Journal of information and
communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 20(2): 96-102, Jun. 2022 Regular paper

Scalable Prediction Models for Airbnb Listing in Spark Big Data Cluster using GPU-accelerated RAPIDS

Samyuktha Muralidharan[1], Savita Yadav[1], Jungwoo Huh[2], Sanghoon Lee[2], and Jongwook Woo[1]*

[1]Department of Information Systems, California State University, 90032, USA
[2]Department of Electrical Engineering, Yonsei University, 03722, Korea

Abstract

We aim to build predictive models for Airbnb's prices using a GPU-accelerated RAPIDS in a big data cluster. The Airbnb Listings datasets are used for the predictive analysis. Several machine-learning algorithms have been adopted to build models that predict the price of Airbnb listings. We compare the results of traditional and big data approaches to machine learning for price prediction and discuss the performance of the models. We built big data models using Databricks Spark Cluster, a distributed parallel computing system. Furthermore, we implemented models using multiple GPUs using RAPIDS in the spark cluster. The model was developed using the XGBoost algorithm, whereas other models were developed using traditional central processing unit (CPU)-based algorithms. This study compared all models in terms of accuracy metrics and computing time. We observed that the XGBoost model with RAPIDS using GPUs had the highest accuracy and computing time.

Index Terms: Airbnb Price, Big Data, XGBoost, RAPIDS, Scalable Predictive Analysis

I. INTRODUCTION

A graphics processing unit (GPU) supports parallel computing and works well with deep learning libraries. Big-data solutions provide distributed files and computing systems. GPUs offer high computing performance because of their massively parallel architecture, comprising thousands of smaller, highly efficient cores designed to handle multiple tasks simultaneously. In contrast, a central processing unit (CPU) comprises a few cores optimized for sequential serial processing [1-5]. Thus, GPUs can process data much faster than configurations containing only CPUs, while offering a lower price per performance. Integrating GPUs into big data platforms, such as Hadoop Spark clusters, can improve high-performance computing. Big data is defined as distributed parallel computing systems used to store and process large-scale datasets [1-3]. This study selects the Hadoop and Spark Big Data clusters, which support distributed file systems and parallel computing. Integration can increase the parallelism between GPU cores and multiple nodes. Furthermore, it reduces the gap between big data and GPU communities because applications and models using GPU can read large-scale data more efficiently and easily from the distributed file systems of the big data platform.

Airbnb is an online marketplace that connects people who want to rent out their homes with people looking for accommodation in that locale. It assists people list, explore, and book unique properties worldwide. This study predicts Airbnb listing prices using legacy and big data platforms with CPUs and GPUs by developing a model that predicts the optimal price of the property by considering the features of the listings. One challenge faced by most Airbnb hosts is determining the optimal rental price per night. Usually, users are presented with a good selection of listings, which they can

Received 07 December 2021, **Revised** 14 march 2022, **Accepted** 29 March 2021

***Corresponding Author** Jongwook Woo (E-mail: jwoo5@calstatela.edu, Tel: +1-323-343-2916)
Department of Information Systems, California State University, 90032, USA.

Open Access https://doi.org/10.6109/jicce.2022.20.2.96 print ISSN: 2234-8255 online ISSN: 2234-8883

filter based on criteria such as price, number of bedrooms, and room type. Considering Airbnb is a marketplace, the amount a host can charge on a nightly basis is closely linked to the dynamics of the marketplace. If the host charges above the market price, users select other affordable alternatives. If the nightly rent price is set too low, the hosts will miss potential revenue. Therefore, machine-learning models are used to predict the optimal prices that the hosts can set for their properties by assisting hosts better understand how different features of the listing, such as bedroom, location, and house type, can be used to predict the price accurately.

We used Airbnb Listings data, comprising various Airbnb properties and their features. The second dataset used was Airbnb Reviews, which includes user reviews for each listing. With better price suggestion estimates, Airbnb home providers can reach an equilibrium price that optimizes profit and affordability. The size of the entire dataset is 4 GB. The sentiments of reviews are determined using sentiment analysis to utilize user reviews for building prediction models. Only the sentiments of the reviews are included, with actual user reviews eliminated from the dataset. Furthermore, we filtered the dataset to contain only Airbnb listings from the United States. Thus, the size of data used to build models for price prediction was reduced to 400 MB using a big data platform, the Hadoop Spark cluster. The legacy platform cannot process a 400 MB dataset to train the models because a traditional single node has resource limitations. Therefore, we built models with a sample dataset of 30 MB using Azure machine learning (ML) Studio's legacy platform. In addition, we implemented models with the entire dataset using the big data platform, Spark cluster. It stores the entire dataset in distributed systems and trains the models in parallel computing, which can achieve high-performance computing time. Then, we trained the models using a Spark cluster with GPUs to achieve faster computing times.

