A Heuristic Approach for Intrusion Detection in IoT Environment

K. Manasa
Department of Computer Science and Engineering
Anurag Group of Institutions, Hyderabad, Telangana, India

J. Apoorva
Department of Computer Science and Engineering
Anurag Group of Institutions, Hyderabad, Telangana, India

L. Vinay
Department of Computer Science and Engineering
Anurag Group of Institutions, Hyderabad, Telangana, India

A. Mallikarjuna Reddy
Department of Computer Science and Engineering
Anurag Group of Institutions, Hyderabad, Telangana, India
Email: mallikarjunreddycse@cvsr.ac.in

Abstract- The number of diverse interconnected Internet of Things (IoT) devices keeps increasing exponentially, introducing new security and privacy challenges. These devices tend to become more pervasive than mobile phones and already have access to very sensitive personal information such as usernames, passwords, etc., making them a target for cyber-attacks. Many smart devices are vulnerable to a variety of attacks and also, they can be considered to be the weakest link for breaking into a secure infrastructure. As a result, there is a need to develop an Intrusion Detection System (IDS) dedicated to monitoring IoT ecosystems, which will be able to adapt to this heterogeneous environment and detect malicious activity on the network.

This project is a novel IDS for the IoT, which employs Machine Learning (ML) and Deep Learning methodologies for successfully identifying network scanning probing and simple forms of Denial of Service (DoS) attacks.

Keywords- Intrusion Detection System, Internet of Things, Recurrent Neural Network, Deep Learning, Network attacks, Gated Recurrent unit, Machine Learning, Hyper-parameters

I. Introduction

Deep learning ‘mimics the brain functionality’ which means, it resembles the functionality of brain, with the help of robust neural network algorithms. The wide range of deep learning applications includes image recognition, computer vision, speech recognition, pattern recognition and behavior recognition.

In the world of IoT, the datasets are high dimensional, temporal and multi-modal. Deep Learning algorithms with robust computation power are more suitable for complex IoT datasets compared to simple machine learning techniques. The application of deep learning to the IoT Environment, particularly in IoT security is still in the initial stages of research and has a great potential to find insights from the IoT data. With smart use of deep learning algorithms, It is believed that IoT solutions can be optimized. For example, recurrent neural networks in deep learning have the capability to learn from previous time-steps of the input data. The data at each time-step is processed and stored and given as input to the next time-step. The algorithm at the next time step utilizes the previous information stored to process the information. Though the neural network structures are complex, the hyperparameters can be tuned to obtain light-weight functionality for IoT solutions. This hypothesis motivated to apply deep learning concepts to IoT network security.

A. *Problem Definition*

Currently, the entire world is witnessing the rapid product launches and high expectations from emerging Internet of Things (IoT) technology. It is growing at an accelerating pace connecting billions of devices in our daily life. But, the massive opportunities and utilities delivered by IoT technologies are shadowed by privacy trade-offs and grievous security concerns. One must consider the numerous connected devices, complexities, competing trends and diversities that must be managed while developing solutions for IoT. The current security protocols are only applicable for high powered computers for short-lived sessions. It is not viable to use the same protection technique for long-running sessions. For these reasons, IoT devices became attractive targets for the hackers making our lives endangered with unexpected threats.

This project is to develop smart security solutions, rather than per-device security for numerous IoT devices. As, It is more feasible to implement security solutions for network data.

B. *Objective of the Project*

The Main Objective of this project is to analyze the applications of deep learning to the Internet of Things (IoT) network security by evaluating recurrent neural network algorithms on the intrusion detection dataset. The importance of security in today’s connected world requires analyzing the large amount of heterogeneous data, and this cannot be possible without the help of artificial intelligence.

II. ANALYSIS

A. *Exisiting System*

To secure an IoT system, the traditional high-end security solutions are not suitable, as IoT devices are of low storage capacity and less processing power. Moreover, the IoT devices are connected for longer time periods without human intervention.

And Traditional IT such as SNORT and Bro only work on conventional IP network, they are not adaptable and they are applicable only to a single platform/protocol.

