ICN Research Group
Internet-Draft
Intended status: Informational
Expires: September 12, 2019

J.K.Choi
N.K.Kim
J.S.Han
M.K.Kim
KAIST
G.M.Lee
Liverpool John Moores University

March 11, 2019

Requirements and Challenges for User-level Service Managements of IoT Network by utilizing Artificial Intelligence

draft-choi-icnrg-aiot-00

Abstract

This document describes the requirements and challenges to employ artificial intelligence (AI) into the constraint Internet of Things (IoT) service environment for embedding intelligence and increasing efficiency.

The IoT service environment includes heterogeneous and multiple IoT devices and systems that work together in a cooperative and intelligent way to manage homes, buildings, and complex autonomous systems. Therefore, it is becoming very essential to integrate IoT and AI technologies to increase the synergy between them. However, there are several limitations to achieve AI enabled IoT as the availability of IoT devices is not always high, and IoT networks cannot guarantee a certain level of performance in real-time applications due to resource constraints.

This document intends to present a right direction to empower AI in IoT for learning and analyzing the usage behaviors of IoT devices/systems and human behaviors based on previous records and experiences. With AI enabled IoT, the IoT service environment can be intelligently managed in order to compensate for the unexpected performance degradation often caused by abnormal situations.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The document explains the effects of applying artificial intelligence/machine learning (AI/ML) algorithms in the Internet of Thing (IoT) service environments.

IoT applications will be deployed in heterogeneous and different areas such as the energy, transportation, automation and manufacturing industries as well as the information and communication technology (ICT) industry. Many IoT sensors and devices can connect to an IoT service environment where IoT objects cannot interoperate with each other and can interact with different applications. The IoT service may not run in a single administrative domain. If market demand exists, the cross-domain service scenarios for IoT applications could be widely deployed. Future IoT applications occur at multiple domains of heterogeneity with various time scales.

The IoT service requirements for common architectures and public APIs poses some challenges to the underlying service environment and networking technologies. Some IoT applications require significant security and privacy as well as significant resource and time constraints. These mission-critical applications can be separated from many common IoT applications that current technology may not provide. It means that IoT service requirements are difficult to classify common requirements and functional requirements depending on IoT service scenario.

Recently, artificial intelligence technologies can help the context-aware IoT service scenarios apply rule-based knowledge accumulation. The IoT service assumes that many sensing devices are connected to single or multiple IoT network domains. Each sensor sends small packets to the IoT servers periodically or non-periodically.

Detection data contains periodic status information that monitors whether the system is in a normal state or not. In some cases, alert information is included for quick processing. Most IoT applications can operate in two modes. One is a simple monitoring mode and the other is an abnormal mode for rapid processing. In a simple monitoring phase, the IoT device periodically sends sensing data to the server. If the measured data is outside the normal range, the IoT service can change the operating mode to an abnormal phase and activate future probes. Alarm conditions should be promptly notified to responsible persons. For mission-critical applications, reliable communication with robust QoS requirements in terms of error and latency is required.

Periodic data accumulation from IoT devices is cumbersome. Under normal conditions, the IoT data is simply accumulated without further action. In an unusual situation, incoming IoT data can cause an urgent action to notify the administrator of the problem. Streaming data traffic from thousands of IoT devices is annoying to store in the database because it is not easy to extract unidentified or future incidents. Only a significant portion of the incoming data stream can be stored in a real-time database that is time-sensitive and capable of rapid query processing. A combination of different IoT detection data, including location, time, and status, allows you to sort and categorize a portion of streaming data when an additional inspection is required, and perform real-time processing. One of the missions of the IoT database is to be able to extract preliminary symptoms of unexpected accidents from a large amount of streaming data.

If some transmitted data is important to invoke the corresponding action, there are some questions about whether the incoming data is correct. If the incoming data contains accurate and time-critical events, appropriate real-time control and management can be performed. However, if the incoming data is inaccurate or intentionally corrupted, additional accidents may occur. In these cases, incoming data can trigger to initiate additional inspections to protect against future unacceptable situations. But, if time-critical data is missed due to errors in the sensing devices and the delivery protocol, there is no reason to configure IoT networks and devices at a high cost.

