International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-05, Issue-08, Nov 2019

Providing Security to Data Using Machine Learning

[1]Amalapurapu Srinag, [2]Tankasala Pravallika, [3]Rabbani Mahaboob

[1,2,3]Assistant Professor, Vignan's Nirula Institute of Technology & Sciences, Guntur & India,

[1]srinag401@gmail.com, [2]pravallika486@gmail.com, [3]rabbanivlsi@gmail.com

Abstract **Present six billions estimated devices are connected to internet; by 2020 it will be 25 billion. During this growth security has been identified as one of the weakest areas in Internet of Things (IOT). So to meet different challenges in securing IOT, we propose using machine learning within an IOT Gateway to help secure the system. By using Regression in a gateway to detect anomalies in the data sent from edge devices**

Keywords — ***Internet of Things, machine Learning, Regression, security, Gateway, Edge Device.***

I. INTRODUCTION

Internet of Things (IoT) is presently a hot technology worldwide. Government, academia, and industry are involved in different aspects of research, implementation, and business with IoT. IoT cuts across different application domain verticals ranging from civilian to defense sectors. These domains include agriculture, space, healthcare, manufacturing, construction, water, and mining, which are presently transitioning their legacy infrastructure to support IoT. Today it is possible to envision pervasive connectivity, storage, and computation, which, in turn, gives rise to building different IoT solutions. IoT-based applications such as innovative shopping system, infrastructure management in both urban and rural areas, remote health monitoring and emergency notification systems, and transportation systems, are gradually relying on IoT based systems.

Over the past decade, the popularity of Python as a mainstream programming language has exploded. Notable advantages of Python over other languages include, but are not limited to; It is a very simple language to learn and easy to implement and deploy, so you don't need to spend a lot of time learning lots of formatting standards and compiling options. It is portable, expandable and embeddable, so, it is not system dependent, and hence supports a lot of single board computers on the market these days, irrespective of architecture and operating system. Most importantly, it has a huge community which provides a lot of support and libraries for the language.

We are living in a world surrounded by billions of computing systems, identifying, tracking, and analyzing some of our intimate personal information, including health, sleep, location, and network of friends. The trend is toward even higher proliferation of such devices, with an estimated 50 billion smart, connected devices by 2020, according to a recent report by Cisco. These devices generate, process, and exchange a large amount of sensitive information and data (often collectively referred to as "security assets" or simply "assets"). In addition to private end-user information, assets include security-critical parameters introduced during the system architecture definition, e.g., fuses, cryptographic, and digital rights management (DRM) keys, firmware execution flows, and on-chip debug modes. Malicious access to these assets can result in leakage of company trade secrets for device manufacturers or content providers, identity theft or privacy breach for end users, and even destruction of human life. Security assurance of a modern computing device involves a number of challenges. One key challenge is the sheer complexity of the design. Most modern computing systems are architected via a system-on-chip (SoC) paradigm, viz., through a composition of predesigned hardware or software blocks [referred to as intellectual properties (IPs)] that interact through a network of on-chip communication fabrics. The IPs themselves are highly complex artifacts optimized for performance, power, and silicon overhead. Adding to the complexity are the communication protocols used in implementing complex system-level use cases. Finally, security assets are sprinkled at different IPs across the design, and access to the assets is governed by complex security policies. The policies are defined by system architects as well as different IP and SoC integration teams, and undergo refinement and modification throughout the system development. This makes it challenging to validate a system, develop architectures to provide built-in resilience against unauthorized access, or update security requirements, e.g., in response to changing customer needs. Another source of challenge is the supply chain involved in the development of a modern computing device. There is a large number of players involved, including IP providers, SoC design house, and foundry. With the increasing globalization of the semiconductor

 DOI : 10.35291/2454-9150.2019.0519

design and fabrication process, each of these players often involves large number of organizations often across.

Geography coordinating to create a complex supply-chain pipeline. Every component of the pipeline is vulnerable to malicious design alterations, subversions, piracy, and other security threats. Even in cases where a component is designed without intended malice, aggressive time-to-market requirements and high optimization needs often result in errors and vulnerabilities inadvertently left in the design, which can be exploited by a malicious adversary in the field. Given the broad spectrum of vulnerabilities and corresponding mitigation strategies, the subject of SoC security today is highly fragmented.