B. *PROPOSED SYSTEM*

This project is to develop smart security solutions which are light-weight, distributed and have a high longevity of service. Rather than per-device security for numerous IoT devices, it is more feasible to implement security solutions for network data. The artificial intelligence theories like Machine Learning and Deep Learning have already proven their significance when dealing with heterogeneous data of various sizes. To substantiate this, we have applied concepts of Deep Learning and to build a light-weight distributed security solution with high durability for IoT network security.

III. REQUIREMENT SPECIFICATION

A. *IDS – DATASETS*

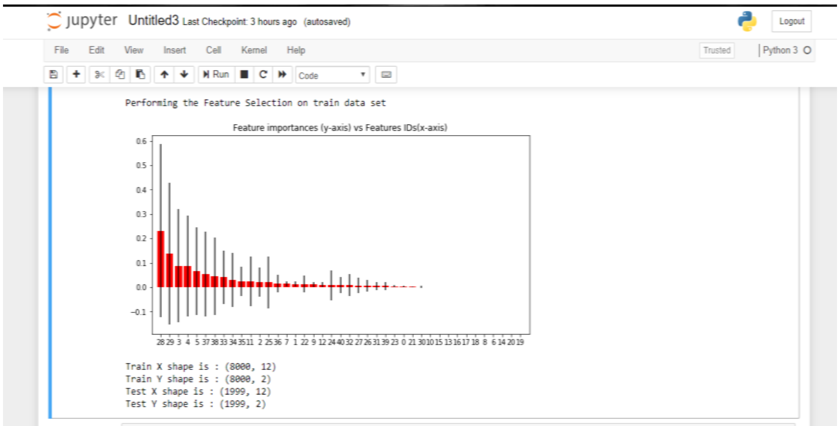
Intrusion detection data for training machine learning algorithms are limited in the literature. The most literature on the application of machine learning algorithms on intrusion detection data set uses the DARPA KDD Cup '99 dataset and hence we selected this for the project.

The DARPA KDD Cup '99 datasets were generated by the Defense Advanced Research Projects Agency (DARPA ITO) on a simulated air force model. The training data was collected for seven weeks and the testing data were collected for two weeks. The whole dataset contains 39 network-based attack types and has more than 200 instances of background traffic compared to an air force base model. The complete network traffic is either classified as one of the attack types or "normal". The datasets can be found on the UCI website where repository links to the three different versions of data set exist. The three versions of the KDD 99'Cup IDS datasets are – full KDD data set, corrected KDD, 10% KDD. Among these three, 10% KDD data set is used in most literature and hence, we are using the same for this research. The 10% KDD dataset contains 24 attack types, which are mainly categorized into four classes – Probe, Denial of Service (DoS), User to Root (U2R) and Remote to Local (R2L). The training and testing samples are represented with 41 features and a label with either "normal" or "attack type". The features can be divided into three types: the first group describes the features that are used for providing information on the command that is used for connections, the second group of features describes the specifications of the commands, and the third group describes the features that convey information about the connections having the same destination with the same service. As the GRU algorithm requires a time series dataset, we have neither randomized the sequence nor removed the duplicates, thus, making it apt for GRU classification analysis.

IV. IMPLEMENTATION

A. *Feature Engineering*

The main motive of this project is to build a light-weight security solution for IoT systems. Therefore, it is important to reduce the number of features and use only the important features required for training and testing the algorithm. We have used a random forest classifier as the feature selection technique which is proven as the best method for reducing the dimensionality for the KDD’99 Cup dataset. The random forest uses tree-based methods that rank the features importance based on their ability to improve the node purity (Gini impurity). We have graphically visualized the importance of each feature and selected the top six features for each dataset before inputting them to the model. The decrease in the features in the input data makes the model faster to train and respond, making it flexible and adaptive.



```
Performing the Feature Selection on train data set
```

```
Train X shape is : (8000, 12)
Train Y shape is : (8000, 2)
Test X shape is : (1999, 12)
Test Y shape is : (1999, 2)
```