It is not easy to analyze data collected through IoT devices installed to monitor complex IoT service environments. If the sensor malfunctions, the data of the sensor cannot be trusted. Additional investigation should be done if abnormal status from specific sensors is collected. The data of the redundant sensor installed in the same area should be received or combined with other sensor information adjacent to the sensor to determine the abnormal state.

For sensors installed in a specific area, sensing records will remain for a certain period of time. IoT service operators can look at the operational history of the sensor for a period of time to determine what problems were encountered when data was collected. When an abnormal situation occurs, IoT sensor should investigate whether it noticed normal operations and notified the IoT service operator. If the abnormal situation is not properly detected, the operator should analyze whether it was caused by malfunction of the IoT sensor or other reasons.

In the IoT service environment, it is possible to analyze the situation accurately by applying recent artificial intelligence and machine learning technologies. If there is an operational record of the past, it is possible to determine when an abnormal situation arises. Most problems are likely to be repeated, so if the past learning experience is accumulated, the anomaly of IoT services can be easily and immediately identified. In addition, when information gathered from various sensors is synthesized, it is possible to accurately determine whether abnormal situations have occurred.

Various types of IoT sensors are installed with certain purposes. It expects that all the IoT sensors intend to monitor the occurrence of special abnormal situations in advance. Therefore, it should be set in advance what actions are required when a specific anomaly occurs. The appropriate work is performed on the abnormal situation according to the procedure, predefined by the human. By using artificial intelligence and machine learning algorithms, the appropriate actions are taken when an abnormal situation is detected from various IoT sensors.

2. Challenging Issues of IoT network

This section describes the challenging issues of data sensing, collection, transfer, and intelligent decision from untrusted data quality and unexpected situations of IoT service environments.

2.1. Untrusted and incorrect IoT devices

IoT traffic is similar to traditional Internet traffic with small packet sizes. Mobile IoT traffic can cause some errors and delays because wireless links are unstable and signal strength may be degraded with device mobility. If the signal strength of the IoT device with a power limit is not so strong, the reception quality of the IoT server may not be sufficient to obtain the measurement data.

For mission-critical applications, such as smart-grid and factory-automation, expensive IoT sensors with self-rechargeable batteries and redundant hardware logic may be required. However, unexpected abnormal situations may occur due to sensor malfunctions. There are trade-offs between implementation cost and efficiency for cost-effective IoT services. When smart-grid and factory-automation applications are equipped with IoT devices, the acceptable quality from IoT solutions can be required. Sometimes, expensive and duplicated IoT solutions may be needed.

2.2. Traffic burstiness of IoT network

IoT traffic includes two types of traffic characteristic: periodic with small packet sizes and bursty with high bandwidth. Under normal conditions, the IoT traffic periodically transmits status information with a small bandwidth, several kilobits/sec. However, in an abnormal state, IoT devices need a high bandwidth, up to several tens of megabits/sec, in order to identify actual events and investigate accurate status information. In addition, traffic volume can explosively increase in response to emergencies. For example, in the case of smart-grid application, the bandwidth of several kilobits/sec is usually used, and when an urgent situation occurs, a broadband channel is required up to several tens of megabits/sec.

The other traffic can be integrated at an IoT network to increase bandwidth efficiency. If an emergency situation occurs in the IoT service, IoT traffic volumes suddenly increase, in which case network processing capacity may be not sufficient. If the IoT service is integrated with voice and video applications, the problem can become more complex. As time goes by, traffic congestion and bottlenecks are frequent in some areas. In addition, if an existing service policy changes (for example, prioritizing certain traffic or suddenly changing the route), other unexpected problems may be encountered. Various congestion control and load balancing algorithms with the help of artificial intelligence can be applied to handle time-varying traffic on a network.