II. Related Work

Gaps within security techniques are to protect sensor nodes, to maintain trust between devices, and to defend against Man in the Middle attacks, Denial of Service (DOS) attacks, etc. They concluded that there is currently extensive work occurring within IoT authentication and access control protocols but other work needs to be done as well. IoT requires intelligent processing and reliable transmission within the network. To provide this, the network architecture contains three layers: the application layer, the transport layer, and the sensing layer. The application layer contains the logical link between the user and the Internet through intelligent applications. Intelligent applications include smart home furnishings and intelligent architectures. The application layer uses machine learning, data mining, data processing, and other analytics to process information from the system and provides an output. The transport layer consists of network communications including Wi-Fi, Bluetooth, ZigBee, and 802.15.4. The transport layer contains the gateway or gateways that process the information and relay the information across the network. The sensing layer contains edge devices that are composed of a variety of sensors and actuators that collect data and send it through the transportation layer to the application layer for analysis. There are many security threats present in the transport layer. Our approach is to add machine learning within the transport layer to help determine if there are interruptions in the data transfer and to monitor the edge devices from the sensing layer. This approach will also address by addressing the entire system security, not simply the authentication and access control protocols.

III. Approach

In examining the approach, we will begin with an overview of our test bed creation and then discuss our machine learning methodology.

A. Test bed creation

First we installed Spyder 3.6 and imported libraries in Spyder 3.6 after that imported dataset by using

Dataset = pd.read_csv (“Data.csv”)

After that I declared 2 variables X & Y: X selects all columns except last one column in dataset and Y selects the only last column in dataset by using iloc. If any missed values are in dataset by using

From sk.learn.preprocessing import imputer

Library it calculate all the values by using mean such that it replaces the missed data in dataset. In the given dataset we

are having France, Germany, Spain, 3 variables which are present in X variable. So the machine learning will get confused regarding that 3 variables because machine learning can understand only mathematical equations and numbers, so to solve this problem I have use dummy variables which is known as One Hot Encoder in the programming. After that I had split the data set in to raining test and test set with ratio of 0.2i.e.., 80% trained data and 20% test data by using library

From sk.learn.cross_validation import train_test_split.

In the given dataset if the values are in different ranges I have used feature scaling to solve the problem in the machine learning. Finally by using simple linear regression technique I have predicted the test values by comparing trained values

B. Machine Learning Methodology

Machine learning is the use of algorithms within a program to learn from collected data. Within machine learning there are various algorithms that exist to learn from data. We chose to implement a Simple Linear Regression technique to monitor the system. A Simple Linear Regression (SLR) is a type of machine learning that is modeled to predict outcome. To create an SLR, we chose to use Python. Spyder 3.6 is a statistical programming tool that allows for computations. Packages are readily available in Python for machine learning, statistics, graphing, probability, etc. We chose to use Pandas package. The Pandas package allows us to analyze data to use for predictions.

IV. Experiments and Results

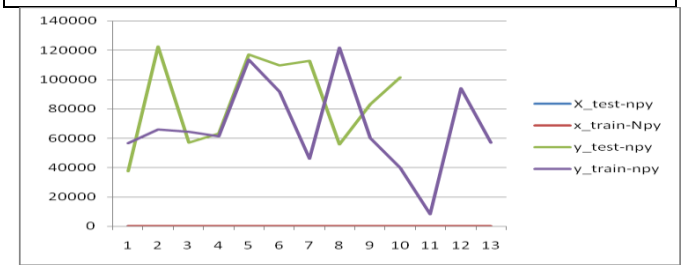
In my data set it contains 50 columns about an organization like salaries, experience, etc.out of these I have taken 20% of data to test set and remaining 80% as trained set.

X_test-npy	x_train-Npy	y_test-npy	y_train-npy
1.5	2.9	37731	56642
10.3	5.1	122391	66029
4.1	3.2	57081	64445
3.9	4.5	63218	61111
9.5	8.2	116969	113812
8.7	6.8	109431	91738
9.6	1.3	112635	46205

 DOI : 10.35291/2454-9150.2019.0519

4	10.5	55794	121872
5.3	3	83088	60150
7.9	2.2	101302	39891
-	5.9	-	8163
-	6	-	93940
-	3.7	-	57189

Table 1. Data Set



Spliting data set in to trained set and test set..

