OKU Fen Bilimleri Enstitüsü Dergisi
5(3): 1383-1392, 2022

OKU Journal of The Institute of Science and Technology, 5(3): 1383-1392, 2022

Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi

Osmaniye Korkut Ata University Journal of The Institute of Science and Technology

An Evaluation of Autoencoder Neural Network Role in IoT Edge Computing

Aygül TEKİN KAKIZ[1], Muhammed Talha KAKIZ[2*], Ramazan ÇOBAN[3]

[1]Osmaniye Korkut Ata Üniversitesi, Rektörlük, Uzaktan Eğitim Uygulama ve Araştırma Merkezi, 80000, Osmaniye
[2]Osmaniye Korkut Ata Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 80000, Osmaniye
[3]Çukurova Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 01250, Adana

[1]https://orcid.org/0000-0001-7372-0664
[2]https://orcid.org/0000-0003-4928-6559
[3]https://orcid.org/0000-0002-4505-0437
*Corresponding author: mtalhakakiz@osmaniye.edu.tr

Research Article

Article History:
Received: 16.12.2021
Accepted: 18.05.2022
Published online: 12.12.2022

Keywords:
Internet of things
Edge computing
Cloud computing
Autoencoder
Artificial neural networks

ABSTRACT

With rapid increase in numbers of connected Internet of Things (IoT) devices, huge amount of data is generated and sent to Cloud Computing nodes to be stored and analysed. Cloud computing is an effective paradigm for storage and data analysis since IoT devices are restricted machines in terms of energy, computation power and storage. Despite the advantages of cloud computing, it causes network congestion and latency due to generally located at long distances. Besides, security and privacy issues are also drawbacks of the cloud. Edge Computing is a promising system to eliminate the flaws of cloud computing by getting computational power closer to data sources. Edge Computing has more computation power than IoTD but lower than cloud computing. Although the deficiencies of cloud computing decrease with edge computing, they are not completely eliminated because computation intensive tasks still should be sent from edge to cloud resources. Since Autoencoder is an unsupervised neural network technique that learns to efficiently encode/compress input data and learns to efficiently decode it as closer to the original input, it is an ideal candidate for reducing data traffic and latency in edge computing and cloud computing. The main purpose of this paper is to investigate the studies using AE in edge computing and their performance implications with respect to network traffic, security, and delay. The performance results of the proposals that have used autoencoder between edge and cloud layer are evaluated in terms of eliminating big data, network traffic and accuracy.

Nesnelerin İnterneti Uç Bilişimde Otokodlayıcı Sinir Ağının Rolüne İlişkin Bir Değerlendirme

Araştırma Makalesi

Makale Tarihçesi:
Geliş tarihi: 16.12.2021
Kabul tarihi:18.05.2022
Online Yayınlanma: 12.12.2022

Anahtar Kelimeler:
Nesnelerin interneti
Uç bilişim
Bulut bilişim
Otokodlayıcı
Yapay sinir ağları

ÖZ

İnternete bağlı IoT cihazların sayısındaki hızlı artış ile çok büyük miktarda üretilen veri depolanmak ve analiz edilmek üzere Bulut Bilişim düğümlerine gönderilir. IoT cihazlar enerji, hesaplama gücü ve depolama açısından kısıtlı makineler olduğundan, Bulut Bilişim depolama ve veri analizi için etkili bir paradigmadır. Bulut Bilişimin avantajlarına rağmen, genellikle uzun mesafelerde konumlandığı için trafik sıkışıklığı ve gecikmelere neden olur. Bunun yanında, güvenlik ve gizlilik meseleleri de Bulut Bilişimin dezavantajlarındandır. Uç bilişim hesaplama gücünü veri kaynağına yaklaştırarak Bulut Bilişimin kusurlarını bertaraf edecek umut verici bir sistemdir. Uç Bilişim, IoT cihazdan daha fazla; Bulut Bilişimden ise daha az hesaplama gücüne sahip. Uç Bilişim ile birlikte Bulut Bilişimin olumsuzluklarının azalmasına rağmen, tamamen ortadan kalkmaz. Çünkü, yoğun hesaplamalı görevlerin hala uçtan bulut kaynaklarına gönderilmesi gerekir. Otokodlayıcı, girdi verisini etkili bir şekilde kodlayan/sıkıştıran ve orijinal girdi verisine daha yakın olacak şekilde kodu çözmeyi öğrenen