Until now, much research has been done on traffic variability in an integrated network service environment. All networks have their own traffic characteristics, depending on geographical area, number of subscribers, subscribers' preferences, and types of applications used. In the case of IoT traffic, the normal bandwidth is very small. If the IoT traffic volume increases abruptly in an abnormal situation, the network may suffer unacceptable delay and loss. If emergency situations detected by IoT networks occur in a smart grid or intelligent transportation system, the processing power of the IoT network alone cannot solve the problem and the help of existing network resources is inevitable.

2.3. Management overheads of heterogeneous IoT sensors

Traffic management in an integrated network environment is not easy. In order to operate the network steadily, a network operator has its own know-hows and experiences. If there are plenty of network resources, it is easy to set up a bypass route even if network failure or congestion occurs in a specific area. For operating network steadily, network resources may be designed to be over-provisioned in order to cope with various possible outages. A network operator predicts the amount of traffic generated by the corresponding equipment and grasps to what extent a transmission bandwidth is required. If traffic fluctuation is very severe, the network operator can allocate network resources in advance. In case of frequent failures or severe traffic fluctuation, some network resources are separated in order not to affect normal traffic.

More than a billion IoT devices are expected to connect to smartphones, tablets, wearables, and vehicles. Therefore, IoT services are targeted at mobile applications. In particular, intelligent transportation systems need the help of IoT technology to provide traffic monitoring and prevent public or private traffic accidents. IoT technology can play an important role in reducing traffic congestion, saving people's travel time and costs, and providing a pleasant journey.

The IoT service has troublesome administrative problems to configure an IoT network which consists of IoT servers, gateways, and many sensing devices. The small-sized but large-numbered IoT devices may incur administrative overhead since all the IoT devices should be initialized and the bootstrapping information of IoT resources should be loaded into the IoT service environments. Whenever some IoT devices are newly added and some devices have to be removed, the dynamic reconfiguration of IoT resources is essential. In addition, the IoT device's preinstalled software should be regularly inspected and upgraded according to its version. Frequent upgrades and changes to some IoT devices may require autonomic management and bootstrapping techniques.

Network management generally assumes that all network resources operate reliably with acceptable quality. In most failure situations, the network operator decides to switch to a redundant backup device or bypass the failed communication path. If some IoT devices are not stable, duplicate IoT devices can be installed for the same purpose. If IoT resources are not duplicated, various mechanisms are needed to reduce the damage. Therefore, it is necessary to prioritize the management tasks to be performed first when an abnormality occurs in the IoT service environment. However, managing duplicate networks can cause another problem. If two IoT devices are running at the same time, the recipient can get redundant information. If two or more unusual situations occur at the same time, it is difficult to solve the problem since tasks for urgent processing should be distinguished from tasks that can be performed over time.

In addition, the operations manager's mistakes or misunderstanding of problem situations can lead to other unexpected complications. Therefore, artificial intelligence technologies can help what kind of network management work is required when an unexpected complicated situation occurs even though a procedure for an abnormal situation is already prepared.

3. Overview of AI/ML-based IoT services

In this section, successful applications of artificial intelligence in IoT domains are provided. The common property of IoT applications and services is that they require fast analytics rather than later analytics with piled data. Recently, neural-network-based artificial intelligence technologies are widely used across many IoT applications.

Simple IoT applications include dynamic contexts that share common features among social relations at the same administration domain. IoT devices in the same domain can provide their service contexts to the IoT server. When a dynamic change occurs in an IoT service context, the IoT device needs real-time processing to activate urgent events, alert notifications, update, and reconnect contexts. The IoT service must support real-time interactions between the IoT device and the system in the same domain. The IoT service contexts must be shared between physical objects and social members in the same domain as well.

