Network Intelligence for Enhanced Multi-Access Edge Computing (MEC) in 5G

Raihan Ul Islam
and Karl Andersson
Pervasive and Mobile Computing Laboratory
Luleå University of Technology
Skellefteå, Sweden.
Email: {raihan.ul.islam, karl.andersson}@ltu.se

Mohammad Shahadat Hossain
Department of Computer Science and Engineering
University of Chittagong
Chittagong, Bangladesh.
Email: hossian_ms@cu.ac.bd

***Abstract*—5G networks will enable people and machines to communicate at high speeds and very low latencies, in a reliable way. This opens up opportunities for totally new usage patterns and the fully connected Industry 4.0-enabled enterprise covering the entire value chain from design, production, deployment, to usage of products. 5G will be rolled out across the whole world, including Sweden where the first 5G test network was launched late 2018. One important new feature in 5G is the emerging edge computing capabilities, where users can easily offload computational tasks to the network's edge very close to the user. At the same time, computational tasks traditionally performed in central nodes can be offloaded from remotely located data centres to the network's edge. Multi-access Edge Computing (MEC) is a promising network architecture delivering solutions along these lines offering a platform for applications with requirements on low latencies and high reliability. This paper targets this environment with a novel Belief-rule-based (BRB) unsupervised learning algorithm for clustering helping 5G applications to take intelligent decisions on software deployment. The scenarios consist of different combinations of numbers of users and connections and mobility patterns. The target environment is built up using a three-tier structure with a container-based solution where software components can easily be spread around the network.**

I. Introduction

Mobile and Internet of things (IoT) devices are becoming more popular day by day. Currently, 63% of the total world population have mobile subscriptions. It is anticipated that there will be one trillion IoT devices by the year 2030 [1]. Nowadays, new services like video, music, social networking, gaming, augmented reality, autonomous driving, and many others are becoming more popular. There is an exponential growth of traffic data volume in the communication network because of the unprecedented increase of mobile and IoT devices as well as the emergence of new applications and services. Consequently, the traditional centralized network architecture usually implemented in a client-server fashion is unable to cope with this new scenario, resulting in long latencies for the various services. This phenomenon may become crucial when it affects safety critical services such as e-health or autonomous vehicle driving. The fifth-generation mobile networks (5G) will enable people and machines (with embedded devices) to communicate at high speeds and very low latencies ($<$1 ms), in a reliable way. This opens up opportunities for totally new usage patterns and the fully connected Industry 4.0-enabled enterprise covering the entire value chain from design, production, deployment, to usage of products. Being a truly global standard, 5G will be rolled out across the whole world, including Sweden where the first 5G test network was launched at KTH in December 2018. One important new feature in 5G is the emerging edge computing capabilities, where users can easily offload computational tasks to the network's edge very close to the user. At the same time, computational tasks traditionally performed in central nodes can be offloaded from remotely located data centres to the network's edge. Multi-access Edge Computing (MEC), standardized by ETSI, is a promising network architecture delivering solutions along these lines offering a platform for applications with requirements on low latencies and high reliability [2][3].

A plethora of research on how to offload computation from mobile devices to MEC servers has been conducted [4][5][6][7]. They considered various network parameters, including latency, energy usage of devices, wireless channel allocation and MEC server resources to determine the offloading of computation. All of them considered computational offloading from mobile devices to MEC servers and they have not included remote cloud servers in their approaches. However, to leverage the full strength of MEC, remote cloud servers should also be considered for computational offloading. This would allow less important components of the software with high latency tolerance to be offloaded to the remote cloud server. It is necessary to decrease the load from the MEC server since it is usually not highly configured in terms of memory, storage and computational power. Hence, the different software components deployed on the MEC servers should rather be optimally distributed across the network. This would allow the allocation of the available resources in an optimal way, allowing the increase of performance of the network significantly.

The optimal distribution of software components across the MEC environment depends upon various factors including size of software, memory consumption, bandwidth, uplink and downlink latency, persistence, proximity, number of cores and many others. Some of the mentioned factors can be measured in a quantitative way, while others can only be described in a qualitative way. Also, most of the factors cannot be measured with 100% certainty, for example proximity, persistence, uplink and downlink latency due to the presence of various kinds of uncertainty such as incompleteness, ignorance, randomness, vagueness, imprecision and ambiguity [8]. Consequently, the decision on offloading of software components either to the MEC server or to the remote cloud server could become inaccurate. Eventually, the optimal performance of the MEC environment would be hampered due to the inaccurate computational offloading. None of the mentioned research [4][5][6][7] considered this uncertainty phenomenon of MEC environment and hence, the decision on distribution of the software components by using these approaches should inherently contain inaccuracy.