denetimsiz sinir ağı tekniğidir. Uç bilişim ve Bulut Bilişimdeki veri trafiği ve gecikmeyi azaltmak için ideal bir adaydır. Bu çalışmanın amacı, ağ trafiği, güvenlik ve gecikme açısından Otokodlayıcı yönteminin uç bilişimde kullanılan çalışmaları ve performans etkilerini araştırmaktır. Uç ve bulut katman arasında Otokodlayıcı kullanan çalışmaların performans sonuçları büyük veri, ağ trafiği ve doğruluk açısından değerlendirilmiştir.

To Cite: Kakız AT., Kakız MT., Çoban R. An Evaluation of Autoencoder Neural Network Role in IoT Edge Computing. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2022; 5(3): 1383-1392.

1. Introduction

Internet of Things (IoT) envisions to enable physical objects of everyday life to be able to see, hear, think, and talk by sensing, analyzing, and communicating with their environments (Al-Fuqaha et al., 2015). These objects can see and hear by means of sensors integrated (i.e., camera, temperature, gyroscope), can think and analyze the sensed data with a processing unit and finally can talk and interact with other objects with a communication interface. Since the objects can communicate with each other in the IoT concept, an unprecedented amount of data is generated by a wide variety of devices such as home appliances, vehicles, surveillance cameras, transportation, and manufacturing systems (Zanella et al., 2014).

The data collected from the real-world environment should be analyzed to extract useful information about the objects and their situations. With the relevant extracted information, possible future events can be predicted, or right decisions can be made about what the devices should perform if necessary. Thus, IoT turns into a paradigm that improves the quality of our daily life (Ge et al., 2018). However, processing large volumes of sensor data on IoT is relatively challenging because of restricted computational power and energy constraints of IoT devices as in Wireless Sensor Networks (WSN) (Akyildiz and Vuran, 2010). Therefore, Cloud Computing services have been proposed for data processing and analysis.

The main objective of this paper is to evaluate Autoencoder (AE) Neural Network role in IoT Edge Computing. To achieve this aim, we focus on the studies developed to be executed on edge devices and using AE between IoT and edge nodes or between edge and cloud nodes.

The rest of the paper is organized as follows. Section 2 explains CC and Edge Computing paradigms and why they are needed. Section 3 gives background information of autoencoder neural network. Section 4 evaluates autoencoder roles in EC with comparative examples. Finally, Section 5 concludes the paper and gives future research directions.

2. Material and Method

Cloud and Edge Computing

Cloud Computing (CC) is a promising way to perform computationally intensive IoT tasks because CC systems have more computational power and storage capacity (Shi and Dustdar, 2016). IoT devices collect data and send their data to distributed powerful CC nodes to be analyzed and stored, and then the IoT nodes or another device may be notified of predictions and decisions for optimization. For example, surveillance camera systems capture image data at the edge but cannot

process it for face recognition because image processing requires more computational power. However, they can send captured image data to CC machines thousands of miles away to be analyzed for face recognition; then, after data analysis, CC nodes share the information of who the detected face belongs to with relevant devices.

While CC has many benefits that meet the computational inadequacy of IoT, it also has several shortcomings: i) Long physical distances between CC nodes and IoT devices can cause major delays, which is not acceptable for delay sensitive applications (e.g., autonomous driving, highly interactive application) (Wang et al., 2020); ii) Sending large amount of data generated by IoT devices to the remote data centers will not work because 500 billion devices, according to Cisco, going to connect to the internet in 2030 (Cisco, 2016; Pan and McElhannon, 2017); iii) Very large volume of data transmission increases the pressure, density and traffic in the backbone network (Wang et al., 2020); iv) Sending data to CC also carries risks in terms of security and privacy (Shi and Dustdar, 2016).