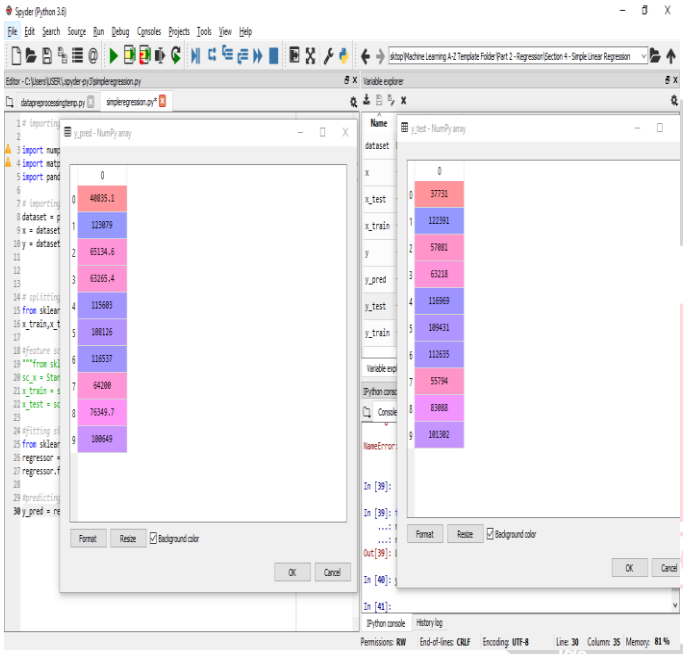


Fig.2 Predicted test set with trained set

V. COMPARISONS OF RESULTS

S.no	y_pred	y-test	Differences	Detected
1	40835	37731	0.3104	yes
2	123079	122391	0.0688	yes
3	65134.6	57081	0.80536	yes
4	63265.4	63218	0.00474	yes
5	115603	116969	-0.1366	Yes
6	108126	109431	-0.1305	Yes
7	116537	112635	0.3902	Yes
8	64200	55794	0.8406	Yes
9	76349.7	83088	-0.67383	Yes
10	100649	101302	-0.0653	Yes

Table 2. Comparison Between y prediction and y test

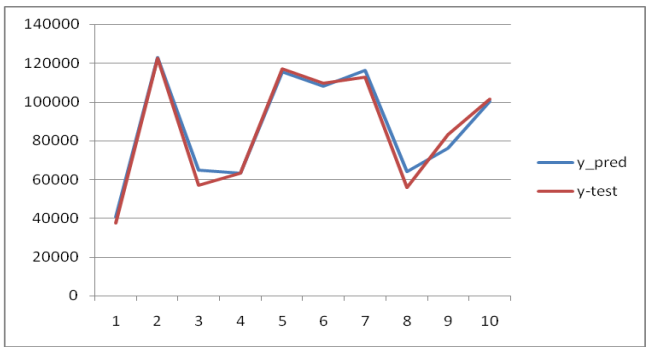


Fig. 3 Comparision Between y_pred and y_test using Data Preprocessing

Comparison		
Fig 2	**Fig 3**	**Accuracy is 78.2%**

Table 3. Comparison figures 2 & 3.

By comparing figures 2 and 3 we can say that after receiving data the accuracy measured is 97.6%

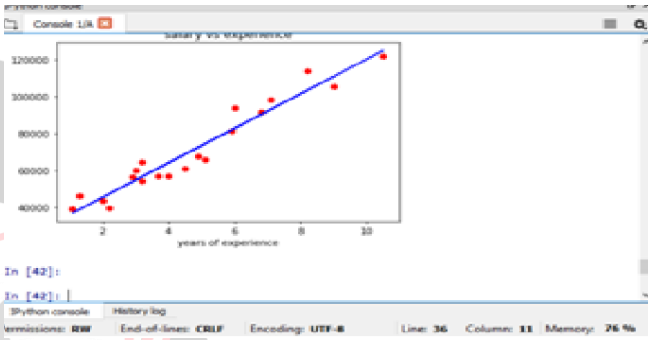


Fig. 4 Comparision Between y_pred and y_test after Data preprocessing

Fig. 5 Comparision Between y_pred and y_test after Data preprocessing using Linear Regression.