Fig. 1 Graph of Feature Importance vs Feature IDs

1) Random Forest Classifier

A Random Forest Classifier is an ensembled machine learning technique for supervised learning tasks. This algorithm has many advantages which are useful for the development of this project. Some of them are,

- Ability to handle numerous input variables without a necessity for variable deletion.
- Can run on huge data bases efficiently
- Provides estimates of important variables for the classification
- Robust to noise and outliers when compared to single classifiers
- Lightweight when compared to other boosting methods

We have made use of the ability of the random classifier method to rank the importance of the features set to the target variables. We have selected those variables based on the maximum importance levels. Those features with low values of the importance will add less information to the learning model and are ignored based on the threshold values of the importance.

B. *Modeling*

For Modeling the dataset, we used a Deep learning algorithm called Recurrent Neural Network. We used Gated Recurrent Unit(GRU) to avoid Vanishing Gradient Problem and set the hyperparameters accordingly, checking the important parameters for efficient and accurate training.

These are the brief explanations of Deep Learning Methods,

1) Recurrent Neural Network

Recurrent neural networks (RNN) in deep learning have the capability to learn from the previous time-steps and can be used with less human intervention. In RNN, the output of each node in the hidden layer is given as input to the same node at each time-step. The useful information is stored in the memory and can be used for learning purposes in future time steps.

2) Long-Short-Term Memory RNN (LSTM)

Recurrent Neural Networks (RNN), when trained in real-time learn from previous timesteps by backpropagation through time (BPTT). A deep neural network is unfolded in time and constructs an FNN for every time-step. Then, the gradient rule updates the weights and biases for each hidden layer, thus, minimizing the loss between the expected and actual outputs. However, standard RNNs cannot perform better when the time-steps are more than 5-10. The prolonged back-propagation leads to vanishing or blow up of error signals, leading to oscillating weights, which makes the network performance poor. To overcome this vanishing gradient problem, researchers came up with the Long-Short-Term-Memory (LSTM) network which bridges the minimal time gaps. LSTM makes use of a gating mechanism to handle long-term dependencies.

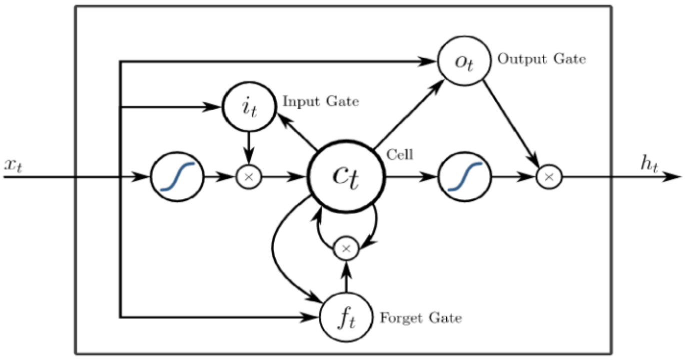


Fig. 2 Long Short Term Memory Cell

LSTM has a cell state which is passed to every-time step. A gating mechanism is used to optimize the information that is passing through. It contains a sigmoid function layer which outputs between one and zero. A value of one means “pass all the information through”, whereas the value of zero means “do not pass any information through”. The "forget gate" decides the information that needs to be let through between the current input and previous cell state output using the sigmoid function. The "input gate" decides what information is required to store in the cell state. This gate contains two functions - "sigmoid" to decide what values need to be updated, and the “tanh” function to create a new vector of values that are to be added to the cell state. The “output gate” decides on what information from the cell state is required to output with the help of a sigmoid function. The output information is passed through the “tanh” function before passing through the “sigmoid”, to make sure that the values are between -1 and +1.