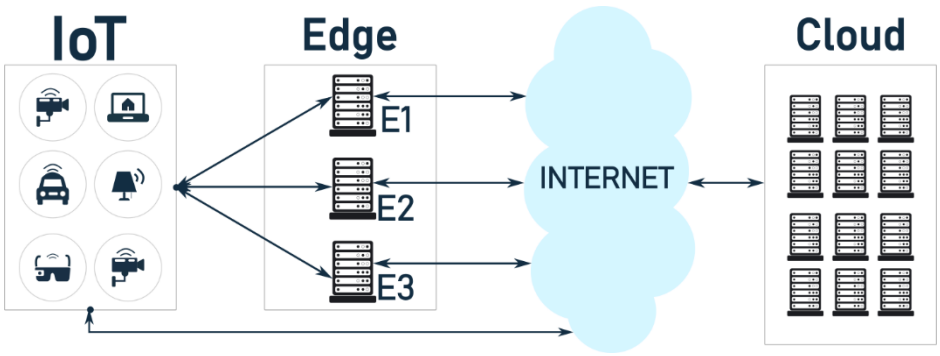

Figure 1. CC and EC system model architecture

To address aforementioned deficiencies and challenges of CC, Edge Computing (EC) has been proposed (Satyanarayanan et al., 2009). The main idea of EC is to bring computational resources closer to the edge IoT devices, which provides better services especially for delay-sensitive applications (Khan et al., 2019). Instead of sending generated data to central cloud devices which are far away from the edge of the network, IoT nodes can send data to nearby edge devices for data analysis. The data received from IoT devices is sent to the edge node, and after certain tasks are completed, the reduced data is sent to the cloud for integration. Edge servers connect via a private network or the Internet and are located at the edge of the network. They can be used for storage, data compression-decompression and computation as well as providing multimedia content (Ghosh and Grolinger, 2019). As a result, EC is a key enabler to tackle the problems of CC (i.e., latency, computational density of cloud devices and network congestion). The comparison between CC and ED is represented in Table 1 with different parameters.

EC reduces latency and traffic, improves user experience, and reduces dependency on the cloud. Therefore, industry and academia place emphasis on EC (Mach and Becvar, 2017; Ghosh and Grolinger, 2019). Edge servers are like a bridge between cloud and IoT devices. Note that it does not

mean that all data generated must be sent to the EC nodes. When needed, IoT nodes can directly send the data to the CC devices, or it is sent from EC to CC because EC machines have more computational power and storage capacity than IoT devices but less than CC machines. Therefore, computation intensive tasks cannot be handled by EC nodes, and they can be directly sent to CC, as represented in Figure 1.

Considering that EC devices may have difficulty in accomplishing Machine Learning (ML) tasks that require high computing power, there is a need for a system where EC and CC systems collaboratively work for data analysis. The mentioned system should aim to send the least amount of data to the cloud (e.g., integration and control data) in order to reduce latency and network congestion. Autoencoder (AE), an artificial neural network mostly used in ML and DL, is an effective way to achieve this goal.

Table 1. CC and EC comparison (Ullah et al., 2018)

Parameters	**CC**	**EC**
Delay	High	Low
Security and Privacy	Low	High
Computing Power	High	Limited
Access	Internet	Edge Network
Distance	Far	Close
Noticing the Location	No	Yes
Topology	Centralized	Distributed
Mobility Support	No	Yes

Autoencoder Neural Network

AE, a neural network that learns to encode data in an unsupervised manner (Ghosh and Grolinger, 2021), is a ML architecture in which the number of nodes at the input layer is equal to the number of output nodes, as shown in Figure 2, and the number of nodes in hidden layers is less than inputs and outputs. When the hidden nodes are less than inputs, the model is trained to learn the best coding of the inputs with hidden units for dimensionality reduction (Alpaydin, 2020). It is not only used for dimensionality reduction but also for many other reasons that will be explained in Section 4.