The reminder of the paper is organized as follows. Section 2 presents the related work. Section 3 describes existing methods. Section 4 presents the experimental results of the models built on both legacy and big data platforms with CPUs and GPUs. Finally, section 5 presents our conclusions regarding the experimental results of the predictive models.

II. RELATED WORK

Choudhary et al. [6] analyzed Airbnb listings in San Francisco to understand how different attributes of the listings can be used to predict the price accurately. They used a random forest regressor to predict the price of listings. They also focused on the availability of listings by segregating them into high/low availability using the clustering and Naïve Bayes algorithms.

Kalehbasti et al. [7] attempted to develop the best-performing model for predicting Airbnb prices in New York City. The model was built based on a limited set of features. ML techniques, such as linear regression, support-vector regression (SVR), neural networks, and feature importance analyses are used to achieve the best results. They also performed sentiment analysis of the reviews using the TextBlob sentiment analysis library, while we used Apache Hive in the Hadoop cluster to perform sentiment analysis. The proposed approach is linearly scalable for handling massive datasets.

Mitchell and Frank [8] implemented an efficient GPU-accelerated tree-construction algorithm within the XGBoost library. The algorithm was developed on top of efficient parallel primitives and switched between two modes of operation depending on tree depth. The tree-construction algorithm was executed entirely on a GPU and showed high performance with various datasets and settings. They showed that the algorithm was memory efficient enough to process the entire Higgs dataset, containing 10 million instances and 28 features, entirely within the GPU memory. The algorithm provided a practical resource for XGBoost users to process large datasets and significantly reduce processing times, showing that gradient boosting tasks are good candidates for GPU acceleration and are no longer solely the domain of multicore CPUs.

The aforementioned studies implemented prediction models using legacy platforms and programming languages such as Python [9]. Our study deals with building ML models using a legacy platform, AzureML Studio, and a big data platform, such as a GPU-accelerated Spark cluster. The Spark cluster utilizes a Spark computing engine with a GPU to store and process massive datasets in distributed systems with parallel computing while also providing RAPIDS for GPUs and a Spark ML library for machine learning.

III. SYSTEM MODEL AND METHODS

In this study, various ML algorithms were used to train and build predictive models to predict the prices of Airbnb listings on both legacy and big data platforms. User reviews are available for listings in the Airbnb reviews dataset. To utilize user reviews in building predictive models, we determined the sentiment of the reviews by performing sentiment analysis using Apache Hive—a part of the Hadoop ecosystem. Hive provides an SQL-like interface to query data, such as Hive QL, similar to SQL. The overall sentiment of each listing, Id, is determined by utilizing the sentiment of the listings. We adopted sentiment as a feature in building the models to predict the prices of listings.

A. Regression Algorithms

Regression is a supervised machine learning technique that predicts continuous values. In this study involving price prediction, we used regression models, price, the target variable, or label as a numeric variable. While building models to predict the price of the listings with a random dataset, three algorithms were used in the legacy platform: AzureML-decision forest regression, boosted decision tree, and Bayesian linear regression. We used the tune model hyperparameters module to improve model performance, cross-validate module to generalize the model, and permutation feature importance module to eliminate less critical features.

When implementing models for price prediction with the entire dataset, we used the Spark cluster in Databricks with the following machine learning algorithms: XGBoost regression, gradient boosted tree regression, decision tree regression, and random forest regression. A pipeline computation was developed for feature transformation and training a regression model. A cross-validator was used to identify the general model with the best performing parameters. The evaluation metrics for accuracy were the root mean square error (RMSE) and the coefficient of determination (R2). We also measured the computing time required to build and train the models.

B. XGBoost

Extreme gradient boosting (XGBoost) is an open-source library that provides an efficient and effective gradient boosting algorithm [4]. It is a decision-tree-based ensemble ML algorithm that uses a gradient-boosting framework. XGBoost is an efficient and easy-to-use algorithm that delivers high performance and accuracy compared with other algorithms. It provides improved performance through system optimization and algorithmic enhancements as it focuses on speed, flexibility, and model performance with high parallelism [5].

