

Homogenous and Heterogenous Parallel Clustering: An Overview

Ahmed Ibrahim
Computer Science Department
Boston University, MA, USA
London, Canada
aibrahim@bu.edu

Rokaya Hassanien
Electrical, Computer, and Biomedical Engineering
Ryerson University
Toronto, Canada
rhassanien@ryerson.ca

Abstract—Recent advances in computer architecture and networking opened the opportunity for parallelizing the clustering algorithms. This divide-and-conquer strategy often results in better results to centralized clustering with a much-improved time performance. This paper reviews key parallel clustering and provides insight into their strategy. The review brings together disparate attempts in parallel clustering to provide a comprehensive account of advances in this emerging field.

Keywords: Parallel clustering, Split Strategies, and Quality Measures.

1. Introduction

Many businesses have adopted various machine learning approaches [1]-[15], with varying definitions of clusters, classes, boundaries, constraints, and similarity measurements. Due to the massive volume of data in diverse applications such as data mining and bioinformatics, the issue of scalability and dealing with large and high dimensional datasets is particularly significant. Clustering such big and high-dimensional datasets in unsupervised machine learning is a current difficulty, making typical centralized clustering algorithms ineffective. Clustering techniques must be parallelized by separating computations so that each node can do a piece of the clustering operation in parallel with the other processing nodes. As a result, faster results are obtained than with a centralized node system. On both the local and global levels, parallel clustering is carried out. On a local level, nodes cluster autonomously, and local models (i.e., representatives) are generated at each node. After the local clustering is completed, a global model is created. Parallelization of clustering algorithms can be categorized into main categories, homogeneous or heterogeneous parallelization depending on the types of clustering algorithms used. An overview of several subjects that fall under the umbrella of parallel data clustering is covered in this study to provide a thorough account of advances in these new disciplines. The following is a breakdown of the paper's structure: Section 2 discusses several clustering parallelization algorithms. Section 3 discusses future smart systems using parallelization strategies. Finally, in section 4, we wrap up the paper with a summary.

2 Parallel Data Clustering

Efficient parallel and large-scale algorithms [16]-[31] minimize the calculation time while clustering big data sets. Splitting the clustering task among P nodes is required for parallel data clustering. The computing process must be balanced for each node in clustering parallelization; the balancing strategy reduces communication costs during the clustering phase.

2.1 Parallel Homogenous Clustering Algorithms

This section focuses on homogenous parallel clustering algorithms. Each node invokes the same clustering strategy, and homogenous local prototypes are generated at each node.

2.1.1 Parallel k -Means (PKM) Algorithm

The PKM [24][25] is an iterative parallel version of the KM using MPI reduction routines [32].

Algorithm: Parallel k -means (PKM)

Inputs: The dataset X, # of clusters k , # of nodes P

Outputs: k clusters

- Split the dataset X on the P nodes.
- N_0 randomly chooses initial centroids and replicates them to other P-1 nodes.

Repeat

Stage1: local nodes N_p calculates the distance of the local vectors to the k centres.

Stage2: Local vectors are mapped to the closest centroid, the local *objective function* is evaluated

Stage3: The local representatives are reduced to produce the global representatives using MPI [32]

Until Convergence

Fig. 1. The PKM Algorithm

2.1.2 Parallel Fuzzy c -Means (PFCM) Algorithm

The PFCM [26] presented in Fig.2 distributes the fuzzy membership function *on P* nodes with a local model containing only the local memberships. The global reduction of local centroids produces global centroids with calls to MPI reduction routines [32]. This step is repeated until convergence is determined by minimizing the distance to global centroids.

Algorithm: Parallel Fuzzy c -means (PFCM)

Inputs: The dataset X, the weighting exponent m , # of clusters k and # of nodes P

Outputs: k clusters

Distribute X on the P nodes such that each node calculates the initial local membership matrix

Repeat

Stage1: Node N_p calculates centroids for the k clusters with global reductions with 2 MPI calls.

