Enhancing Support for Machine Learning and Edge Computing on an IoT Data Marketplace

Kurian Karyakulam Sajan
University of Southern California
Los Angeles, CA, USA
karyakul@usc.edu

Gowri Sankar Ramachandran
University of Southern California
Los Angeles, CA, USA
gsramach@usc.edu

Bhaskar Krishnamachari
University of Southern California
Los An geles, CA, USA
bkrishna@usc.edu

ABSTRACT

IoT applications are increasingly employing machine learning (ML) algorithms to manage and control the operational environment autonomously while predicting future actions. To leverage these emerging technologies, the application developers require an enormous amount of data to build models. Data marketplaces enable the IoT application developers to buy data from IoT device owners to train machine learning models. Contemporary data marketplaces only focus on connecting the IoT infrastructure owner (seller) with application developers (buyer) while lacking integrated support for data analytics. Application developers are required to manually create and manage machine learning pipelines by combining edge computing resources with data sources. In this paper, we present an architectural framework to build machine learning pipelines for data marketplaces automatically. Our framework enables application developers (buyers) to leverage the edge computing resources provided by the sellers and compose low-latency IoT applications that incorporate ML-based processing. We present a proof-of-concept implementation on the I^3 data marketplace and outline open challenges in combining machine-learning, AI, and edge computing technologies with data marketplaces.

CCS CONCEPTS

• **Information systems** → Computing platforms.

KEYWORDS

Machine learning, Artificial intelligence, Data marketplace, IoT, Internet of Things, Edge Computing

ACM Reference Format:
Kurian Karyakulam Sajan, Gowri Sankar Ramachandran, and Bhaskar Krishnamachari. 2019. Enhancing Support for Machine Learning and Edge Computing on an IoT Data Marketplace. In *First International Workshop on Challenges in Artificial Intelligence and Machine Learning (AIChallengeIoT'19), November 10–13, 2019, New York, NY, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3363347.3363364

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

AIChallengeIoT'19, November 10–13, 2019, New York, NY, USA

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-7013-4/19/11…$15.00
https://doi.org/10.1145/3363347.3363364

1 INTRODUCTION

The advancement of artificial intelligence (AI) and machine learning (ML) algorithms and their adoptions in real-world applications in the area of Internet of Things (IoT), smart cities and connected and autonomous vehicles increase the need for the data. To meet the increasing demands for the data, IoT data marketplaces [4, 7, 12, 13] are being developed to allow the IoT device owners to sell their data with application developers. The marketplace model has the potential to decrease the application development time and relieve the application developers from managing the deployment while incentivizing the data providers.

State-of-the-art data marketplaces such as Intelligent IoT Integrator (I^3) [7], Terbine [13], Streamr [12], and IOTA data marketplace [4] focus more on connecting the sellers and buyers, but they lack native support for data analytics. Besides, the emerging edge computing technology facilitates the application developers to process the data close to the source to minimize the latency. Existing marketplace frameworks do not provide support for edge computing to build low latency applications. We believe that integrating support for edge computing and data analytics is a natural extension to data marketplaces.

A wide array of frameworks have been developed to create ML pipelines, including PyTorch [6] and Tensorflow [1]. However, such frameworks focus on processing on static data sets and operate in a siloed setting typically in a single stakeholder environment. On the other hand, the literature proposes edge computing frameworks with support for ML and AI [2, 5, 8, 10, 11] to create intelligent IoT and smart city applications. Such frameworks focus on issues such as off-loading, scheduling of computation, or energy-latency tradeoffs. Several solutions have been proposed to meet the growing demands of ML and edge computing technologies, but these existing literature failed to discuss the integration of data analytics and edge computing with data marketplaces.