Artificial intelligence technologies have been shown promising in many areas, including IoT. For example, contextual information for a car-sharing business must interact with customers, car owners, and car sharing providers. All entities in the value chain of a car sharing business must share the corresponding situation to pick up, board, and return shared cars. Communication networks and interactive information, including registration and payment, can be shared tightly among the entities. Home IoT service environment can be equipped with sensors for theft detection, door lock, temperature, fire detection, gas detection, short circuit, air condition to name a few. Office IoT service environments, including buildings such as shopping centers and bus/airport terminals, have their own sensors, including alarm sensors. When an alarm signal is detected by the sensor, the physical position and occurrence time of the sensor is determined in advance. All signals from various sensors are analyzed comprehensively to make the right decision. If some sensors frequently malfunction, the situation can be grasped more accurately by analyzing the information of the adjacent sensor. In particular, when installing multiple sensors in a particular building (e.g., surveillance camera, location monitoring, temperature, etc.), a much wider range of sensors can be used when utilizing artificial intelligence and machine learning technologies.

(Smart home) Smart home concept span over multiple IoT applications, health, energy, entertainment, education, etc. It involves voice recognition, natural language processing, image-based object recognition, appliance management, and many more artificial intelligence technologies integrated with IoT. Smart connected-devices monitor the house to provide better control over home supplies and expenses. The energy consumption and efficiency of home appliances are monitored and analyzed with deep learning based technologies, such as artificial neural network, long-short-term-memory, etc.

(Smart city) Smart city, as well, contains multiple IoT domains, transportation, infrastructure, energy, agriculture, etc. Since heterogeneous data from different domains are gathered in smart cities, various artificial intelligence approaches are studied in smart-city application. Public transportation behaviors and crowd movements patterns are important issues, and they are often dealt with neural network based methods, long-short-term-memory and convolutional neural network.

(Smart energy) As two-way communication energy infrastructure is deployed, smart grid has become a big IoT application, which requires intelligent data processing. The traditional energy providers are highly interested in recognizing local energy consumption patterns and forecasting the needs in order to make appropriate decisions on real-time. Moreover, the energy consumers, as well, want analyzed information on their own energy consumption behaviors. Recently, many works on energy consumption prediction, energy flexibility analysis, etc. are actively ongoing. Most works are based on the latest deep learning technologies, such as multi-layered-perceptron, recurrent neural network, long-short-term-memory, autoencoder, etc.

(Smart transportation) The intelligent transportation system is another source of big data in IoT domains. Many use cases, such as traffic flow and congestion prediction, traffic sign recognition, vehicle intrusion detection, etc., have been studied. Moreover, a lot of advanced artificial intelligence technologies are required in autonomous and smart vehicles, which require many intelligent sub-tasks, such as pedestrian's detection, obstacle avoidance, etc.

(Smart healthcare) IoT and artificial intelligence are integrated into the healthcare and wellbeing domain as well. By analyzing food images with convolutional neural network on mobile devices, dietary intakes can be measured. With voice signal captured from sensor devices, voice pathologies can be detected. Moreover, recurrent neural network and long-short-term-memory technologies are actively being studied for early diagnosis and prediction of diseases with time series medical data.

(Smart agriculture) To manage a vast area of land, IoT and artificial intelligence technologies are recently used in agriculture domains. Deep neural network and convolutional neural network are utilized for crop detection or classification and disease recognition in the plants. Moreover, for automatic farming with autonomous machine operation, obstacle avoidance, fruit location, and many more sub-tasks are handled with advanced artificial intelligence technologies.

4. Requirements for AI/ML-based IoT services

(to be included)

4.1. Requirements for AI/ML-based IoT data collection and delivery

(to be included)

4.2. Requirements for intelligent and context-aware IoT services

(to be included)

5. State of arts of the artificial intelligence/machine learning technologies for IoT services

In this section, well-known machine learning and artificial intelligence technologies applicable to IoT applications are reviewed.

5.1. Machine learning and artificial intelligence technologies review

The classical machine learning models can be divided into three types, supervised, unsupervised, and reinforcement learnings. Therefore, in this subsection, machine learning and artificial intelligence technology reviews are done in four different categories: supervised, unsupervised, reinforcement, and neural-network-based.

5.1.1. Supervised learning for IoT

Supervised learning is a task-based type of machine learning, which approximates function describing the relationship and causality between input and output data. Therefore, the input data needs to be clearly defined with proper output data since supervised learning models learn explicitly from direct feedback.