Stage2: Node N_p computes the local membership of each local vector to the k clusters using the updated global k centroids and assigns local points to new clusters

Stage3: Global reduction is performed to produce global J value

Until Convergence

Fig. 2. The PFCM Algorithm

2.1.3 Parallel k -Windows (PK-windows) Algorithm

The k -windows algorithm uses the concept of windows to determine clusters with the Orthogonal Range Search (ORS) algorithm described in Fig.3. In the PK-windows algorithm [21], the range search

procedure requires the most computational effort; thus, a parallel algorithmic scheme that uses a Multi-Dimensional Binary Tree [33] with a Server-Slave model for a range search [34] is presented in Fig.4.

Algorithm: Orthogonal Range Search (ORS)

Inputs: The tree TR , the i^{th} coordinate and the d -range query Q

Outputs: SS

$SS = \{ \}$

Stage1: x_r is the root of TR

Stage2: Examine the tree TR , if $x_r \in Q$, then Add x_r to SS , else recursively explore the left and right trees of x_r wrt to next coordinate $i+1$

Fig.3. The ORS

Algorithm: Parallel k -windows

Inputs: The dataset X , # of windows l , and the d -ranges area a .

Outputs: k clusters

The l centroids are chosen with initial d -ranges windows W_i , $i=1,\dots,l$ centred on these initial centroids with area a .

Phase1: Repeat

Stage 1: Repeat

- The master N_i ; sends a “*start new sub-search*” message to the ideal slave N_j ; $i \neq j$
- At the slave node, if a new sub-search is necessary,

The slave node N_j sends a “*necessary new sub-search*” message to the master node N_i and spawns to another ideal node, otherwise
Return the set of vectors that lie within the given d -range query Q .

Until no new sub-search messages are desired

Stage 2: New centroids are calculated as new d -ranges

Until each window has no significant increment of vectors

Phase2, Phase3 and Phase4 are typical in the classical k -windows algorithm [21].

Fig.4 The Parallel k -windows Algorithm

2.1.4 Distributed Clustering using Principal Component Analysis (DCPCA) Algorithm

Figure 5 depicts the Collective Principal Component Analysis (CPCA) process. The computations for PCA could be done locally in a parallel environment, reducing the quantity of data transfer and processing at one central node. The distributed clustering approach [35] works with a provided centralized algorithm B module and respects the user's choice of any local clustering algorithm. Figure 6 depicts a distributed CPCA-based system.

Algorithm: Collective Principal Component Analysis (CPCA)

Inputs: The distributed vertical partitioned local data X_p

Outputs: Global Principal Components of the original data X

Stage1: Node N_p performs a PCA algorithm locally and selects dominant eigenvectors.

Stage2: Each node N_p sends projected data with the eigenvectors to the facilitator

Stage3: The facilitator combines the data received from all the nodes and performs PCA on the global set, identifies the dominant eigenvectors, and transforms them to the original space.

Fig.5. The CPCA Algorithm

Algorithm: Distributed Clustering CPCA-based

Inputs: The distributed datasets $X_p; p=0,1,..,P-1$

Outputs: k clusters

Stage1: Node N_p performs PCA locally and projects the local data on the local PCs

Stage2: Node N_p applies the clustering algorithm B.

Stage3: Each cluster's representative points are selected at every node N_p . Let I_p is the set of indices at node N_p , i.e., the chosen representative points.

Stage4: All nodes communicate and send the data rows to the facilitator node.

Stage5: The facilitator performs the global PCA and transmits the global PCs to all nodes.

Stage6: Node N_p maps the local data to the global PCs and performs B clustering.

Stage7: Node N_p communicates and sends a sketch of the local clusters to the facilitator node.

Stage8: The facilitator combines the different local descriptions (models) obtained from the local nodes to construct global clusters.

Fig.6. The Distributed Clustering CPCA-based Algorithm

2.1.5 Distributed Density-Based Clustering (DDBC) Algorithm

In the DDBC algorithm [36], the local model is created using two local models, $REP_{k\text{-means}}$ and REP_{scor} [36][37], using the notion of specified core points. The $REP_{k\text{-means}}$ based on local specific core points at node N_p is acquired using the k -means as shown in Fig.7. Each local cluster S_i in REP_{scor} is characterized by a full set of specific core points $Scor_{S_i}$.