This paper aims to fill this gap by supporting ML and edge computing directly on an IoT data marketplace. Besides, the involvement of multiple stakeholders in the marketplace environment creates multiple deployment configurations for edge and cloud-based ML applications. We present different configuration possibilities that arise when integrating edge computing and data analytics with a data marketplace. We also implement an extension of the I^3 data marketplace with new built-in support for deploying and managing edge computing and machine learning pipelines. We present a deployment and assessment engine for deploying computation, including model training and inference tasks on a remote edge computing platform owned by sellers and other third parties reliably and securely. We present a proof-of-concept implementation and outline open challenges in combining machine learning, AI, and edge computing technologies with data marketplaces.

The rest of the paper is structured as follows: Section 2 provides the background on data marketplace, edge computing, and motivate the need to support ML, AI, and edge computing on the data marketplace. Our novel framework is presented in Section 3. Section 4 presents our proof-of-concept implementation, and the evaluation results are presented in Section 5. We list some of the open challenges in Section 6 before concluding the paper in Section 7.

2 BACKGROUND

In this section, we introduce the relevant concepts to the paper.

2.1 I^3 Data Marketplace

The IoT Marketplace approach aims to enable a world where application developers can gain access to the myriad of sensors and actuators that others have deployed and connected to the network. The sensor owners can take the initiative and deploy intelligent sensors in anticipation of an emerging and independent application market that will utilize their data for the benefit of its users.

The I^3 data marketplace [7] that is being developed at USC allows diverse device owners to contribute (sell) data streams, while different application developers can connect to the I^3 marketplace to obtain (buy) one or more streams meaningful to their application. Furthermore, I^3 allows third-party data brokers to buy *raw* data, apply data analytics (using machine learning and artificial intelligence pipelines) and sell *refined* data streams back through I^3. Furthermore, as device owners begin to monetize the IoT sensors and actuators, brokers will emerge to provide value-added processing and analytics as well as control services to enhance the IoT data further and manage actuation.

With the I^3 Marketplace platform, IoT device owners will be allowed to access or trade their sensor data and actuator access with different vendors to create a supportive environment for advanced data analytics programs. Data analytics can be efficiently developed and supported in a multi-vendor-multi-owner device environment. The IoT Marketplace provides ease to build and deploy IoT applications and devices by maximizing the level of data reuse and interoperability.

Other data marketplace efforts, including Terbine.IO [13], Streamr [12], and the IOTA data marketplace [4] also assumes that the buyers handle the analytics outside the marketplace.

2.2 Edge Computing for IoT Data Analytics

Edge computing is applied in IoT and smart city applications to process data at edge devices, which are close to the data sources. The edge devices can be used to train models or run inference or prediction close to the origin [2, 5, 8, 10]. Such an approach not only offloads the processing cost to remote edge devices but also reduce long-distance traffic and latency. Existing literature presents a software framework for deep learning [11] and real-time media-based IoT applications [3] at the edge devices and highlights the benefits of employing edge computing for IoT data analytics.

Throughout this paper, when we say *edge computing platform*, we mean either a dedicated computation platform close to the IoT sensors (typically, one-hop away from the IoT sensor node) that generate the data or a sensor node with a powerful onboard computation platform.

2.3 Enhancing Support for Edge Computing in IoT Data Marketplace

The existing data marketplaces including I^3 [7], Terbine [13], Streamr [12], and IOTA data marketplace [4] focus mainly on bridging the data providers with data consumers and generally assume that the consumers manually compose data analytics pipelines using cloud or edge platforms. The literature on edge computing for IoT data-analytics shows the benefits of performing analytics on edge computing platforms, but to our knowledge integrating such an approach has not been investigated under a data marketplace model. The I^3 data marketplace platform encourages data buyers to resell the processed data without providing built-in software support to simplify the integration of both edge computing and data analytics techniques. This paper fills this gap by:

- Presenting a framework that combines edge computing and data analytics with an IoT data marketplace.
- Showing the different possibilities to schedule model training and inference tasks on computation infrastructures owned by the seller, buyer, or other third parties on resources located at edge or cloud platforms.
- Introducing a *deployment and assessment engine* to manage the remote edge or cloud platforms.

Throughout the rest of this paper, we will use I^3 as a reference data marketplace to describe our proposed framework and to implement and evaluate our proposal.

