Technological University Dublin
ARROW@TU Dublin

Conference papers | School of Electrical and Electronic Engineering

2022-07-11

Experimenting an Edge-Cloud Computing Model on the GPULab Fed4Fire Testbed

Vikas Tomer
Graphic Era Deemed to be University

Sachin Sharma
Technological University Dublin, sachin.sharma@tudublin.ie

Follow this and additional works at: https://arrow.tudublin.ie/engscheleart

Part of the Digital Communications and Networking Commons

Recommended Citation

V. Tomer and S. Sharma, "Experimenting an Edge-Cloud Computing Model on the GPULab Fed4Fire Testbed," 2022 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 2022, pp. 1-2, doi: 10.1109/LANMAN54755.2022.9820006.

This Conference Paper is brought to you for free and open access by the School of Electrical and Electronic Engineering at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, gerard.connolly@tudublin.ie.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 4.0 License
Funder: NGIAtlantic H2020 project under agreement no. OC3-292

Experimenting an Edge-Cloud Computing Model on the GPULab Fed4Fire Testbed

Vikas Tomer
Graphic Era Deemed to be University
vikastomer.cse@geu.ac.in

Sachin Sharma
Technological University Dublin
Sachin.Sharma@TUDublin.ie

***Abstract*—There are various open testbeds available for testing algorithms and prototypes, including the Fed4Fire testbeds. This demo paper illustrates how the GPULAB Fed4Fire testbed can be used to test an edge-cloud model that employs an ensemble machine learning algorithm for detecting attacks on the Internet of Things (IoT). We compare experimentation times and other performance metrics of our model based on different characteristics of the testbed, such as GPU model, CPU speed, and memory. Our goal is to demonstrate how an edge-computing model can be run on the GPULab testbed. Results indicate that this use case can be deployed seamlessly on the GPULAB testbed.**

***Index Terms*—Edge Computing, IoT, Machine Learning**

I. Introduction

As more devices connect to the Internet via 5G/6G, the amount of data being transmitted over the Internet increases. Consequently, there may be an increase in the number of Internet of Things (IoT) attacks. Recent research [1] shows that ensemble machine learning algorithms detect attacks with a higher degree of accuracy than neural or kernel methods. Ensemble algorithms combine the results of basic machine learning (ML) classifiers (such as Naïve Bayes, K-Nearest Neighbors, and decision trees) with an ensemble technique (such as stacking, bagging, and voting) to produce optimal results (for example, accuracy). The performance of these algorithms depends on what basic ML classifiers are used and what ensemble technique is chosen.

Recently, we developed a method that selects the most appropriate combination of basic classifiers and an ensemble technique based on execution time and performance metrics (such as accuracy, etc.) [2]. However, this process requires a lot of computing power, which may not be available on IoT devices or near by devices (such as edge/fog devices). As a result, we proposed in [2] that the training (i.e., selection of basic classifiers and ensemble techniques) of our approach can be conducted on the cloud while real-time decisions can be made on the edge/fog nodes (Fig. 1). Fig. 1 illustrates our approach, in which the thing layer consists of IoT devices, the fog/edge layer comprises devices near the IoT devices, and the cloud layer comprises servers located on the cloud. Further, the three steps of our approach running on cloud and fog/edge layers are provided in Fig. 1.