Algorithm: DBSCAN Clustering using k -means

Inputs: local cluster $S_i, i=1,2,..,k$, using centralized DBSCAN at local nodes N_p

Outputs $REP_{k\text{-means}}$

Stage1: Apply the local clustering S_i runs on the N_p , re-clustered using k -means, the set of $|Scor_{S_i}|$ centroids $c_{i1}, c_{i2}, .., c_{i|Scor_{S_i}|}$ is shaped

Stage2: centroid $c_{ij}; j=1,2,..,|Scor_{S_i}|$ is allocated with a ϵ -range value to indicate the area c_{ij} .

Stage3: The k clusters will have a local model as:

$$LocalModel_p = \bigcup_{i=1,..,k} \bigcup_{j=1,..,|Scor_{S_i}|} \{(c_{ij}, \epsilon_{c_{ij}})\}$$

Fig. 7. Local DBSCAN using k -means

Each local model comprises k local clusters with matching representatives, and the facilitator node creates a global model using local models. Every local representative develops its own cluster in the DBSC algorithm. Figure 8 depicts the DDBC algorithm.

Algorithm: Distributed Density-Based Clustering

Inputs: local data X_p , Eps_{local} , and $MinPts_{Local}$

Outputs: k clusters

Stage1: Node N_p execute the DBSCAN using Eps_{local} and $MinPts_{Local}$ and generates local models

Stage2: N_p sends the local model models to the facilitator node.

Stage3: facilitator executes a global DBSCAN with global $MinPts_{global}$ and Eps_{global}

Stage4: facilitator sends the global model to nodes N_p .

Stage5: Node N_p re-assign local points, the two clusters (i.e., independent) are merged

Fig. 8. The DDBC Algorithm

2.2 Parallel Heterogeneous Clustering Algorithms

Heterogeneous clustering methods are utilized in cascade (i.e., end-result level) to cluster the dataset in parallel hybrid clustering. This cascading approach is mostly used to increase the quality of solutions created by a previous method (s). The hybrid clustering in the parallel mode is accomplished by either communicating with all the nodes or interacting with only facilitator nodes.

2.2.1 Parallel Hybrid PDDP and k -means (PDDP-KM) Algorithm

Figure 9 depicts the multiple stages in the parallel PDDP method [27]. The PKM improves the clustering solutions obtained by the parallel PDDP algorithm. End-result coordination between the concurrent PDDP and PKM underpins this hybrid combination. This collaboration technique primarily improves the solutions shaped by the PDDP algorithm and provides a strong starting point for the k-means method. The parallel PDDP-KM algorithm is shown in Fig. 10.

Parallel PDDP Algorithm

Inputs: The matrix M, the tree *height*, # nodes P

Outputs: k clusters

- The entire matrix M is the tree root
- Split the matrix M

Stage1:

For level $i = 1$ to $height$

For cluster S_i at level i

If S_i is a *singleton*, then process the next cluster; **otherwise**, N_p computes the local mean vector c^p_i of the cluster S_i , and the local leading eigenvector

- Global centroid c_i and leading eigenvector \mathbf{u} are produced using MPI reduction call [32].

For each data point x in S_i

If $(\mathbf{u} \cdot \mathbf{x}) \geq 0$, allocate x to the left child of S_i , otherwise allocate x to the right child of S_i

Stage2: the desired set of k clusters is located at the leaf nodes.