3 A FRAMEWORK TO SUPPORT MACHINE LEARNING AND EDGE COMPUTING IN IOT DATA MARKETPLACE

While today's I^3 data marketplace allows processing of raw data through the use of third party data brokers (see Figure 2-left), such an approach have some disadvantages:

- The application suffers from *high latency* because the data is typically processed far away from the data source.
- The seller must *trust* the third party with their raw data.
- The *buyer cannot easily customize the processing* they want on the raw data (they would need to coordinate with the data broker outside of the I^3 platform).

To address these issues, we propose and implement a new approach on the I^3 platform, in which the seller can post access to computation in addition to the data itself on the marketplace. This allows the buyer to send custom models to process the data while the seller has the peace of mind that the data is processed at its site (which could be useful for more privacy-sensitive data). Further, this approach can leverage the latency benefits of edge computing.

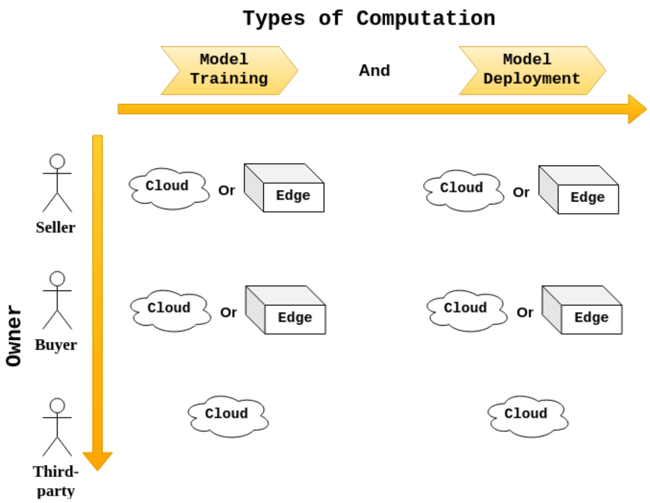

Figure 1: Different configurations emerge from the integration of machine learning and edge computing with IoT Data Marketplace.

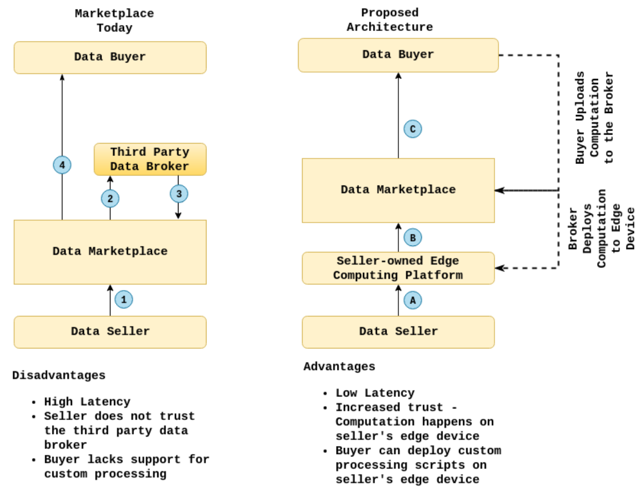

Figure 2: Proposed architecture Compared against today's (I^3) marketplace.

3.1 Possible Configurations for Deploying Machine Learning and Edge Computing on a Marketplace

A data marketplace consists of a seller, a buyer, and a broker (marketplace operator). The broker acts as an intermediary, and it allows the seller and the buyer to exchange data with each other while handling payments, privacy issues, and other bookkeeping activities. If we consider a simple setting with a single data seller and a buyer interested in the processed version of that data, there can be many different configurations pertaining to three key dimensions, which is shown in Figure 1:

- Types of Computation: Model training and model deployment
- Owner of Computation Point: Seller or buyer or third party
- Location of Computation Point: In-Cloud or on-edge (in case the owner is the seller or buyer)

Thus, these give rise to many different possibilities (18, in fact) including the following two that we highlight and focus on in this work:

- Model training and deployment on third party cloud.
- Model training on buyer-owned cloud and model deployment on seller-owned edge.