Mishra stated that that XGBoost could be used in various use cases, such as ranking, classification, regression, and user-defined prediction problems, and referred to it as an ‘All in One’ algorithm. It is an ideal blend of software and hardware optimization techniques to yield prevalent outcomes using fewer computing resources in the shortest amount of time. XGBoost is highly flexible and uses parallel processing; it supports regularization and handles missing data using its built-in features [10, 11].

C. RAPIDS

According to Nvidia, data science workflows have traditionally been slow and cumbersome, relying on CPUs to load, filter, and manipulate data, and train and deploy models. GPUs substantially reduce infrastructure costs and provide superior performance for end-to-end data science workflows utilizing Nvidia CUDA, a parallel-computing architecture that supports parallel operations [12–14].

Spark 3.0 in the GPU supports deep learning and legacy ML, as shown in Figure 1. It also resolves the issue of transferring large-scale datasets to the GPU for processing, as illustrated in Fig. 1.

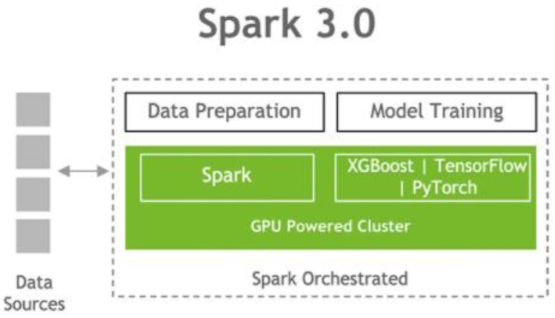

Fig. 1. Apache Spark 3.0 in GPU [12].

Nvidia created Rapids for Spark 3.0, drastically improving the Extract-Transform-Load (ETL) performance, data engineering, data analysis, and data prediction. RAPIDS is a suite of open-source software libraries for executing end-to-end data science and analytics pipelines entirely on GPUs, allowing substantial speed-up, particularly on large datasets. The RAPIDS team works closely with the distributed machine learning common (DMLC) XGBoost organization. XGBoost now includes seamless, drop-in GPU acceleration, significantly speeding up model training and improving the accuracy for better predictions. It has integrated support for running across multiple GPUs, delivering even more significant performance improvements [9].

Table 1. Technical specifications of our big data cluster

Azure ML Studio	Spark Cluster
Workspace Type: Free **Storage**: 10GB **Nodes**: 1 **Region**: South Central US	**Spark version**: DBR 8.2 ML (Apache Spark 3.1.1, GPU, Scala 2.12) **Cloud Instance**: g4dn.xlarge **Number of GPUs**: 1 per instance **Nodes**: 4 **Memory**: 64.0 GB **CPU Cores**: 16

The RAPIDS accelerator for Apache Spark leverages GPUs to accelerate processing using RAPIDS libraries. The RAPIDS accelerator GPU-accelerates Apache Spark 3.0, data science pipelines without code changes, and speeds up data processing and model training while substantially lowering infrastructure costs [12-14].

IV. BIG DATA PREDICTIVE ANALYSIS USING CPUS AND GPUS

We implemented various ML models to predict the price of Airbnb listings using legacy, AzureML Studio, and big data cluster, Spark ML. The entire dataset was of 4 GB, including worldwide listings. However, we filtered USA listings using only Hive in the big data cluster, which becomes 400 MB. Its sampled dataset of size 30 MB was used to perform predictive analysis in Azure ML Studio, while in the Spark cluster, the entire dataset of 400 MB in the USA was used to build predictive models.

Big data platform was built in the Databricks Enterprise edition. Here, a GPU-accelerated Spark cluster is used to build predictive models for the entire dataset using the Scala and Spark ML library. Table 1 lists the technical specifications of these two platforms.

A. Predictive Analysis using Legacy Azure ML Studio

We built the models with the sample dataset in Azure ML Studio, following the algorithms below. With a sampled file of 30 MB, we split the data into 70% training and 30% testing. The Tune model hyperparameters module was used to identify the best-performing model.

Table 2. Comparison of metrics for Azure price prediction

AzureML Studio	RMSE	R2	Computing Time (seconds)
Decision Forest Regression	32.41	0.6030	90
Boosted Decision Tree Regression	29.23	0.6723	120
Bayesian Linear Regression	29.25	0.6720	120

The permutation feature importance module was used to determine the best features for an accurate model, and the cross-validation module was used to generalize the model. The evaluation result of the decision forest regression is an RMSE of 32.4 and R2 of 0.6; the model took 90 s to build and run. The boosted decision tree regression had an RMSE of 29.2 and R2 of 0.67; the model took 120 s to run. Bayesian linear regression presents the experimental results with an RMSE of 29.2 and R2 of 0.67, with a computing time of 120 s. Table 2 lists the evaluation results of the three machine learning algorithms.