(K-Nearest Neighbor) Given a new data point in K-Nearest Neighbor (KNN) classifier, it is classified according to its K number of the closest data points in the training set. To find the K nearest neighbors of the new data point, it needs to use a distance metric which can affect classifier performance, such as Euclidean, Mahalanobis or Hamming. One limitation of KNN in applying for IoT network is that it is unscalable to large datasets because it requires the entire training dataset to classify a newly incoming data. However, KNN required less processing power capability compared to other complex learning methods.

(Naive Bayes) Given a new data point in Naive Bayes classifiers, it is classified based on Bayes' theorem with the "naive" assumption of independence between the features. Since Naive Bayes classifiers don't need a large number of data points to be trained, they can deal with high-dimensional data points. Therefore, they are fast and highly scalable. However, since its "naive" assumptions are somewhat strong, a certain level of prior knowledge on the dataset is required.

(Support Vector Machine) Support Vector Machine (SVM) is a binary and non-probabilistic classifier which finds the hyperplane maximizing the margin between the classes of the training dataset. SVM has been the most pervasive machine learning technology until the study on neural network technologies are advanced recently. However, SVM still has advantages over neural network based and probabilistic approaches in terms of memory usage and capability to deal with high-dimensional data. In this manner, SVM can be used for IoT applications with severe data storage constraint.

(Regression) Regression is a method for approximating the relationships of the dependent variable, which is being estimated, with the independent variables, which are used for the estimation. Therefore, this method is widely used for forecasting and inferring causal relationships between input data and output data in time-sensitive IoT application.

(Random Forests) In random forests, instead of training a single decision tree, a group of trees is trained. Each tree is trained on a subset of the training set using a randomly chosen subset of M input variables. Random forests considering various tree structures have very high accuracy, so it can be utilized in the accuracy-critical IoT applications.

5.1.2. Unsupervised learning for IoT

Unsupervised learning is a data-driven type of machine learning which finds hidden structure in unlabeled dataset without feedback during the learning process. Unlike supervised learning, unsupervised learning focuses on discovering patterns in the data distributions and gaining insights from them.

(K-means clustering) K-means clustering aims to assign observations into K number of clusters in which each observation belongs to the cluster having the most similarities. The measure of similarity is the distance between K cluster centers and each observation. K-means is a very fast and highly scalable clustering algorithm, so it can be used for IoT applications with real-time processing requirements such as smart transportation.

(Density-based spatial clustering of applications with noise) Density-Based approach to Spatial Clustering of Applications with Noise (DBSCAN) is a method that clusters dataset based on the density of its data samples. In this model, dense regions which include data samples with many close neighbors are considered as clusters, and data samples in low-density regions are classified as outliers [Kriegal]. Since this method is robust to outliers, DBSCAN is efficient data clustering method for IoT network environments with untrusted big datasets in practice.

5.1.3. Reinforcement learning for IoT

Reinforcement learning is a reactive type of machine learning that learn a series of actions in a given set of possible states, actions, and rewards or penalties. It can be seen as the exploring decision-making process and choosing the action series with the most reward or the least penalty which can be cost, priority, time to name a few. Reinforcement learning can be helpful for selecting action of IoT device by providing a guideline.

(Q-learning) Q-Learning is a model-free, off-policy reinforcement learning algorithm based on the well-known Bellman Equation. The goal is to learn an action-selection policy maximizing the Q-value, which tells an agent what action to take. It can be used for IoT device to determine which action it should take according to conditions.

(State-Action-Reward-State-Action) Though State-Action-Reward-State-Action (SARSA) is a much similar algorithm to Q-learning, the main difference is that it is an on-policy algorithm in which agent interacts with the environment and updates the policy based on actions taken. It means that the Q-value is updated by an action performed by the current policy instead of the greed policy that maximizes Q-value. In this perspective, it is relevant when an action of one IoT device will greatly influence the condition of the environment.