3) Gated Recurrent Unit (GRU):

A Gated Recurrent Unit (GRU) is a lighter version of an LSTM where the complexity in the structure is reduced by decreasing the gates in the architecture. The GRU merges both the “forget gate” and “input gate” in an LSTM to an “update gate” and combines the hidden state and cell state, resulting in a simpler architecture of the network as shown below Figure:

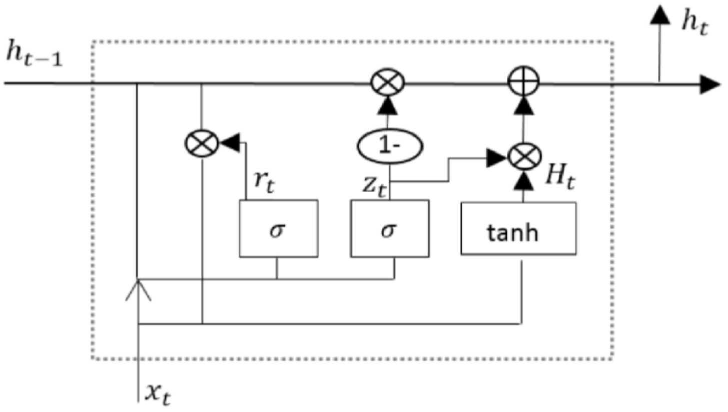


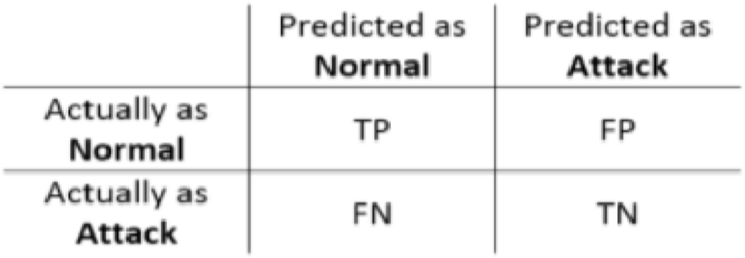
Fig. 3 Gated Recurrent Unit Cell

4) Hyper-Parameters

The hyper-parameters used in the design of the recurrent neural network have a great impact on the performance of the network [5]. Although there are many hyper-parameters involved in the design of a recurrent neural network, the parameters having the largest impact on the performance of the network are learning rate, number of hidden layers, number of units/cells in the hidden layer and the number of time-steps.

5) Evaluation Metrics

To evaluate the performance of the classification model the following metrics are used in machine learning research. In general, the confusion matrix visualizes the performance of the algorithm in a tabular form as shown in the figure below:



	Predicted as **Normal**	Predicted as **Attack**
Actually as **Normal**	TP	FP
Actually as **Attack**	FN	TN

Where,

• True Positive (TP) is the total number of samples predicted as “normal” while they were “normal”.

• False Negative (FN) is the total number of samples predicted “normal” while they were “attack”.

• False Positive (FP) is the total number of samples predicted “attack” while they were “normal”.

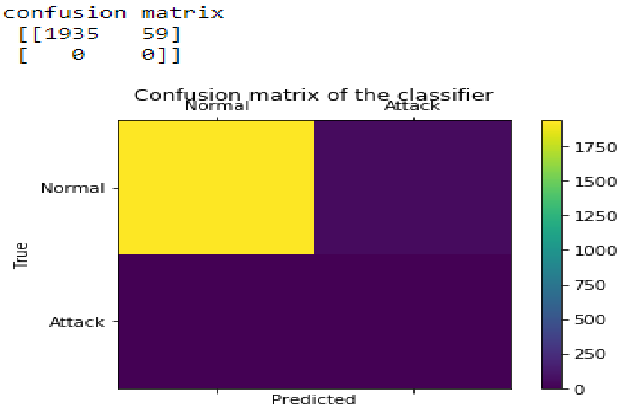
• True Negative (TN) is the total number of samples predicted “attack” while they were “attack”.

All other important metrics such as Precision, Accuracy, Recall, False Alarm Rate (FAR) and Area under ROC curve (AUC) can be calculated using these 4 measures taken from the confusion matrix as shown below:

Recall = TP / (TP+FN)

$$\text{Precision} = TP / (TP+FP)$$
$$\text{Accuracy} = (TP+TN) / ((TP+FP) + (FN+TN))$$
$$\text{FAR} = FP / (FP+TN)$$

V. Evaluation



```
confusion matrix
 [[1935   59]
 [   0    0]]
```

Fig. 4 Confusion Matrix

VI. CONCLUSION AND FUTURE ENHANCEMENT

This project is all about developing light-weight architecture for an Intrusion Detection System in an IoT network. To improve its performance, we can place IDS classifiers at each layer in TCP/IP layer architecture.