Table 2 shows the best results with the smallest RMSE and R2 values closest to 1. The Bayesian linear and boosted decision tree regression models calculated good RMSE (29.23) and R2 (0.6723) values.

B. Predictive Analysis using Big Data Spark cluster with CPUs and GPUs

We built models for price prediction in the big data Spark cluster using four algorithms: XGBoost regression, gradient boosted tree regression, decision tree regression, and random forest regression models. The entire dataset was stored in the AWS S3 bucket, imported and read as a Spark dataframe in the Databricks Enterprise Edition.

We implemented the XGBoost models using both GPU accelerated prediction and CPU prediction. The models were implemented in a distributed multi-node spark cluster with multi-GPUs using the RAPIDS accelerator and libraries in Databricks. Regression models other than XGBoost were run and evaluated using only multiple CPUs in the Spark Cluster [15-18].

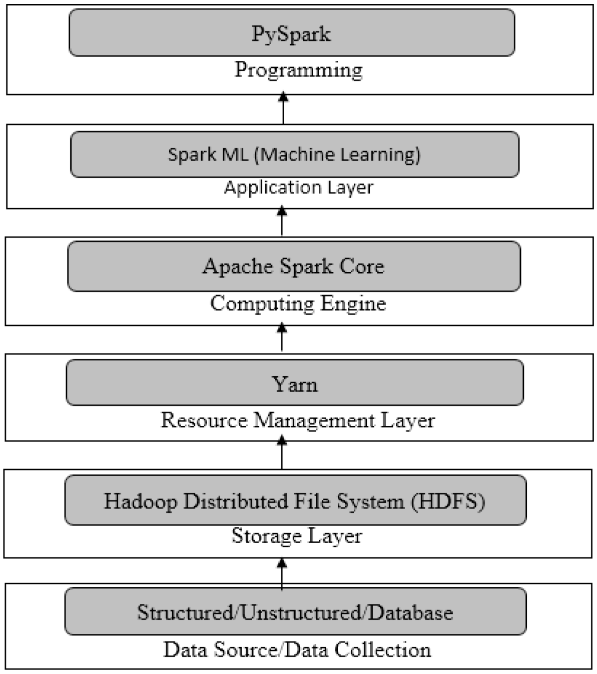
PySpark
Programming
Spark ML (Machine Learning)
Application Layer
Apache Spark Core
Computing Engine
Yarn
Resource Management Layer
Hadoop Distributed File System (HDFS)
Storage Layer
Structured/Unstructured/Database
Data Source/Data Collection

Fig. 2. System and configuration workflow in big data analysis and prediction.

Fig. 2 shows the system and configuration workflow diagram used to collect the data and build a model to predict the listing price.

Table 3. Evaluation metrics for price prediction in big data

Databricks	RMSE	R2	Computing Time (seconds)
XGBoost Regression (GPU)	35.99	0.7162	60
XGBoost Regression (CPU)	35.85	0.7184	77.82
Gradient Boosted Tree Regression	37.24	0.6962	227
Random Forest Regression	37.40	0.6936	232
Decision Tree Regression	38.79	0.6703	112.36

In the first phase, we processed the data engineering to remove outliers and handle null values. The dataset was split into an 80:20 ratio for the training and testing sets. A pipeline was defined for feature transformation and model training. The pipeline consisted of (1) string indexer, which converted string values to indices for categorical features, (2) vector indexer, which created indices for a vector of categorical features, (3) minmax scaler, which normalized the continuous numeric features, (4) vector assembler, which created a vector of categorical and continuous features, and (5) regression algorithm, which trained a regression model.

We adopted a cross-validator to generalize the model and improve its accuracy by finding the optimal performance parameters. The number of folds of the cross-validator was three for all the four algorithms. The model produced by the pipeline was applied to the test data to determine the prediction accuracy. Table 3 shows the evaluation metrics and computing time required for training and evaluating the models with the GPU-accelerated Spark cluster. The root mean square error (RMSE) provides the average difference between the predicted and actual values. Thus, the RMSE indicates the average difference in dollars between the predicted and actual prices of the listings. The coefficient of determination (R2) indicates how well the model fits the data by evaluating the scatter of data points around the fitted regression line. Thus, the closer R2 is to 1, the better the model accuracy.

