Intelligent Monitoring of IoT Devices using Neural Networks

Ashima Chawla
Software Research Institute
Athlone Institute of Technology
Athlone, Ireland
ashima.chawla@ericsson.com

Pradeep Babu
Cloud Pipeline Engineering
Workday
Dublin, Ireland
pradeep.babu@workday.com

Trushnesh Gawande
Application Development Platform
Ericsson
Athlone, Ireland
trushnesh.gawande@ericsson.com

Erik Aumayr
Network Management Lab
Ericsson
Athlone, Ireland
erik.aumayr@ericsson.com

Paul Jacob
Software Research Institute
Athlone Institute of Technology
Athlone, Ireland
pjacob@ait.ie

Sheila Fallon
Software Research Institute
Athlone Institute of Technology
Athlone, Ireland
sheilafallon@ait.ie

***Abstract*—The Internet of Things (IoT) has seen expeditious growth in recent times with 7 billion connected devices in 2020, thus leading to the vital importance of real-time monitoring of IoT devices. Through this paper, we demonstrate the idea of building a cloud-native application to monitor smart home devices. The application intends to provide valuable performance metrics from the perspective of end-users and react to anomalies in real-time. In this demo paper, we conduct the demonstration using Autoencoder (an unsupervised technique) based Deep Neural Networks (DNNs) to learn the normal operating conditions of power consumption of smart devices. When an anomaly is detected, the DNNs take proactive action and send appropriate commands back to the device. In addition, the users are provided with a real-time graphical representation of power consumption. This will help to save electricity on a domestic as well as industrial level. Finally, we discuss the future prospects of monitoring IoT devices.**

***Index Terms*—IoT devices, Deep Learning, Microservices, cloud-native application**

I. Introduction

In recent years, there has been a tremendous growth of modern communication, and the proliferation of IoT has been widely increasing in societies throughout the world. As per the market survey [1], the number of connected IoT devices will reach 26.9 billion by 2026, which amounts to a 13% compound annual growth rate.

The importance of intelligent monitoring [2] of IoT devices is imperative for consumer oriented scenarios. Even though the IoT devices can be connected with Artificial Intelligence, the challenge is to monitor them in real-time and handle the increasing number of connected devices.

In this demo paper, we build a cloud-native application that is orchestrated by Kubernetes which makes the application scalable to address increasing number. Additionally, Kubernetes provides high availability, and self-healing features. This solution can be deployed on any cloud (public or private) that provides Kubernetes as a Service (KaaS). Our research enhances the previous work with an intelligent monitoring framework which detects anomalies and takes proactive actions for IoT devices in real-time.

The paper is organized as follows. Section II outlines the cloud-native microservice solution. Followed by the model architecture, components, experimental settings environment and demonstration. In Section III, we discuss the future prospects and section IV concludes the paper.

II. Cloud-Native Microservice Solution

We demonstrate our approach by leveraging the concept of a microservices architecture to build a cloud-native application to intelligently monitor real-world smart home devices. As compared to traditional monolithic solutions, our proposed microservice based solution makes it possible to scale modules independently when more IoT devices are connected in the future.

Various applications [3], [4], [5] have been deployed to monitor and control IoT devices but they lack the ability to integrate proactive deep learning solution with a scalable cloud deployment. We illustrate our demonstration approach below.

A. Model Architecture

Fig. 1 illustrates the workflow of the proposed real-time streaming solution. The microservices Stream Data Processor, Deep Learning Anomaly Detection and PM Metric Processor are custom built while Kafka, Zookeeper and Prometheus microservices are open source real-time processing solutions used for this demonstration. All microservices are packaged using helm and deployed on a Kubernetes cluster. Each microservice is self-isolated from each other that provides the ability to optimize and scale individually.

B. Components

- **Stream Data Processor (SDP)**: Stream Data Processor is real-time event processing solution that processes and

Irish Research Council- EBPPG/2019/76

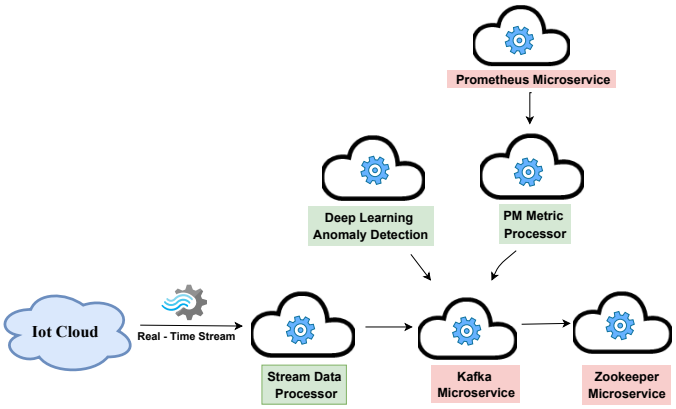

Fig. 1: Home Automation Workflow. The microservices Stream Data Processor, Deep Learning Anomaly Detection and PM Metric Processor (green) are custom built while Kafka, Zookeeper and Prometheus microservices (red) are open source real-time processing solutions used for this demonstration.

decodes the IoT streams. The processed data is then later encoded as JSON strings and forwarded to a Kafka topic.