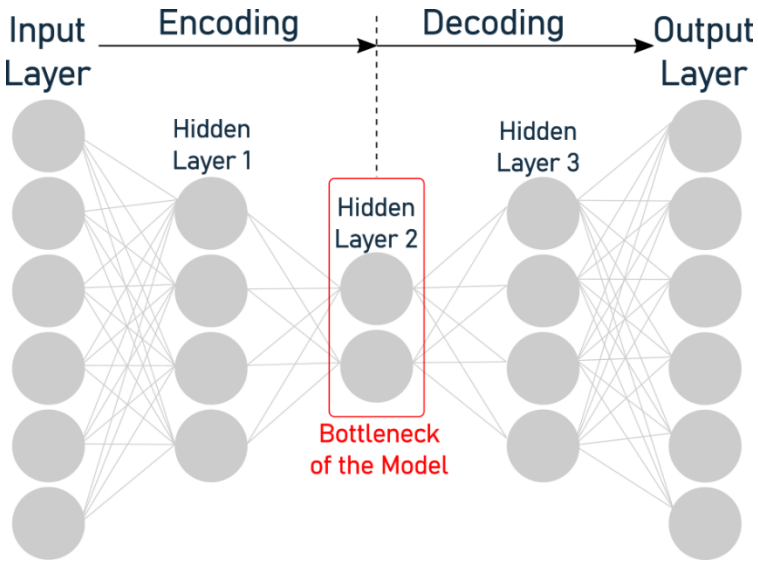

Figure 2. Autoencoder neural network

An AE consists of two parts, an encoder and a decoder and each of these parts may consist of one or more hidden layers. Since the encoder part of AE is responsible for reducing the data size, the number of neurons decreases from the input layer to the bottleneck of the model, as represented in Figure 3. On the other hand, the decoder side reconstructs the input values from the encoded data. Therefore, the decoder part consists of layers with an increasing number of neurons from the bottleneck of the model to the output layer. AE can be used for noise removal and anomaly detection, but it often serves as a preprocessing step for another ML task. This preprocess can be a dimensional reduction of input data (Ghosh and Grolinger, 2021).

Encoding data and then trying to reconstruct it may seem meaningless, but it is used for many different aims in a variety of applications. For example, in terms of compressing data, other data compression algorithms may perform more efficient than AE, but they cannot learn anything from the compressed data. However, AE tries to learn and prioritize some aspects of the input that resembles training data (Goodfellow et al., 2016).

3. Results and Discussion

Even though there exist different types of AEs which are used for different purposes, there are three main use cases of AE in EC, which are anomaly detection, noise removal, dimensionality reduction/data compression. In these use cases, encoder and decoder parts of AE can be located in distinct nodes such as EC, CC and IoT. It only depends on the aim of the use case. A histogram of the type of proposals in IEEE Xplore is shown in Figure 3. Let us take a deeper look at the use cases.