As noted above, the former approach is already enabled within I^3 by having the third party acting as a data broker. In this paper, we extend the I^3 to implement the latter scheme and focus on the inference phase, not the training, which relieves us from addressing storage issues.

3.2 Data Marketplace with Support for Edge Computing and Machine Learning

Figure 2 (left) presents the approach followed by the contemporary data marketplaces, which does not provide built-in support for edge computing or data analytics close to the data source. There are *four-hops* between the seller and the buyer, which not only increase the latency but also imposes a high processing load on the marketplace broker. The marketplace infrastructure may suffer from performance and scalability issues when it handles a large volume of data in the order of megabytes (MB) or more [3].

Our proposed approach, presented in Figure 2 (right), enables the buyer to run custom processing scripts, including model training and inference on the seller-owned edge device. Such an approach reduces the number of hops, the latency and also minimizes the load on the marketplace. In addition, the seller is in control of his data when the data processing happens in his/her edge, but this may impact the privacy of the buyer, as the seller has the visibility to the buyer's computation. We will discuss this issue in Section 6.

3.3 Deployment and Assessment Engine

When integrating the edge computing capability to a data marketplace, it is essential to a) verify the capability of the edge device, b) check the software for compatibility, and c) provide support for managing and deploying the edge computing resource. The *deployment and assessment engine* is created to meet those demands.

The I^3 is extended with support for service daemon, which runs on the edge device to manage and coordinate with the marketplace. The engine's functionality pertaining to the seller's edge device includes verification and maintenance of connectivity, assess the closeness to the data source, survey the computation capability, and communication with service daemon on the device. The service daemon listens for configuration commands from the marketplace platform such as setup status, connectivity check, dummy model test, logging data, and system upgrades.

4 PROOF-OF-CONCEPT IMPLEMENTATION USING I^3 DATA MARKETPLACE

4.1 Current I^3 Marketplace

The present version of the I^3 platform enables IoT device owners to register their data products on the I^3 platform. The Create Product page, which is shown in Figure 4, allows the user to create a product by defining the topic under which the data will be published to the marketplace along with the price and other relevant metadata. Buyers can browse through the marketplace using either the web frontend or Web Services API to buy data products. The I^3 platform uses the MQTT publish-subscribe broker for data exchange between the sellers and buyers. An authentication framework, that is part of the I^3 platform, manages the authorization for the sellers. The present version I^3 platform does not provide built-in support for data analytics and edge computing. To enable the sellers and buyers to develop applications involving data analytics and edge computing technologies easily, we have extended the I^3 platform, as detailed in Section 4.2.

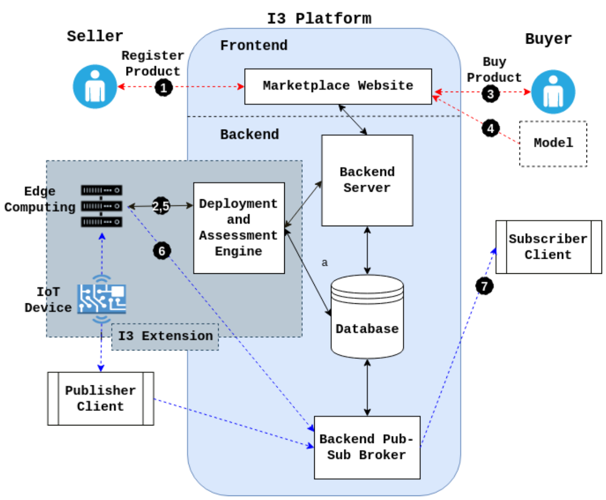

Figure 3: Architecture of the extended I^3 Marketplace

4.2 Extensions to I^3 Marketplace

In this section, we explain the extensions that were made to the I^3 to provide support for ML and edge computing.