(Deep Q Network) Deep Q network (DQN) is developed to solve the exploration problem for unseen states. In the case of Q-learning, the agent is not capable of estimating value for unseen states. To handle this generality problem, DQN leverages neural network technology. As a variation of the classic Q-Learning algorithm, DQN utilizes a deep convolutional neural net architecture for Q-function approximation. In real environments not all possible states and conditions are not able to be observed. Therefore, DQN is more relevant than Q-learning or SARSA in real applications such as IoT. Since DQN could be used within only discrete action space, it can be utilized for traffic routing in the IoT network.

(Deep Deterministic Policy Gradient) DQN has solved generality and exploration problem of the unseen or rare states. Deep Deterministic Policy Gradient (DDPG) takes DQN into the continuous action domain. DDPG is a deterministic policy gradient based actor-critic, model-free algorithm. The actor decides the best action for each state and critic is used to evaluate the policy, the chosen action set. In IoT applications, DDPG can be utilized for the tasks that require controlled in continuous action spaces, such as energy-efficient temperature control, computation offloading, network traffic scheduling, etc.

5.1.4. Neural Network based algorithms for IoT

(Recurrent Neural Network) Recurrent Neural Network (RNN) is a discriminative type of supervised learning model that takes serial or time-series input data. RNN is specifically developed to address issue of time dependency of sequential time-series input data. It processes sequences of data through internal memory, and it is useful in IoT applications with time-dependent data, such as identifying time-dependent patterns of sensor data, estimating consumption behavior over time, etc.

(Long Short Term Memory) As an extension of RNN, Long Short Term Memory (LSTM) is a discriminative type of supervised learning model that is specialized for serial or time-series input data as well [Hochreiter]. The main difference of LSTM from RNN is that it utilizes the concept of gates. It actively controls forget gates to prevent the long term time dependency from waning. Therefore, compared to RNN, it is more suitable for data with long time relationship and IoT applications requiring analysis on the long lag of dependency, such as activity recognition, disaster prediction, to name a few [Chung].

(Convolutional Neural Network) Convolutional neural network (CNN) is a discriminative type of supervised learning model. It is developed specifically for processing 2-dimensional image data by considering local connectivity, but now generally used for multidimensional data such as multi channel sound signals, IoT sensor values, etc. As in CNN neurons are connected only to a small subset of the input and share weight parameters, CNN is much more sparse compared to fully connected network. However, it needs a large training dataset, especially for visual tasks. In CNN, a new activation function for neural network, Rectified Linear Unit (ReLU), was proposed, which accelerates training time without affecting the generalization of the network [Krizhevsky]. In IoT domains, it is often used for detection tasks that require some visual analysis.

(Variational Autoencoder) Autoencoder (AE) is a generative type of unsupervised learning model. AE is trained to generate output to reconstruct input data, thus it has the same number of input and output units. It is suitable for feature extraction and dimensionality reduction. Because of its behavior to reconstructing the input data at the output layer, it is often used for machinery fault diagnosis in IoT applications. The most popular type of AE, Variational Autoencoder (VAE) is a generative type of semi-supervised learning model. Its assumptions on the structure of the data are weak enough for real applications and its training process through backpropagation is fast [Doersch]. Therefore, VAE is suitable in IoT applications where data tends to be diverse and scarce.

(Generative Adversarial Network) Generative Adversarial Network (GAN) is a hybrid type of semi-supervised learning model which contain two neural networks, namely the generative and discriminative networks [Goodfellow]. The generator is trained to learn the data distribution from a training dataset in order to generate new data which can deceive the latter network, so-called the discriminator. Then, the discriminator learns to discriminate the generated data from the real data. In IoT applications, GAN can be used in situations when something needs to be generated from the available data, such as localization, way-finding, and data type conversion.

5.2. Technologies for lightweight and real-time intelligence

As the era of IoT has come, some sort of light-weight intelligence is needed to support smart objects. Prior to the era of IoT, most of the works on learning did not consider resource-constrained environments. Especially, deep learning models require many resources such as processing power, memory, stable power source, etc. However, it has been recently shown that the parameters of the deep learning models contain redundant information, so that some parts of them can be delicately removed to reduce complexity without much degradation of performance [Ba], [Denil]. In this section, the technologies to achieve real-time and serverless learning in IoT environments are introduced.