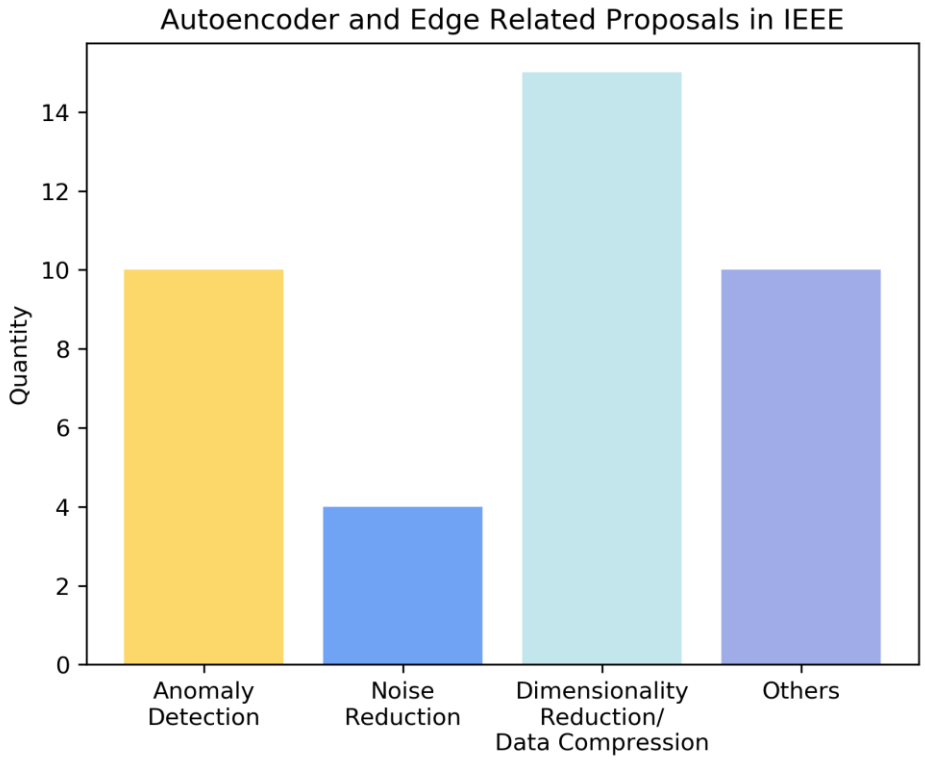

Figure 3. The histogram of edge and autoencoder related proposals in IEEE xplore

Anomaly Detection

The most common use case of AE in EC is detecting anomaly of input data. Since IoT devices are restricted nodes in terms of memory and computation, they cannot execute complex security algorithms and they are vulnerable to malware attacks. The input data sent from IoTs to EC or CC may include anomalies caused by the malicious software. Therefore, detecting any anomaly is an essential task with regards to accuracy of data analysis. There are many studies in the literature that are proposed for anomaly detection in EC, and they use different types of AEs to detect anomalies.

Tzagkarakis et al. have proposed a botnet attack detection method on EC node (Tzagkarakis et al., 2019), which is a sparsity representation framework detecting error rate between original input and reconstructed data. If the error rate of a sample is greater than a threshold value, it is extracted as an abnormal data. A similar approach is proposed by (Li et al., 2021) in which anomaly detectors are constructed by using LSTM (Long Short-Term Memory) AE. Original and reconstructed sequences are compared, and if the error is more than a threshold, then it is abnormal. Otherwise, it is considered as a normal data. Besides, detecting anomalies in cellular networks is another proposal in which ADM-Edge integrated into an NB-IoT (Narrowband IoT) tries to detect anomalies over a single data point (Savic et al., 2021). If a larger amount of time series is generated, it is sent to more computationally powerful Fog devices for anomaly detection.

Kim et al. proposed anomaly detection for industrial IoT (IIoT) by using AE model named Squeezed Convolutional Variational Autoencoder (Kim et al., 2018). The proposed method has been embedded in IIoT devices. Also, Park et al. embedded their proposed anomaly detection model in IoT devices to

identify electric motor failures. In this method, two AE structures have been used to obtain the best method (Park et al., 2021). Another anomaly detection method has been proposed to detect anomalies for Bridge Health Monitoring (Moallemi et al., 2022) with benchmarking Fully Connected AE and Convolutional AE. They also embedded the proposed method in IoT devices. Unlike previous studies, the proposal of anomaly detection in smart farming ecosystem (Adkisson et al., 2021) has not clearly indicated in which layer the method has been embedded. They have used unsupervised AE model.

Removing Noise

One of the most common uses of AEs is to remove noise from data. Thus, the data is transformed into a more suitable form for learning with ML models. This application usually takes place in the EC and the resulting representative compressed data is sent to the cloud for analysis. Since AEs represent data with smaller nodes by learning only useful information, noise is eliminated from the data by ignoring it. The type of AE that is mostly used for noise removal is Denoising AE.

A smart parking with user activity has been proposed by Kim et al. (Kim et al., 2021) that tries to eliminate noise from data generated by smartphone sensors. The data used in the study is sensed in the car, out of the car but it has also noise. They use Denoising AE for noise removal in smartphone and the reconstructed data from which the noise is removed is sent to EC for parking location and user activity analysis. Also, a similar approach has been adopted in (Feng at al., 2021) for noise removal from space launch system data. Embedded edge nodes in a rocket tries to remove noise to make space launch mission more reliable and secure. Besides, Auto-Key trains denoising AE to remove noise and obtain the repaired signal from the initial noisy one. Therefore, it accelerates the key generation based on gait in body area networks (Wu et al., 2020).

Unlike previous methods, PrivStream method injects noise to make data stream away from adversary attacks and uses an AE to realize data minimization (Wang et al., 2019). The proposed method is distributed on IoT and Edge devices.

Dimensionality Reduction/Data Compression

Considering the traffic density in the network, dimensionality reduction/data compression is one of the most important uses of AE in EC devices. The purpose of this use is to send the bottleneck, where the data is represented by fewer nodes, instead of sending all of the generated data to the network. Thus, the data traffic in the network will be reduced. The performance of the studies developed for this use case depends on the ability to learn from the compressed data and reach an accuracy close to that obtained with the original data.