Frontend: We have modified the create product page to allow the seller to register his/her edge computing capability to the marketplace. The seller must prove the edge connectivity, computation capacity and closeness to the data source during the product creation phase. On the buyer side, we have modified the payment handling page to allow the buyer to pay for edge computing and deploy the necessary computation packages to the remote edge device owned by the seller. Note that the buyer is not provided direct access to the edge computing node (which could potentially raise security concerns for the seller). Instead, the extended I^3 marketplace deploys the necessary computation tasks on the seller's edge device. Besides, we allow the buyer to monitor the setup process through the GUI to ensure that the computation tasks are deployed correctly on the seller's edge device.

Backend: At the backend, we have developed a deployment assessment engine, which we described in Section 3.3, to make sure that the seller has registered a capable edge device on the marketplace and to execute the installation scripts provided by the buyer on the seller's edge device.

The above changes impact the workflow as follows:

(1) The seller registers a data product by entering the necessary details, including the topic and the availability of edge computing.
(2) Once the option of edge computing is selected, I^3 creates a key-pair and allows the user to download the public key. The user is required to setup SSH connectivity to the edge device using the key as part of the product registration.
(3) The marketplace server hands over the responsibility to the backend assessment engine to approve and list the edge device.
(4) Once the engine verifies the product and the setup of service daemon is done, the product gets listed on the I^3 Marketplace. The seller may track the status of approval and setup on the status Page. This concludes the flow for the seller on product registration.
(5) When a buyer purchases a product, he/she could request for the edge computation, if the seller provides the service. If the edge computation is selected, he/she would be required to submit the model and prediction script.
(6) Once the necessary information is received from the buyer, the marketplace server hands over the responsibility to the assessment and deployment engine to set up and run the model.
(7) The setup status can be monitored by the buyer on the Status Page.
(8) Once the model is set up, the buyer can now subscribe to the processed data.

5 EVALUATION

Figure 2 shows the data flow for the current and the extended I^3 marketplace implementation. In this section, we compare the performance of the current version with our extended I^3 marketplace using two representative applications.

5.1 Latency for Two Representative Applications

1. Parking Prediction for LAX Airport: One of the active data products in our I^3 marketplace is parking data from LAX Airport. Our dataset contains parking availability and vacancies of multiple LAX parking structures over two weeks, where a single data input has details for all the structures . A linear regression model was used over the dataset for predicting the parking availability and inputs to the model are latest 7 live parking data which is updated hourly with the number of vacancies, total availability, and location, along with the computed input of the expected number of vacancies

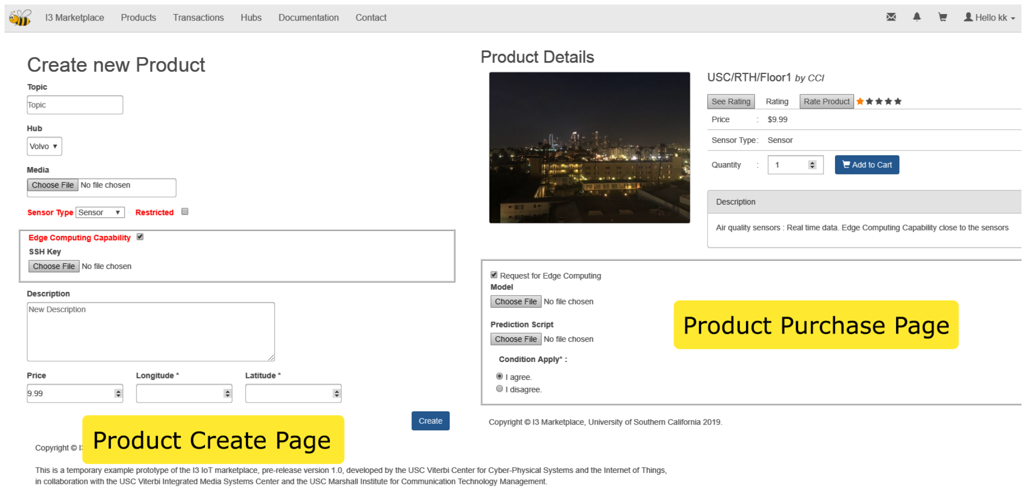

Figure 4: Create and Purchase Product Pages on I^3 Marketplace

for the day of the week. The model predicts a real-time response to the number of occupied parking spots for the next hour. The model can be used to predict parking availability of up to the next 4 hours with an accuracy of 80%. We were able to estimate the parking availability with high accuracy for an hourly granularity on a simple linear regression model. We required to compute on an edge device that had minimal computation capacity thus we maintained a high throughput to compute parking for multiple structures. The average input data size is 2.75 KB, and the same was used in our latency evaluation.