(network compression) Network compression is a method to convert a dense network into a sparse one. With this technology the network can be reduced in its size and complexity. By pruning irrelevant parts or sharing redundant parameters, the storage and computational requirements can be decreased [Han]. After pruning, the performance of the network is examined and the pruning process is repeated until the performance reaches the minimum requirements for the specific applications and use cases. As many parameters are removed or shared, the memory required is reduced, as well as computational burden and energy. Especially as most energy in neural network is used to access memory, the consumed energy dramatically drops. Although its main limitation is that there is not a general solution to compress all kinds of network, but it rather depends on the characteristics of each network. However, network compression is still the most widespread method to make deep learning technologies to be lightweight and IoT-friendly.

(approximate computing) Approximate computing is an approach to support deep learning in smart devices [Venkataramani], [Moons]. It is based on the facts that the results of deep learning do not need to be exact in many IoT applications but still valid if the results are in an acceptable range. By integrating approximate computing into deep learning, not only the execution time but also the energy consumption are reduced [Mohammadi]. Based on the optimal trade-off between accuracy and run-time or energy consumption, the network can be adjustably approximated. The network approximate technology can be well-used in such situations when the response time is more important than sophisticatedly analyzed results. Although it is a technology to facilitate real-time and lightweight intelligence, the process of training models and converting it to approximate network require some amount of resource. Therefore, the approximated model can be deployed on smart devices but the learning and approximation processes still need to take places on resource rich platforms.

6. IANA Considerations

This document requests no action by IANA.

7. Acknowledgements

8. Contributors

9. Informative References

[Hochreiter] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[Chung] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555v1 [cs.NE]*, 2014.

[Krizhevsky] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097-1105.

[Doersch] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908v2 [stat.ML]*, 2016.

[Goodfellow]. I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672-2680.

[Ba] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in Proc. Adv. Neural Inf. Process. Syst., Montreal, QC, Canada, 2014, pp. 2654-2662.

[Denil] M. Denil, B. Shakibi, L. Dinh, N. de Freitas, and M. Ranzato, "Predicting parameters in deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2148-2156.

[Han] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 1135-1143.

[Venkataramani] S. Venkataramani, A. Ranjan, K. Roy, and A. Raghunathan, "AxNN: Energy-efficient neuromorphic systems using approximate computing," in *Proc. Int. Symp. Low Power Electron. Design*, ACM, 2014, pp. 27-32.

[Moons] B. Moons, B. De Brabandere, L. Van Gool, and M. Verhelst, "Energy- efficient ConvNets through approximate computing," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Lake Placid, NY, USA: IEEE, 2016, pp. 1-8.

[Mohammadi] Mohammadi, Mehdi, et al. "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Communications Surveys &*

Tutorials, 2018, pp. 2923-2960.

[Kriegel] Kriegel, Hans-Peter, et al. "Density-based clustering," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2011, pp. 231-240.

Authors' Addresses

Jun Kyun Choi (editor)
Korea Advanced Institute of Science and Technology (KAIST)
193 Munji Ro, Yuseong-gu, Daejeon
Korea

Email: jkchoi59@kaist.ac.kr

Na Kyoung Kim
Korea Advanced Institute of Science and Technology (KAIST)
193 Munji Ro, Yuseong-gu, Daejeon
Korea

Email: nkim71@kaist.ac.kr

Jae Seob Han
Korea Advanced Institute of Science and Technology (KAIST)
193 Munji Ro, Yuseong-gu, Daejeon
Korea

Email: j89449@kaist.ac.kr

Min Kyung Kim
Korea Advanced Institute of Science and Technology (KAIST)
193 Munji Ro, Yuseong-gu, Daejeon
Korea

Email: mkkim1778@kaist.ac.kr

Gyu Myoung Lee
Liverpool John Moores University
Barkhill Rd, Merseyside, Liverpool L17 6BD
United Kingdom

Email: G.M.Lee@ljmu.ac.uk