Fig. 9. PDDP

Algorithm: Parallel Hybrid PDDP - KM Clustering

Inputs: The matrix M, the height of the tree, # nodes P, and # clusters k

Outputs: k clusters

Split M on the parallel P nodes

Stage1: Use the PDDP to produce k centroids

Stage2: Adopt the initial k centroids as input to the PKM

Stage3: PKM produces global clusters k

Fig. 10. Parallel PDDP- KM Algorithm

3 Parallelization and Smart Solutions

In the era of the Internet of Things (IoT), various methods have been developed in Outlier detection [38][42], Recommendation Systems [43]-[48], Cyber Attacks Detection [49]-[52], Smart Systems [53]-[56], and forecasting methods [59]-[64]. Although centralized approaches in these fields have shown promises in improving the entire decision-making process using real-data or simulated data [65][67], scalability of these methods is a crucial step in real-time applications. Thus, applying split analysis using federated learning is mandatory to ensure proper implementations.

4 Conclusion and Future Directions

The scalability of traditional clustering methods is questioned when there is a need to cluster extensive and high-dimensional data. Parallel data clustering is an effective solution to cluster such enormous data. Several parallel clustering algorithms were presented in this paper; their clustering strategies were reviewed. New parallelization strategies have generated more efficient and challenging tasks. Different communication protocols (with different order of communication cost) have been established between nodes (or nodes and a facilitator node) to send local models or local data or local representatives and construct the final global model. Obviously, with the current trend in parallelizing computer architecture, it is more affordable and sensible to solve highly computationally intensive problems such as clustering in a parallel manner. Future directions would include the investigations of parallel performance measures that are well fit to the parallel environment compared to centralized measures [68].

References

- [1] Kashef, R., & Kamel, M. S. (2010). Cooperative clustering. *Pattern Recognition*, 43(6), 2315-2329.
- [2] Kashef, R., & Kamel, M. S. (2009). Enhanced bisecting k-means clustering using intermediate cooperation. *Pattern Recognition*, 42(11), 2557-2569.
- [3] Kashef, R. (2008). Cooperative clustering model and its applications.
- [4] Ibrahim, A., Rayside, D., & Kashef, R. (2014, May). Cooperative based software clustering on dependency graphs. In 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-6). IEEE.
- [5] Kashef, R., & Kamel, M. S. (2007, October). Cooperative partitional-divisive clustering and its application in gene expression analysis. In 2007 IEEE 7th International Symposium on BioInformatics and BioEngineering (pp. 116-122). IEEE.
- [6] Kashef, R., & Kamel, M. S. (2007, October). Hard-fuzzy clustering: a cooperative approach. In 2007 IEEE International Conference on Systems, Man and Cybernetics (pp. 425-430). IEEE.
- [7] Kashef, R., & Kamel, M. S. (2008, June). Efficient bisecting k-medoids and its application in gene expression analysis. In International Conference Image Analysis and Recognition (pp. 423-434). Springer, Berlin, Heidelberg.
- [8] Abu-Zeid, N., Kashif, R., & Badawy, O. M. (2012, November). Immune based clustering for medical diagnostic systems. In 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT) (pp. 372-375). IEEE.
- [9] Hass, G., Simon, P., & Kashef, R. (2020, December). Business Applications for Current Developments in Big Data Clustering: An Overview. In 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 195-199). IEEE.
- [10] Kashef, R. (2021). A boosted SVM classifier trained by incremental learning and decremental unlearning approach. *Expert Systems with Applications*, 167, 114154.
- [11] Sewani, H., & Kashef, R. (2020). An autoencoder-based deep learning classifier for efficient diagnosis of autism. *Children*, 7(10), 182.
- [12] H. Sharma, H. Sewani, R. Garg and R. Kashef, "Face Mask Detection: A Real-Time Android Application Based on Deep Learning Modeling," 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2021, pp. 0106-0112, doi: 10.1109/IEMCON53756.2021.9623222.
- [13] Alam, F., Kashef, R., & Jaseemuddin, M. (2021, April). Enhancing The Performance of Network Traffic Classification Methods Using Efficient Feature Selection Models. In 2021 IEEE International Systems Conference (SysCon) (pp. 1-6). IEEE.
- [14] Kashef, R. (2022). ECNN: Enhanced convolutional neural network for efficient diagnosis of autism spectrum disorder. *Cognitive Systems Research*, 71, 41-49.
- [15] Kashef, R., & Kamel, M. S. (2008, June). Towards better outliers detection for gene expression datasets. In 2008 International Conference on Biocomputation, Bioinformatics, and Biomedical Technologies (pp. 149-154). IEEE.
- [16] Kashef, R., & Kamel, M. (2006, November). Distributed cooperative hard-fuzzy document clustering. In Proceedings of the Annual Scientific Conference of the LORNET Research Network.
- [17] Kashef, R., & Nirajan, A. (2017, December). Handling Large-Scale Data Using Two-Tier Hierarchical Super-Peer P2P Network. In Proceedings of the International Conference on Big Data and Internet of Thing (pp. 52-56).