2. Object Recognition: The second representative application classifies objects on camera images. This example application shows that the data size could be in the order of megabytes when the seller directly posts camera feeds to the marketplace and application of edge computing to this application significantly reduces the load on the I^3 broker. We have tested the object detection using Convolution Neural Network to classify between five classes of objects. The average data size is 2.6 MB, which is one of the message sizes used in our latency evaluation.

5.2 Latency Evaluation

During our evaluation, a single publisher was configured to publish data to a broker at different sizes (2.75 KB and 2.6 MB), and our broker was not subjected to heavy load. Under these conditions, we estimate the end-to-end latency for two different data products - 2.75 KB and 2560 KB - to show how the data size influences the latency on the data marketplace. The evaluation is carried out for the current I^3 and the extended marketplace platforms and the results are presented in Figure 5 and Figure 6. The median latency for the extended version of I^3 and I^3 is 1.1 seconds and 1.3 seconds respectively for 2.75 KB of data, as shown in Figure 5. Similarly, the median latency is 5.8 seconds and 7.2 seconds for the extended version of I^3 and I^3 respectively for 2560 KB of data. These results show that the use of edge computing on the data marketplace not only minimizes the latency but also reduces the load on the broker and the network. The difference is because the third-party broker

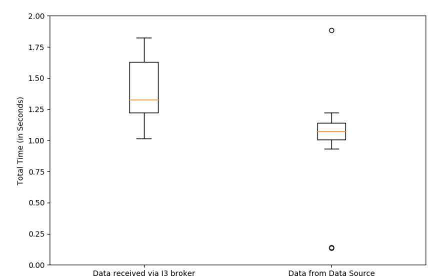

Figure 5: Total Time for 2.75 KB of data to flow via Data broker vs directly to Edge (LAX parking prediction app.).

used I^3 platform broker for both the raw and the processed data propagation.

The other benefits of our extended I^3 data marketplace are that it provides a simple and easy to use platform for the application developers to easily create ML pipelines leveraging the edge computing resources offered by either the seller or other service providers. This feature is hard to evaluate quantitatively, however, we plan to evaluate by inviting users to try both I^3 and the extended version of I^3 to answer qualitative survey questions related to the user interface and user experience.

6 OPEN CHALLENGES

This section discusses the open challenges in combining edge computing and data analytics with IoT data marketplaces.

1. Privacy-Preserving Computation: With our framework, we can allow the seller to specify some amount of privacy, e.g., what function of their data can be revealed. In addition, the buyer can also specify some level of privacy in that exactly what function they implement over the revealed data is not made visible to the seller. The former is straightforward as the seller can control any

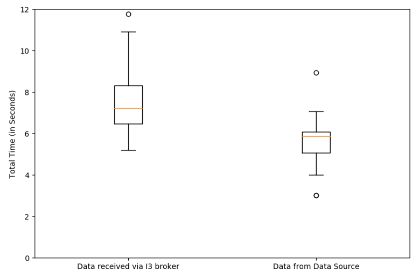

Figure 6: Total Time for 2560 KB of data to flow via Data broker vs directly to Edge (Object detection app.).

pre-processing of their data themselves; We can leverage homomorphic encryption [9] for the latter part, which enables the buyer to run an arbitrary computation on encrypted data.

2.Pricing: We are extending the I^3 marketplace with support for machine learning and edge computing. At present, the seller prices his/her data when it is posted on the marketplace. With the introduction of edge computing, the owner of the computation infrastructure may have to price the computation based on the processing, communication, and the storage used for the computation tasks. As a next step, the marketplace may also sell machine learning models for purchase, which allows the buyers to choose the desired model and deploy it on an edge device provided by the sellers.