- **Kafka Microservice**: Kafka provides a distributed framework for streaming solutions. Here, we use Kafka as a messaging system to send stream data between producer and consumers. A dedicated topic "IoTDevices" is created for IoT streams. SDP forwards processed IoT stream to a Kafka topic, consumers can subscribe to the "IoTDevices" topic and pull IoT device data.
- **Zookeeper Microservice**: Kafka is dependent on Zookeeper where all the metadata (e.g. Kafka Topics, clusters etc.) information is stored.
- **Deep Learning Anomaly Detection**: This microservice subscribes to the Kafka "IoTDevices" topic and polls data. Autoencoders as shown in Fig. 2 are a type of neural network [6] trained to learn a compressed representation of the input data and to reconstruct the input from this representation. Anomaly detection is then carried out by identifying instances which can not be accurately reconstructed, and these are identified as anomalous instances. For inference, the real-time test data is streamed using the Kafka microservice to the deep learning model. Here KafkaGroupIODataset [7] reads the test data in batches and has been configured to predict the anomalies in real-time.
- **PM Metric Processor**: The PM metric processor is one of the subscribers of the Kafka "IoTDevices" topic. This decodes the JSON encoded data and performs correlation/aggregation and generates valuable metrics for visualization.
- **Prometheus Microservice**: Prometheus is a monitoring and alerting tool. Prometheus scrapes IoT performance metrics generated by PM Metric Processor and generates graphs and alerts.

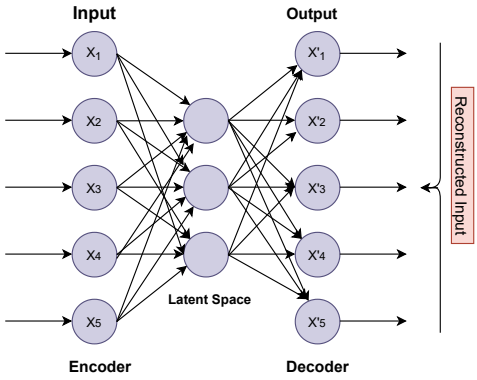

Fig. 2: Autoencoder

C. Environment

All experiments have been conducted on computational machine which includes Intel® Core™ i7-8650U CPU @ 1.90GHz × 8, 25.8 GiB memory, Ubuntu 18.04.4 LTS operating system. The Keras Python library 2.3.1 is used for running on top of a source build of TensorFlow 2.1.0, Docker engine version 19.03, Kubernetes v1.18.

D. Demonstration

The demonstration is based on the microservice workflow as mentioned in Fig.1.

- Firstly, the stream data processor processes IoT cloud devices data, such as a device list, power consumption, switch status and timestamp related information. For the purpose of this demonstration we have considered power consumption of "**TV**" (TV with a smart socket) as the feature column to train the neural network.

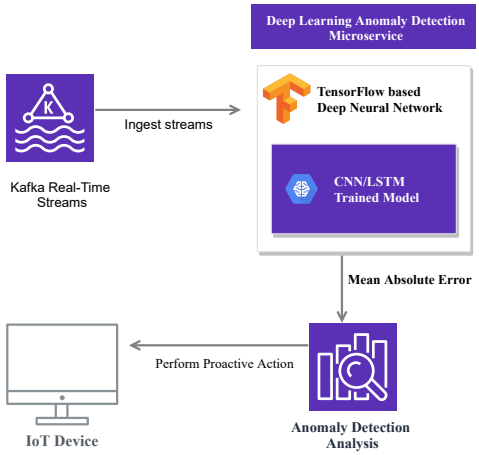

Fig. 3: Anomaly Detection Workflow

- The time series Convolutional Neural Network-Long Short Term Memory (CNN-LSTM) inspired by [8] is a Keras [9] based autoencoder model. Here, 1D convnets (filter size=32 , kernel size=15, padding ='same') have been used as a pre-processing step to make the time