- [18] Manjunath, Y. S. K., & Kashef, R. F. (2021). Distributed clustering using multi-tier hierarchical overlay super-peer peer-to-peer network architecture for efficient customer segmentation. *Electronic Commerce Research and Applications*, 47, 101040
- [19] Kashef, R., & Warraich, M. (2020). Homogeneous Vs. Heterogeneous Distributed Data Clustering: A Taxonomy. In *Data Management and Analysis* (pp. 51-66). Springer, Cham
- [20] Kashef, R., & Kamel, M. S. (2008, September). Distributed Peer-to-Peer Cooperative Partitional-Divisive Clustering for gene expression datasets. In *2008 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology* (pp. 143-150). IEEE
- [21] Alevizos, P., and D. Tasoulis and M. Vrahatis. “*Parallelizing the unsupervised k-windows clustering Algorithm*”. *Lecture Notes in Computer Science*, Vol. 3019, pp: 225-232, 2004
- [22] Petrosino, A. and M. Verde, “*P-AFLC: a Parallel Scalable Fuzzy Clustering Algorithm*”, ICPR2004, Proc. of the 17th International Conference on. Pattern Recognition, pp: 809-812, 2004.
- [23] Pizzuti, C., and D. Talia, “*P-AutoClass: Scalable Parallel Clustering for Mining Large Data Sets*”, *IEEE Transactions on Knowledge and Data Engineering*, Vol.15, Issue 3, pp: 629-641, 2003.
- [24] Stoffel, K., and A. Belkoniene, “*Parallel K-Means Clustering for Large Data Sets*”, Proceedings Euro-Par ’99, LNCS 1685, pp: 1451-1454, 1999.
- [25] Dhillon, I. and D. Modha. “*A Data Clustering Algorithm on Distributed Memory Multiprocessors*”, *Large Scale Parallel Data Mining*, LNCS Vol. 1759, Springer, 2000.
- [26] Kwok, T., K. Smith, S. Lozano, and D. Taniar. “*Parallel Fuzzy c-Means Clustering for Large Data Sets*”, *Proceedings EUROPAR02*, vol. 2400 of *LNCS*, pp: 365-374, 2002.
- [27] Xu, S., and J. Zhang, “*A Hybrid Parallel Web Document Clustering Algorithm and its Performance Study*”, *Journal of Supercomputing*, Vol. 30, Issue 2, pp: 117-131, 2004.
- [28] Jiménez, D., and V. Vidal, “*Parallel Implementation of Information Retrieval Clustering Models*”, *Lecture Notes in Computer Science*, Vol. 3402, pp: 129-141, 2005.
- [29] Hung, Z., “*k-means-type Algorithms on Distributed Memory Computer*”, *International Journal of High-Speed Computing*, Vol. 11, pp: 75-91, 2000.
- [30] Tasoulis, D., and M. Vrahatis. “*Unsupervised Distributed Clustering*”, *IASTED International Conference on Parallel and Distributed Computing and Networks*, pp: 347-351, 2004.