3. Verifiable Computation: When a buyer deploys a computation task on a remote edge device owned by either a seller or third party, the buyer needs to be able to verify that the computation device has performed the computation. The marketplace platform should, therefore, provide support for verifiable computation to validate the edge computing platforms. Our prior work, SmartEdge [14] proposes a solution for verifying the computation, which we plan to integrate with our framework.

4. Orchestrating computation over multiple sellers: In this work, we have considered the simplest setting of processing data from a single seller; a more complicated challenge we plan to tackle in future work is to orchestrate distributed edge computing over multiple data providers in the marketplace.

7 CONCLUSION

IoT data marketplaces enable the IoT device owners to sell their data to application developers (buyers). The contemporary data marketplaces lack support for edge computing and machine learning. In this paper, we have shown how IoT data marketplaces can be extended to offer built-in support for machine learning and edge computing. We have implemented and evaluated our proposal using the I^3 platform. To the best of our knowledge, this is the first implementation of an enhanced IoT data marketplace that combines machine learning and AI with edge computing in an integrated manner.

ACKNOWLEDGMENT

This material is based in part upon work supported by Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001117C0053. Any views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. USENIX Association, Savannah, GA, 265–283. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[2] G. Ananthanarayanan, P. Bahl, P. BodÃŋk, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha. 2017. Real-Time Video Analytics: The Killer App for Edge Computing. *Computer* 50, 10 (2017), 58–67. https://doi.org/10.1109/MC.2017.3641638

[3] Giorgos Constantinou, Abdullah Alfarrarjeh, Seon Ho Kim, Gowri Sankar Ramachandran, Bhaskar Krishnamachari, and Cyrus Shahabi. 2019. A Crowd-Based Image Learning Framework Using Edge Computing for Smart City. In *To appear at the Fifth IEEE International Conference on Multimedia Big Data*.

[4] IOTA Foundation. 2019. IOTA Data Marketplace. https://data.iota.org/#/demo/#list.

[5] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin. 2017. Software-Defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach. *IEEE Communications Magazine* 55, 12 (Dec 2017), 31–37. https://doi.org/10.1109/MCOM.2017.1700246

[6] Nikhil Ketkar. 2017. Introduction to pytorch. In *Deep learning with python*. Springer, 195–208.

[7] Bhaskar Krishnamachari, Jerry Power, Seon Ho Kim, and Cyrus Shahabi. 2018. I3: an IoT marketplace for smart communities. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 498–499.

[8] H. Li, K. Ota, and M. Dong. 2018. Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. *IEEE Network* 32, 1 (Jan 2018), 96–101. https://doi.org/10.1109/MNET.2018.1700202

[9] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. 2011. Can Homomorphic Encryption Be Practical?. In *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop (CCSW '11)*. ACM, New York, NY, USA, 113–124. https://doi.org/10.1145/2046660.2046682

[10] P. Patel, M. Intizar Ali, and A. Sheth. 2017. On Using the Intelligent Edge for IoT Analytics. *IEEE Intelligent Systems* 32, 5 (Sep. 2017), 64–69. https://doi.org/10.1109/MIS.2017.3711653

[11] Gowri Sankar Ramachandran, Sharon L.G Contreras, and Bhaskar Krishnamachari. 2019. Publish-Pay-Subscribe Protocol for Payment-driven Edge Computing. In *2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 19)*. USENIX Association, Renton, WA. https://www.usenix.org/conference/hotedge19/presentation/ramachandran

[12] Streamr. 2019. Streamr. https://www.streamr.com/marketplace.

[13] Terbine.IO. 2019. Terbine.IO. https://www.terbine.io/.

[14] Kwame-Lante Wright, Martin Martinez, Uday Chadha, and Bhaskar Krishnamachari. 2018. SmartEdge: A Smart Contract for Edge Computing. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 1685–1690.