The applications of deep learning to IoT applications to develop security solutions, is still in the naïve stage, and we believe, it has a lot of potentials. As the IoT deals with user's personal data and industry's information, it is crucial to implement robust solutions to protect from security threats. This can be possible with the concepts of machine learning and deep learning as IoT generate a humongous amount of heterogeneous data. We have applied Gated-Recurrent-Unit neural networks to the dataset. However, there are many improvised versions of recurrent neural networks such as Dynamic RNN, Bi-Directional RNN which can achieve better performances than basic GRU cells. One can also build a hybrid network using convolutional neural networks and recurrent neural networks to deal with multi-modal data. This project focused on dealing with IoT devices where the processing power is low and the data size is not huge. This project can be taken forward by applying it to large amounts of real-time IoT data.

At Last, the Internet of Things is a revolution rather than an evolution. As the IoT evolves, the security issues evolve. The IoT is a boon to the society only when it is secured and this can possible with artificial intelligence.

REFERENCE

[1] *https://ieeexplore.ieee.org/document/8379722/-Pulse: An Adaptive Intrusion Detection for the Internet of Things*

[2] *https://www.hindawi.com/journals/scn/2019/4301409/-Recent Advancements in Intrusion Detection Systems for the Internet of Things*

[3] *https://github.com/manojkumar-github/Intrusion-Detection-System-for-IoT-networks-using-Gated-Recurrent-Neural-Networks-GRU*

[4] An adaptive intrusion detection and prevention system for Internet of Things - Sheikh Tahir Bakhsh, Saleh Alghamdi, Rayan A Alsemmeari, Syed Raheel Hassan, 2019.

[5] C. Ou, "Host-based Intrusion Detection Systems Inspired by Machine Learning of Agent-Based Artificial Immune Systems," 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), Sofia, Bulgaria, 2019, pp. 1-5.

[6] M. H. Ali, M. Fadlizolkipi, A. Firdaus and N. Z. Khidzir, "A hybrid Particle swarm optimization -Extreme Learning Machine approach for Intrusion Detection System," 2018 IEEE Student Conference on Research and Development (SCOReD), Selangor, Malaysia, 2018, pp. 1-4.doi: 10.1109/SCORED.2018.8711287

[7] K. Greff, et al., “LSTM: A Search Space Odyssey”, arXiv preprint arXiv: 1503. 04069, 2015.

[8] Li, Y., & Guo, L. (2007). An active learning based TCM-KNN algorithm for supervised network intrusion detection. Computers & security, 26(7), 459-467.

[9] Leung, K., & Leckie, C. (2005, January). Unsupervised anomaly detection in network intrusion detection using clusters. In Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38 (pp. 333-342). Australian Computer Society, Inc..

[10] Breiman, L., 2001. Random forests. Machine Learning 45 (1), 5–32.

[11] Hasan, M. A. M., Nasser, M., Ahmad, S., & Molla, K. I. (2016). Feature Selection for Intrusion Detection Using Random Forest. Journal of Information Security, 7(03),129.

[12] Kim, J., Kim, H. (2015). An appproach to Build an efficient Intrusion Detection Classifier. JOURNAL OF PLATFORM TECHNOLOGY, 3(4), 43-52.

[13] Devaraju S., & Ramakrishnan, S. (2014). PERFORMANCE COMPARISON FOR INTRUSION DETECTION SYSTEM USING NEURAL NETWORK WITH KDD DATASET. ICTACT Journal on Soft Computing, 4(3).

[14] Tao, X., Kong, D., Wei, Y., & Wang, Y. (2016). A Big Network Traffic Data Fusion Approach Based on Fisher and Deep Auto-Encoder. Information, 7(2), 20.