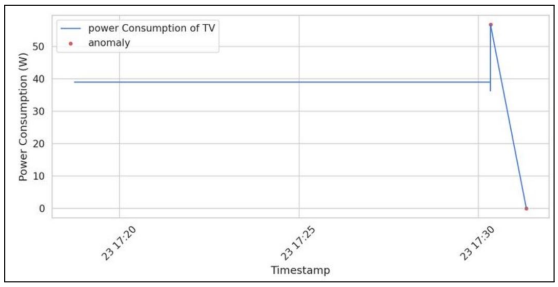

Fig. 4: Real-Time IoT Devices Analytics using Microservices

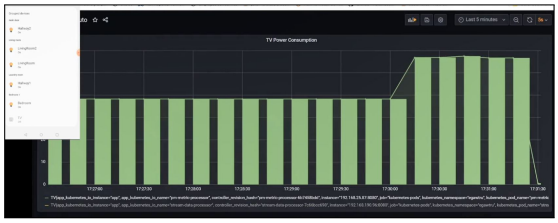

Fig. 5: Real-Time IoT Devices Anomaly Detection using DNNs

series sequence smaller, resulting in a faster training. Thus, the CNN layer extracts the higher level local features, which are then passed on to the LSTM as input. LSTMs are special types of Recurrent Neural Networks (RNN) which are typically used to learn time series sequential data using Backpropagation Through Time (BPTT) algorithm.

The LSTM (120 neuron units) encoder part feeds in the input and then learns the compressed vector state into the latent space. Later, then decoder uses the bottleneck (latent space - reduced dimension state) information to decode the information to reconstruct the input to its original form. The CNN-LSTM model(dropout as regularization) learns the normal pattern and detects the anomalies based on the the Mean Absolute Error (MAE, Eq. 1). MAE calculates the measure of errors between the true value x_i and predicted x'_i values of a set of n. Here the reconstruction error has been used to identify previously unseen anomalies as shown in Fig. 3.

$$MAE = (\frac{1}{n}) \sum_{i=1}^{n} |x'_i - x_i| \quad (1)$$

- Once the anomalous power consumption is detected (as shown in Fig. 4) a command is then sent back to the IoT device to perform some proactive action. In this scenario, a "**turn off**" command is sent back to the **TV** smart socket using a REST API.
- In parallel, Fig. 5 shows the real-time analysis of the other IoT devices (smart home appliances e.g. sockets, bulbs etc.) using the Prometheus microservice in Grafana [10]. Further work is in progress to integrate and monitor additional IoT devices.

III. Future

With the evolution of 5G and IoT, it is predicted that more and more devices will be connected to each other in the future. With billions of connected devices, it will be difficult to manually monitor all the devices. Hence, a proactive approach is needed that requires automated real-time monitoring and detection of anomalies in IoT devices. We further plan to expand our approach with multivariate anomaly detection for IoT devices using interpretable deep neural network models.

IV. Summary

In this demo paper, we introduced the idea of using cloud-native application for intelligently monitoring the IoT devices in real-time. Our solution detects the anomalies based on CNN-LSTM based neural network model in an unsupervised way and performs proactive actions. This automated solution can save a lot of manual monitoring and troubleshooting, with a resulting reduction in costs and resources. Like any other cloud-native application, security and complexity are the key concerns which still needs to be addressed.

V. Acknowledgement

This work is funded by the Irish Research Council Enterprise Partnership Scheme Postgraduate Scholarship 2020 under Project EBPPG/2019/76.

References

[1] IoT connections outlook, https://www.ericsson.com/en/mobility-report/dataforecasts/iot-connections-outlook/, Last accessed on January 28, 2021.
[2] S. Kalaivanan and S. Manoharan, Monitoring and Controlling of Smart Homes using IoT and Low Power Wireless Technology, vol. 9(31), Indian Journal of Science and Technology, 2016.
[3] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu· and J. Qin, A survey on application of machine learning for Internet of Things, vol. 9, International Journal of Machine Learning and Cybernetics , 2018, p. 1399–1417.
[4] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan and A. Sadeghi, "DÏoT: A Federated Self-learning Anomaly Detection System for IoT," 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 2019, pp. 756-767, doi: 10.1109/ICDCS.2019.00080.
[5] C. Lai, F. Boi, A. Buschettu and R. Caboni, "IoT and Microservice Architecture for Multimobility in a Smart City," 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud), Istanbul, Turkey, 2019, pp. 238-242, doi: 10.1109/FiCloud.2019.00040.
[6] A. Chawla, P. Jacob, S. Feghhi, D. Rughwani, S. v. d. Meer and S. Fallon, Interpretable Unsupervised Anomaly Detection For RAN Cell Trace Analysis, Izmir: 16th International Conference on Network and Service Management (CNSM), 2020, pp. 1-5.
[7] Tensorflow, https://www.tensorflow.org/io/tutorials/kafka, Last accessed on December 27, 2020.
[8] Chawla, A., Lee, B., Fallon, S., and Jacob, P. (2018, September). Host based intrusion detection system with combined CNN/RNN model. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 149-158). Springer, Cham.
[9] Keras Home Page, https://keras.io/, Last accessed on December 27, 2020
[10] Grafana Labs, Home Page, https://grafana.com/, Last accessed on December 27, 2020.