To reduce network traffic, Ghosh et al. have developed an architecture by combining EC and CC (Ghosh and Grolinger, 2020). The encoder part of AE is located on EC and the decoder part is placed on CC. When sensor data is received from IoT devices, EC encodes the data and sends it to CC for data analysis. By this way, %80 data is reduced without significant loss in accuracy. Another method of dimensionality reduction with AE in EC has been proposed for online resource scheduling system

(Jiang et al., 2020). Stacked AE is used for compression and representation of high dimensional channel quality information in large scale mobile EC networks.

Trilla et al. proposed to compress vibration monitoring data up to 10 times without affecting the performance of the process (Trilla et. al., 2020). They use AE to realize this improving with three configurations: denoising, sparse and contractive. Besides, Lv et al. also compare the performance of stacked noise, stacked, stacked contractive, stacked sparse and deep belief AEs in terms of dimension reduction (Lv et al., 2021). They perform the comparison with respect to accuracy rate, false negative rate and false positive rate.

Preprocessing for another ML Task

AE can also be used as a preprocessing step before entering another ML model (L'heureux et al., 2017). This step can occur in two different ways; i) the bottleneck nodes of AE can be directly used by the ML model in EC or CC, ii) the compressed representation of data is reconstructed and then it is used as an input data of another ML model in EC or CC. Apart from these use cases, AE can also be used as a classification tool (AbdulsalamYa'u et al., 2019).

4. Conclusions

Autoencoder neural network model is used for many different applications with a variety of purposes. In this paper, we have briefly explained what autoencoder, edge computing and cloud computing are and why we need of autoencoder in edge and cloud. Also, we evaluated the role of autoencoder neural network model in IoT Edge Computing by giving the most common uses cases. In future studies, coding and decoding methods can be developed in studies using AE, and it can be tried to reach the ideal point of minimum energy and maximum accuracy.

Conflict of Interest Statement

The authors of the article declare that there is no conflict of interest between them.

Contribution Rate Statement Summary of Researchers

The authors declare that they have contributed equally to the article.

References

AbdulsalamYa'u G., Job GK., Waziri SM., Jaafar B. Sabon Gari NA., Yakubu IZ. Deep learning for detecting ransomware in edge computing devices based on autoencoder classifier. 2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), 13-14 December 2019, page no: 240–243, Mysuru, India.

Adkisson M., Kimmell JC., Gupta M., Abdelsalam M. Autoencoder-based anomaly detection in smart farming ecosystem. 2021 IEEE International Conference on Big Data (Big Data), 15-18 December 2021, page no: 3390-3399, Orlando, FL, USA.

Akyildiz IF., Vuran MC. Wireless sensor networks. 1st ed. UK: John Wiley & Sons; 2010.

Alpaydın E. Introduction to machine learning. 3rd ed. London: MIT Press; 2014.

Al-Fuqaha A., Guizani M., Mohammadi M., Aledhari M., Ayyash M. Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials 2015; 17(4): 2347–2376.

Challenges in real-world edge computing architecture, https://www.cisco.com/c/en/us/solutions/internet-of-things/iot-edge-computing-architecture.html. Cisco, Accessed: 2021-08-15.

Feng Y., Liu Z., Chen J., Lv H., Wang J., Yuan J. Make the rocket intelligent at iot edge: stepwise gan for anomaly detection of lre with multi-source fusion. IEEE Internet of Things Journal 2021; 9(4): 3135-3149.

Ge M., Bangui H., Buhnova B. Big data for internet of things: a survey. Future Generation Computer Systems 2018; 87: 601–614.

Ghosh AM., Grolinger K. Deep learning: edge cloud data analytics for iot. 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), 5-8 May 2019, page no: 1-7, Canada.

Ghosh AM., Grolinger K. Edge-cloud computing for internet of things data analytics: embedding intelligence in the edge with deep learning. IEEE Transactions on Industrial Informatics 2020; 17(3): 2191–2200.

Goodfellow I., Bengio Y., Courville A. Deep learning. 1st ed. UK: MIT press; 2016.

Jiang F., Wang K., Dong L., Pan C., Yang K. Stacked autoencoder-based deep reinforcement learning for online resource scheduling in large-scale mec networks. IEEE Internet of Things Journal 2020; 7(10): 9278–9290.