Table 3 states that the XGBoost regression model provides an RMSE of approximately 36, better than other regression models. Considering the value of the coefficient of determination, we observe that XGBoost provides an R2 value of 0.71, higher than other regression models. Following XGBoost, we observe that random forest and gradient boosted tree regression models provide an RMSE of approximately 37 and R2 of 0.69. In terms of the computing time to train and build the model, XGBoost regression takes 60 s—the shortest time; it is much lesser when compared to the computing time of the other regression models. XGBoost regression model provides the best combination of prediction performance and processing time compared to the other algorithms. Thus, XGBoost is an optimal approach for building price prediction regression model in Airbnb listings.

Furthermore, the XGBoost regression model for price prediction was implemented using GPUs and RAPIDS libraries in the Spark cluster. We built and executed the same XGBoost application using only CPUs. The computation results of XGBoost in the GPU and CPU are listed in the first and second rows of Table 3, respectively. We observe that the computing time of the XGBoost model using GPUs (60 s) is much less when compared to the time taken using CPUs (78 s). XGBoost in the GPU provides a 23% performance gain over XGBoost in the CPU.

V. CONCLUSION

This study implemented and compared the performance of traditional and big data approaches for predictive analysis using AzureML Studio and Spark clusters, respectively. We also discussed the benefits of leveraging GPUs in predictive analysis. We showed that the RMSE and R2 results of the price prediction were better in Spark than in AzureML Studio. Azure ML had difficulty building models with 400 MB; thus, we created the models using a sampled dataset of 30 MB. However, the Spark Cluster can utilize the entire dataset of 400 MB to train and evaluate regression models. The big data Spark platform is linearly scalable, making it possible to add more servers (nodes) when the dataset increases. The data are processed in parallel, resulting in less computing time compared to a single node in legacy AzureML Studio. Thus, the big data approach for predictive analysis provides better performance and efficiency than the traditional machine learning approach.

We showed that the XGBoost regression in the Spark cluster had the best evaluation results with RMSE and R2 compared to the other regression models. We observed that the computing time of XGBoost was much shorter than that of the other algorithms. The experimental results showed an approximately 74% improvement in the accuracy of XGBoost over the gradient-boosted decision tree and random forest algorithms. We also observed that the XGBoost regression model in the Spark cluster utilizing GPUs with RAPIDS libraries reduced the computing time to 23% compared with using only CPUs.

ACKNOWLEDGEMENTS

The Databrick University Alliance supported this research. We appreciate the support of Rob Reed, Program Director at Databricks University Alliance.

REFERENCES

[1] D. Dauletbak, J. Heo, S. Kim, Y. Kim, and, J. Woo, “Scalable traffic predictive analysis for smart city using GPU in big data,” *KSII The 16th Asia Pacific International Conference on Information Science and Technology (APIC-IST)*, pp 144-148, 2021.

[2] J. Woo, Market Basket Analysis Algorithms with MapReduce, DMKD-00150, Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery, vol. 3, Issue 6, pp. 445-452, 2013.

[3] J. Woo, and Y. Xu,. Market Basket Analysis Algorithm with Map/Reduce of Cloud Computing, *The 2011 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2011)*.

[4] J. Brownlee, XGBoost for Regression, *Machine Learning Mastery*, 2021. [Online] Available: https://machinelearningmastery.com/xgboost-for-regression/.

[5] Chip ICT b.v., GPU Computing, the basics: Chip ICT, 2021, [online] Available: https://www.chipict.com/gpu-computing-the-basics/.

[6] P. Choudhary, A. Jain, and R. Baijal, Unravelling Airbnb predicting price for new listing,. *ArXiv*, 2018, [online] Available: https://arxiv.org/pdf/1805.12101.pdf.

[7] P. R. Kalehbasti, L. Nikolenko, and H. Rezaei, Airbnb price prediction using machine learning and sentiment analysis, *ArXiv*, 2019, [online] Available: https://arxiv.org/pdf/1907.12665.pdf.

[8] R. Mitchell, and E. Frank, Accelerating the XGBoost algorithm using GPU computing, *PeerJ Computer Science*, 3, e127, 2017, [online] Available: https://doi.org/10.7717/peerj-cs.127.