We used Arduino Uno devices to emulate edge devices. Each Arduino Uno was connected to an ESP8266 WIFI chip. We then connected 6 edge devices to a Raspberry Pi Model 3 device to implement a gateway. The Raspberry Pi Model 3 contains 1GB of RAM which allows for machine learning of smaller data sets. We were limited in the number of devices we can connect because of the wireless driver used in the Raspberry Pi 1. The table-1 data was sent from that edge devices to the gateway. Finally I predicted the data as shown in table 1.2 with accuracy 97.6%.

Comparison		
Fig 4	**Fig 5**	**Accuracy is 97.6%**

Table 4.. Comparison figures 2 & 3.

 DOI : 10.35291/2454-9150.2019.0519

VI. CONCLUSION

As billions of IoT devices become connected together, securing these devices and systems is fundamental. Currently, there is a weakness in IoT security that needs to be addressed. While many people are addressing weaknesses in authentication and access control, securing the system as a whole is crucial. To accomplish this, we propose the use of machine learning to detect anomalies in an IoT system. By using Linear regression, we were able to train the network to detect invalid data points. During our tests, because of our limited availability of valid data points we encountered difficulties. We were able to add additional data points that were invalid readings and retrained our model to detect both valid and invalid data points. We determined that our approach shows promise in detecting invalid data points in IoT systems. We are convinced that the next steps is to create a large scale test bed and collect an increased amount of data to train the system. Finally, we conclude that the use of Linear Regression within an IoT gateway can offer the foundation for a more secure IoT system.

VII. FUTURE SCOPE

By Using Linear Regression we maintained accuracy 97.6% but there is deviation of 3.7%. By implementing Deep Learning we can achieve 100% Accuracy. Because ANN supports large Data sets.

REFERENCES

[1] M Janice Ca˜nedo, Anthony Skjellum, "Using Machine Learning to Secure IoT Systems," Auburn Cyber Research Center. Samuel Ginn College of Engineering. Auckland, New Zealand, pp. 219-221, 2016.

[2] Quamar Niyaz, Weiqing Sun, Ahmad Y Javaid, and Mansoor Alam on A Deep Learning Approach for Network Intrusion Detection System, 3rd ed., vol. 2. Oxford: USA, 1892, pp.456-462, 2017.

[3] Ren Junn Hwang and Yan Zhi Huang, "Secure Data Collection Scheme for Wireless Sensor Networks," 31st International Conference on Advanced Information Networking and Applications Workshops. New Taipei City, Taiwan New York: Academic, 2017, pp. 553-558.

[4] Bhavin Patel, Neha Pandya, on "Data Transfer Security solution for Wireless Sensor Network," International Journal of Computer Applications Technology and Research Volume 2– Issue 1, 63-66, 2013, ISSN: 2319–8656.

[5] Ionut Indre, Camelia Lemnaru, "Detection and Prevention System against Cyber Attacks and Botnet Malware for Information Systems and Internet of Things," 978-1-5090-3899-2/16/$31.00 ©2016

[6] Rodr´ıguez-Mota∗, P.J. Escamilla-Ambrosio†, J. Happa‡, J.R.C. Nurse, "Towards IoT Cyber security Modeling: From Malware Analysis Data to IoT System Representation," 978-1-5090-5137-3/16/$31.00 2016 IEEE.

[7] Alessandro Sforzin† and Mauro Conti, "RPiDS: Raspberry Pi IDS A Fruitful Intrusion Detection System for IoT," 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress 978-1-5090-2771-2/16 $31.00 © 2016 IEEE DOI 10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.114

[8] S. Sridhar, Dr.S.Smys, "Intelligent Security Framework for IoT Devices," 978-1-5090-4715-4/17/$31.00 ©2017 IEEE.

 DOI : 10.35291/2454-9150.2019.0519