- [31] Datta, S., C. Giannella, and H. Kargupta. “*K-means clustering over peer-to-peer networks*”. In *Workshop on High Performance and Distributed Mining (HPDM05)*. SIAM International Conference on Data Mining (SDM05), 2005.
- [32] Gropp, W., E. Lusk and A. Skjellum, “*Using MPI, Portable Parallel Programming with Message Passing Interface*”. The MIT Press, Cambridge, MA, 1996.
- [33] Alevizos, P., “*An algorithm for Orthogonal Range Search in $d \geq 3$ dimensions*”, proceeding of the 14th European workshop on Computational Geometry, 1998.
- [34] Alevizos,P., B. Boutsinas, D. Tasoulis, M. Vrahatis, “*Improving the Orthogonal Range Search k-windows clustering algorithm*”, Proceedings of the 14th IEEE Int. Conference on Tools with Artificial Intelligence, 2002.
- [35] Kargupta, H., Huang, W., Sivakumar, K., Park, B. H., & Wang, S. (2000, September). Collective principal component analysis from distributed, heterogeneous data. In *European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 452-457).
- [36] Ester, M., H. Kriegel, J. Sander, and X. Xu,” *A Density-Based Algorithm for DiscoveringClusters in Large Spatial Databases with Noise*”. In: Proc. of 2nd Int'l Conf. On Knowledge Discovery and Data Mining pp:226–231,1996.
- [37] Januzaj, E., H. Kriegel, M. Pfeifle. “*Towards Effective and Efficient Distributed Clustering*”, Proceedings International Workshop on Clustering Large Data Sets (ICDM), pp: 49-58, 2003.
- [38] Kashef, R., & Kamel, M. S. (2008, June). Towards better outliers detection for gene expression datasets. In *2008 International Conference on Biocomputation, Bioinformatics, and Biomedical Technologies* (pp. 149-154). IEEE.
- [39] Kashef, R. F. (2018, January). Ensemble-based anomaly detetction using cooperative learning. In *KDD 2017 Workshop on Anomaly Detection in Finance* (pp. 43-55). PMLR.
- [40] Li, M., Kashef, R., & Ibrahim, A. (2020). Multi-level clustering-based outlier’s detection (MCOD) using self-organizing maps. *Big Data and Cognitive Computing*, 4(4), 24
- [41] Close, L., & Kashef, R. (2020). Combining artificial immune system and clustering analysis: A stock market anomaly detection model. *Journal of Intelligent Learning Systems and Applications*, 12(04), 83.
- [42] Kashef, R., Gencarelli, M., & Ibrahim, A. (2020). Classification of Outlier’s Detection Methods Based on Quantitative or Semantic Learning. In *Combating Security Challenges in the Age of Big Data* (pp. 45-59). Springer, Cham
- [43] Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., & Kashef, R. (2020). Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *applied sciences*, 10(21), 7748.