Khan WZ., Ahmed E., Hakak S., Yaqoob I., Ahmed A. Edge computing: a survey, Future Generation Computer Systems 2019; 97: 219–235.

Kim D., Yang H., Chung M., Cho S., Kim H., Kim M., Kim K., Kim E. Squeezed convolutional variational autoencoder for unsupervised anomaly detection in edge device industrial internet of things. 2018 IEEE International Conference on Information and Computer Technologies (ICICT), 23-25 March 2018, page no: 67-71, DeKalb, IL, USA.

Kim S., Park S., Lee SH., Yang T. Smart parking with learning aided user activity sensing based on edge computing. 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), 9-12 January 2021, page no: 1–2, Virtual Conference.

Li R., Li Q., Zhou J., Jiang Y. Adriot: An edge-assisted anomaly detection framework against iot-based network attacks. IEEE Internet of Things Journal 2021; 9(13): 10576-10587.

Lv Z., Qiao L., Li J., Song H. Deep-learning-enabled security issues in the internet of things. IEEE Internet of Things Journal 2021; 8(12): 9531-9538.

L'heureux A., Grolinger K., Elyamany HF., Capretz MA. Machine learning with big data: challenges and approaches. IEEE Access 2017; 5: 7776–7797.

Mach P., Becvar Z. Mobile edge computing: a survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials 2017; 19(3): 1628–1656.

Moallemi A., Burrello A., Brunelli D., Benini L. Exploring scalable, distributed real-time anomaly detection for bridge health monitoring. IEEE Internet of Things Journal; 9(18): 17660-17674.

Pan J., McElhannon J. Future edge cloud and edge computing for internet of things applications. IEEE Internet of Things Journal 2017; 5(1): 439–449.

Park Y., Kim M. Design of cost-effective auto-encoder for electric motor anomaly detection in resource constrained edge device. 2021 IEEE 3rd Eurasia Conference on IoT, Communication and Engineering (ECICE), 29-31 October 2021, page no: 241-246, Yunlin, Taiwan.

Satyanarayanan M., Bahl P., Caceres R., Davies N. The case for vm-based cloudlets in mobile computing. IEEE Pervasive Computing 2009; 8(4): 14–23.

Savic M., Lukic M., Danilovic D., Bodroski Z., Bajovi´c D., Mezei I., Vukobratovic D., Skrbic S., Jakoveti´c D. Deep learning anomaly detection for cellular iot with applications in smart logistics. IEEE Access 2021; 9: 59406–59419.

Shi W., Dustdar S. The promise of edge computing. Computer 2016; 49(5): 78–81.

Trilla A., Miralles D., Fernández V. Pushing distributed vibration analysis to the edge with a low-resolution companding autoencoder: industrial iot for phm. In Annual Conference of the PHM Society, 9-13 November 2020, page no: 1-9, Virtual Conference.

Tzagkarakis C., Petroulakis N., Ioannidis S. Botnet attack detection at the iot edge based on sparse representation. 2019 Global IoT Summit (GIoTS), 17-21 June 2019, page no: 1–6, Denmark.

Ullah R., Ahmed SH., Kim BS. Information-centric networking with edge computing for iot: research challenges and future directions. IEEE Access 2018; 6: 73465–73488.

Wang D., Ren J., Xu C., Liu J., Wang Z., Zhang Y., Shen X. PrivStream: enabling privacy-preserving inferences on iot data stream at the edge. IEEE 21st International Conference on High Performance Computing and Communications, 10-12 August 2019, page no: 1290-1297, Zhangjiajie, China.

Wang F., Zhang M., Wang X., Ma X., Liu J. Deep learning for edge computing applications: a state-of-the-art survey. IEEE Access 2020; 8: 58322–58336.

Wu Y., Lin Q., Jia H., Hassan M., Hu W. Auto-Key: using autoencoder to speed up gait-based key generation in body area networks. Proceedings of the ACM on Interactive Mobile Wearable and Ubiquitous Technologies 2020; 4(1): 1–23.

Zanella A., Bui N., Castellani A., Vangelista L., Zorzi M. Internet of things for smart cities. IEEE Internet of Things journal 2014; 1(1): 22–32.