978-1-6654-8353-7/22/$31.00 ©2022 IEEE

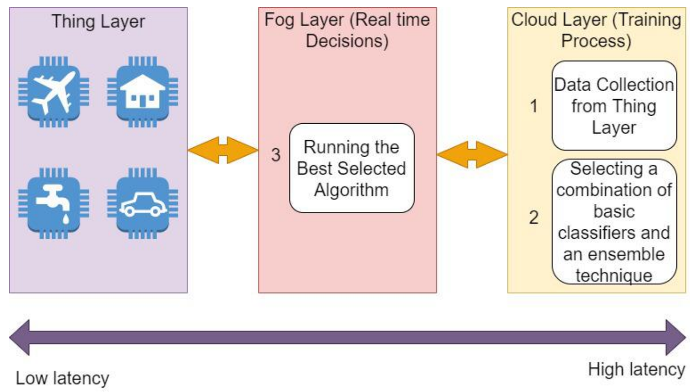

Fig. 1: Ensemble Machine Learning Algorithm on IoT

We considered denial of service, authentication and probe attacks in our work. In [2], we tested the above approach on a virtual machine installed on a laptop. However, in the training process, this approach took around two hours (after collecting the data) with a relatively limited set of basic classifiers and ensemble techniques (i.e., only 10 models). Therefore, in this paper, we demonstrate the use of the Fed4Fire GPULab to run our approach. GPULAB is a freely available Fed4Fire testbed which contains GPU and CPU resources to run resource intensive algorithms [3].

In this demo, we first demonstrate how our approach can be deployed on GPULAB. Using GPULAB to run our above training process with 10 models took around 10 minutes. Therefore, we run a larger number of models on GPULab (i.e., 30 models) and test their performance. The results show the applicability of our approach to detect attacks in IoT. This work is part of an NGIAtlantic project, which includes experiments on EU/US testbeds [4].

II. Running the proposed Edge Model on GPULAB

Cluster ID	#GPU	#CPU	CPU Memory (GB)	#Slaves	Used in our Exp
1	2	16	31.47	1	Yes
3	1	12	31.66	1	No
4	44	128	1072.77	4	Yes
5	8	80	536.24	2	No
6	16	96	1619.95	1	Yes
7	24	176	2158.91	2	No

TABLE I. GPULAB Resources [3]. Exp stands for Experiment

Resources at GPULAB are divided into Clusters from 1 to 7. Table I lists the total resources available in the GPULAB in terms of GPU and CPU cores, CPU memory and slaves. The availability of resources and clusters for an experiment at a time depends on the resources used by other experimenters at that time. Table I also shows the clusters we used in our experiments (Cluster 1, 4 and 6). At the time of our experiments, Cluster 1 had only 1 CPU and 8 GB of CPU memory, Cluster 4 had 6 GPUs, 12 CPUs and 32 GB of memory, and Cluster 6 had 12 CPUs and 32 GB of memory.

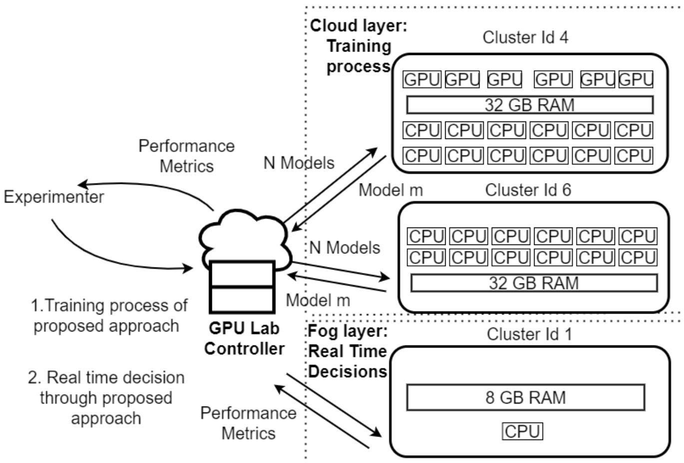

Fig. 2: Running the proposed approach on the GPULAB

To test our approach, we need a system with numerous resources for the cloud layer steps and a system with minimal resources for the fog/edge layer steps. Therefore, we run the cloud layer tasks at Cluster 4 and 6, and the fog/edge layer tasks at Cluster 1 (Fig. 2). An experimenter running our approach in Fig. 2 performs two tasks: (1) training ensemble models at the cloud layer and (2) making real-time decisions at the fog layer. For training, our approach containing N ensemble models (which contain a combination of basic classifiers and an ensemble technique) is tested with the NSL-KDD dataset, and the best model (model m) with respect to the execution time and performance is selected. This model is then used to make real-time decisions in the fog/edge layer, where Cluster 1 is used to run the model (see Fig. 2). In our experiments, N is 30 and all the other testing procedures are the same as given in [2].