- [44] Nawara, D., & Kashef, R. (2020, September). IoT-based Recommendation Systems—An Overview. In 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) (pp. 1-7). IEEE.
- [45] Kashef, R. (2020). Enhancing the Role of Large-Scale Recommendation Systems in the IoT Context. *IEEE Access*, 8, 178248-178257
- [46] Yeh, T. Y., & Kashef, R. (2020). Trust-Based collaborative filtering recommendation systems on the blockchain. *Advances in Internet of Things*, 10(4), 37-56.
- [47] Nawara, D., & Kashef, R. (2021, April). Deploying Different Clustering Techniques on a Collaborative-based Movie Recommender. In 2021 IEEE International Systems Conference (SysCon) (pp. 1-6). IEEE
- [48] D. Nawara and R. Kashef, "Context-Aware Recommendation Systems in the IoT Environment (IoT-CARS)—A Comprehensive Overview," in *IEEE Access*, vol. 9, pp. 144270-144284, 2021, doi: 10.1109/ACCESS.2021.3122098.
- [49] Ebrahimian, M., & Kashef, R. (2020). Detecting shilling attacks using hybrid deep learning models. *Symmetry*, 12(11), 1805.
- [50] Ebrahimian, M., & Kashef, R. (2020, December). Efficient Detection of Shilling's Attacks in Collaborative Filtering Recommendation Systems Using Deep Learning Models. In 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 460-464). IEEE.
- [51] K. Mohammed, A. H., Jebamikyous, H., Nawara, D., & Kashef, R. (2021). IoT cyber-attack detection: A comparative analysis. In Int. Conference on Data Science, E-learning and Information Systems 2021 (pp. 117-123).
- [52] Soleymanzadeh R, Aljasim M, Qadeer MW, Kashef R. Cyberattack and Fraud Detection Using Ensemble Stacking. *AI*. 2022; 3(1):22-36. <https://doi.org/10.3390/ai3010002>
- [53] Karami, Z., & Kashef, R. (2020). Smart transportation planning: Data, models, and algorithms. *Transportation Engineering*, 2, 100013.
- [54] Kashef, R. (2020). Adopting Big Data Analysis in the Agricultural Sector: Financial and Societal Impacts. In *Internet of Things and Analytics for Agriculture*, Volume 2 (pp. 131-154). Springer, Singapore
- [55] Shirazi, Z. A., de Souza, C. P., Kashef, R., & Rodrigues, F. F. (2020). Deep learning in the healthcare industry: theory and applications. In *Computational intelligence and soft computing applications in healthcare management science* (pp. 220-245). IGI Global
- [56] H. -H. Jebamikyous and R. Kashef, "Autonomous Vehicles Perception (AVP) Using Deep Learning: Modeling, Assessment, and Challenges," in *IEEE Access*, vol. 10, pp. 10523-10535, 2022, doi: 10.1109/ACCESS.2022.3144407.
- [57] Pano, T., & Kashef, R. (2020). A complete VADER-based sentiment analysis of bitcoin (BTC) tweets during the era of COVID-19. *Big Data and Cognitive Computing*, 4(4), 33.
- [58] Tan, X., & Kashef, R. (2019, December). Predicting the closing price of cryptocurrencies: a comparative study. In Proc. of the Second International Conference on Data Science, E-Learning and Information Systems (pp. 1-5).
- [59] Ibrahim, A., Kashef, R., Li, M., Valencia, E., & Huang, E. (2020). Bitcoin network mechanics: Forecasting the btc closing price using vector auto-regression models based on endogenous and exogenous feature variables. *Journal of Risk and Financial Management*, 13(9), 189.
- [60] Ibrahim, A., Kashef, R., & Corrigan, L. (2021). Predicting market movement direction for bitcoin: A comparison of time series modeling methods. *Computers & Electrical Engineering*, 89, 106905
- [61] Tobin, T., & Kashef, R. (2020, June). Efficient Prediction of Gold Prices Using Hybrid Deep Learning. In *International Conference on Image Analysis and Recognition* (pp. 118-129). Springer, Cham.
- [62] Ibrahim, A. F., Corrigan, L., & Kashef, R. (2020). Predicting the Demand in Bitcoin Using Data Charts: A Convolutional Neural Networks Prediction Model. In 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-4). IEEE
- [63] Pano, T., & Kashef, R. (2020, September). A Corpus of BTC Tweets in the Era of COVID-19. In 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) (pp. 1-4). IEEE
- [64] Aasi, B., Imtiaz, S. A., Qadeer, H. A., Singarajah, M., & Kashef, R. (2021, April). Stock Price Prediction Using a Multivariate MultiStage LSTM: A Sentiment and Public Engagement Analysis Model. In 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) (pp. 1-8). IEEE
- [65] Attanayake, N., Kashef, R. F., & Andrea, T. (2014, May). A simulation model for a continuous review inventory policy for healthcare systems. In 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-6). IEEE.
- [66] Aldhubaib, H. A., & Kashef, R. (2020). Optimizing the utilization rate for electric power generation systems: A discrete-event simulation model. *IEEE Access*, 8, 82078-82084
- [67] Lai, J., Che, L., & Kashef, R. (2021, September). Bottleneck Analysis in JFK Using Discrete Event Simulation: An Airport Queuing Model. In 2021 IEEE International Smart Cities Conference (ISC2) (pp. 1-7). IEEE
- [68] Kashef, R. (2021, April). Scattering-based Quality Measures. In 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) (pp. 1-8). IEEE.