[9] A. Mishra, "XGBoost an efficient implementation of gradient boosting, *DataScience Foundation*, 2020, [online] Available: https://datascience.foundation/datatalk/xgboost-an-efficient-implementation-of-gradientboosting

[10] Airbnb Ratings Dataset. Kaggle, 2021, [online] Available: https://www.kaggle.com/samyukthamurali/airbnb-ratings-dataset?select=airbnb-reviews.csv

[11] Airbnb - Listings. Opendatasoft, 2020, [online] Available: https://public.opendatasoft.com/explore/dataset/airbnb-listings/table/?disjunctive.host_verifications&disjunctive.amenities&disjunctive.features.

[12] V. Morde, XGBoost algorithm: Long may she reign!, *Towards Data Science*. Medium, 2019, [online] Available: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long she-may-rein-edd9f99be63d.

[13] Nvidia-spark rapids. (n.d.). Home. Spark-Rapids, 2021, [online] Available: https://nvidia.github.io/sparkrapids/#:%7E:text=The%20RAPIDS%20Accelerator%20for%20Apache,processing%20via%20the%20RAPIDS%20libraries.&text=The%20RAPIDS%20Accelerator%20library%20also,GPU%20communication%20and%20RDMA%20capabilities.

[14] Nvidia. (n.d.-b). What's New in Deep Learning & Artificial Intelligence, 2021, [online] Available: https://www.nvidia.com/en-us/ai-data-science/spark-ebook/gpu-accelerated-spark-3/

[15] Nvidia. (n.d.). NVIDIA GPU Accelerated Solutions for Data Science, 2021, [online] Available: https://www.nvidia.com/en-us/deep-learning-ai/solutions/data-science/#:%7E:text=Data%20science%20workflows%20have%20traditionally,%E2%84%A2%20open%20source%20software%20libraries.

[16] RAPIDS. (n.d.). Open GPU Data Science | RAPIDS. Rapids.Ai., 2021, [online] Available: https://rapids.ai/about.html#:%7E:text=The%20RAPIDS%20suite%20of%20open,hardware%20and%20data%20science%20experience.

[17] RAPIDS, Open GPU Data Science (n.d.), 2021, [online] Available: https://rapids.ai/.

[18] GPU Accelerated Apache Spark (n.d.), 2021, [online] Available: https://www.nvidia.com/en-us/deep-learning-ai/solutions/data-science/apache-spark-3/.

Samyuktha Muralidharan

Samyuktha Muralidharan has a Master's in Information Systems from California State University, Los Angeles. She is currently working for Dine Brands Global as an Analyst in Applebee's Business Analytics team. She is involved with analyzing Applebee's overall performance, performance of campaigns/ tests and providing rich insights driving the growth of Applebee's business. Her research interests include big data analytics and predictive analytics.

Savita Yadav

Savita Yadav has completed her Master's in Information Systems from California State University, Los Angeles. She has worked as a Software Engineer in Java technology in different domains, including Banking and Pharmaceutical companies. She has experience using Python, SAS tools, R, and Big Data-related technologies, including Hadoop, Hive, Pig, and Spark, to design applications. Her research interests include big data, machine learning, and predictive analysis.

Jungwoo Huh

Jungwoo Huh was born in South Korea, in 1994. He received his B.S. degree in Electrical and Electronic Engineering from Yonsei University, Seoul, South Korea, in 2017, where he is currently pursuing his M.S. and Ph.D. degrees with the Multidimensional Insight Laboratory. His research interests include machine learning, computer vision, motion capture, and motion analysis.

Sanghoon Lee

Dr Lee is a Full-time Professor at the Department of EE, Yonsei University. He has been an Associate Editor of the IEEE Trans. Image Processing and an Editor of the JCN, and the Chair of the IEEE P3333.1 QA Working Group. He has served as the Technical Comm. and the General Chair of the IEEE IVMSP workshop, and a guest editor of IEEE Trans. Image Processing. He has received a special service award from IEEE Broadcast Technology and Signal Processing Society. His research interests include image/video QA, medical image processing, cloud computing, wireless multimedia communications, and wireless networks.

Jongwook Woo

Dr. Woo received his Ph.D. from USC and went to Yonsei University. He is a Professor at CIS Department of California State University Los Angeles and serves as a Technical Advisor of Teradata, Spark Technology Center and KSEA-SC. He has consulted companies in Hollywood. He has published more than 50 papers regarding Scalable Deep Learning, Big Data Analysis and Prediction. He has been awarded Teradata TUN faculty Scholarship and received grants from Databricks, NVidia, Amazon, IBM, Oracle, Microsoft, Cloudera, Hortonworks, SAS, QlikView, and Tableau.