III. Experimental Results and Demonstration

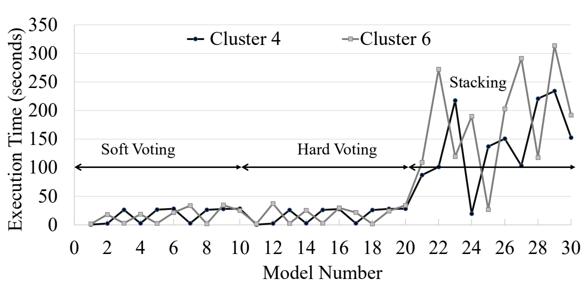

Fig. 3: Experimental Results with 30 ensemble models

We selected five basic classifiers for our experiments: (1) Naive Bayes, (2) KNN, (3) Decision Tree, (4) Logistic Regression, and (5) Random Forest, and three ensemble techniques: (1) soft voting, (2) hard voting, (3) stacking. Different models are created by combining the above classifiers and selecting one of the ensemble techniques. In the training process, all these models are tested on Clusters 4 and 6, as shown in Fig. 2. Fig. 3 shows this execution time. The total training process took less than 2000 seconds (i.e., $<$ 30 minutes) in both the clusters. Furthermore, we calculated accuracy, F-score, Precision, Recall, MCC, ROC, Kappa, and Errors. Our approach selected model 24 to be run on the fog layer based on the best performance metrics and shortest execution time. There are three basic classifiers in this model: Naive Bayes, Decision Tree, Logistic Regression, and the stacking ensemble model.

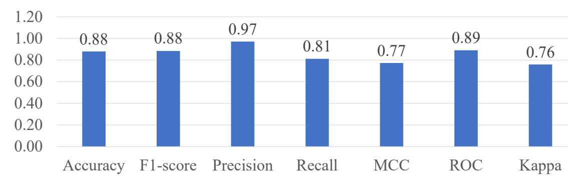

Fig. 4: Experimental Results with model 24

Model 24 was then tested on Cluster 1 (which contains minimal resources) and the execution time was 351 seconds. The values of performance metrics are given in Fig. 4.

This work shows an example of an edge-cloud computing model where high resource intensive tasks can be run on the cloud and the real time decisions could be run on the edge/fog layer which is closer to the thing layer.

Our demonstration shows first how to create an account on Fed4Fire, then demonstrates that how to experiment with our method on the GPULAB testbed with a variety of resources such as GPUs and CPUs, each with a specific amount of CPU memory. During our demonstration, we use Clusters 4 and 6 for the training process of our approach, and Cluster 1 for real-time decision-making as described in the previous section. Following that, we demonstrate how to calculate the results provided in this paper. The code implementing our approach and all the configurations is available at [5].

Acknowledgement

This research was funded by the EU H2020 NGIAtlantic project under agreement No. OC3-292.

References

[1] Q. Abu Al-Haija and A. Al-Badawi, "Attack-aware iot network traffic routing leveraging ensemble learning," *Sensors*, vol. 22, no. 1, 2022.

[2] V. Tomer and S. Sharma, "Detecting iot attacks using an ensemble machine learning model," *Future Internet*, vol. 14, no. 4, 2022. [Online]. Available: https://www.mdpi.com/1999-5903/14/4/102

[3] "Gpulab fed4fire eu testbed." [Online]. Available: https://www.fed4fire.eu/testbeds/gpulab/

[4] S. Sharma, S. Urumkar, G. Fontanesi, B. Ramamurthy, and A. Nag, "Future wireless networking experiments escaping simulations," *Future Internet*, vol. 14, no. 4, 2022. [Online]. Available: https://www.mdpi.com/1999-5903/14/4/120

[5] "Our code with all configuration." [Online]. Available: https://github.com/VikasTomar32/LANMAN