proceedings, vol. 235. Springer, 2018, pp. 30--44.

[8] M. Nobakht, V. Sivaraman, and R. Boreli, "A host-based intrusion detection and mitigation framework for smart home iot using open flow," in Availability, Reliability and Security (ARES), 2016 11th International Conference on. IEEE, 2016, pp. 147-156.

[9] H. Summerville, K. M. Zach, and Y. Chen, "Ultra-lightweight deep packet anomaly detection for internet of things devices;' in Computing and Communications Conference (IPCCC), 2015 IEEE 34th International Performance. IEEE, 2015, pp. 1-8

[10] S. Garg, K. Kaur, N. Kumar, S. Batra, and M. S. Obaidat, "Hyclass: Hybrid classification model for anomaly detection in cloud environment," in 2018 IEEE International Conference on Communications (ICC). IEEE, 2018, pp. 1-7.

[11] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, vol. 60, pp. 19-31, 2016.

[12] J. Howell. Number of connected iot devices will surge to 125 billion by 2030, ihsmarkit says - ihs technology. [Online]. Available: https://technology.ihs.com/596542/ last accessed: 11/07/2018.

[13] Borgia, "The internet of things vision: Key features, applications and open issues," Computer Communications, vol. 54, pp. 1-31, 2014.

[14] Restuccia, S. D'Oro, and T. Melodia, "Securing the internet of things: New perspectives and research challenges," IEEE Internet of Things Journal, vol. 1, no. 1, pp. 1-14, 2018.