BRBs provide an integrated framework of addressing various types of uncertainties [9][10][11][12][13][14]. The distribution of software components at various locations, including end-user device, MEC server and remote cloud server is an example of an optimization problem. Therefore, the BRB environment should be equipped with learning or training by optimizing its various parameters, necessary to allocate the various software components at appropriate locations in the MEC environment. Usually, the values of these parameters are determined by the domain experts and hence, they are inherently inaccurate. Therefore, different learning mechanisms were proposed to acquire optimal values of the learning parameters by using training data [21][22][23][24][25][26][27]. Our recently developed BRBaDE (Belief Rule Base adaptive Differential Evolution) optimization algorithm [28] provides a way of finding the accurate optimal values of the parameters in various domains since the algorithm addresses the issues of both optimal exploitation and exploration in a balanced way. A BRB environment can also be used as a classifier [29], where the training is achieved by using labeled data and hence, the learning can be considered as supervised. However, the allocation of software components at three different locations in the MEC environment is an example of a clustering problem, which requires unsupervised learning. Therefore, our proposed solution targets this environment with a novel Belief-rule-based (BRB) unsupervised learning algorithm for clustering helping 5G applications to take intelligent decisions on software deployment.

II. Related Work

MEC is a new concept, allowing a mobile device to offload the computation of its software components at various locations of the network, resulting in increasing its reliability and performance. There exists a plethora of techniques, enabling optimally offloading the computation from mobile devices to MEC servers. In addition to the offloading decision, Wang et al. [4] considered resource allocation and content caching strategies as examples of optimization problems. They used Alternating Direction Method of Multipliers (ADMM) based optimization algorithm for allocating resources to MEC servers in a multi user multi connectivity scenario. However, they considered such key factors as inputs, which are actually difficult to obtain because of the varying wireless channels.

Yu et al. [5] developed a dynamic offloading framework using deep learning in an ANN environment by considering the limited communication and computation resources of the MEC server. They considered a single user single connectivity scenario. However, this deep learning approach, because of the presence of ANN environment, lacks the transparency and hence, works as a black-box.

Wang et al. [6] proposed Deep Reinforcement Learning (DRL) techniques for optimizing computing, caching and communication resources of MEC in a multi user multi connectivity scenario. Furthermore, an “In-Edge AI” framework has been developed for better collaboration among devices and edge nodes for improving the training and inferencing the mathematical models, which will help in better optimization of resources of MEC servers and devices, while reducing unnecessary communication load. However, there should be a balance of exploration and exploitation to achieve optimal results using DRL, which is very critical.

Asheralieva [7] proposed a joint computational offloading and content caching method in the wireless heterogeneous MEC environment, where each small cell base station (BS) is equipped with a MEC server having the content caching/processing capabilities. Asheralieva [7] considered ANN techniques in a multi user single connectivity scenario for computational offloading and content caching.

None of the above-mentioned methods considered uncertainty associated with the factors necessary to determine the optimal offloading of computation and content caching of the mobile devices in the MEC environment. Moreover, the above-mentioned methods lack the procedures to process uncertain data and hence, the decision on offloading of computation or on content caching could not be accurate. Eventually, the performance of the MEC environment will significantly be degraded.

The BRB environment not only provides an opportunity to address various uncertainty but also handles both qualitative and quantitative data in an integrated way [15][16][17][18]. BRBs have also been successfully used in location-allocation problems where multiple factors of both qualitative and quantitative nature under uncertainty are considered [19][20]. However, the accuracy of the decisions of such BRB environments depend upon obtaining optimal values of various learning parameters. This is achieved by training input-output or labeled data by using various optimization algorithms. Hence, the learning in BRB environments is supervised. Yang et al. [21] proposed sequential quadratic programming based optimization technique which suffers from local optima problem. Therefore, Yang et al. [26] proposed Differential Evolution (DE) based joint optimization for BRB. However, it lacks the procedures of being a balanced optimization algorithm because it keeps the mutation and cross-over factors constant during each iteration [30] [31]. However, our recently developed BRBaDE algorithm [28] overcomes this limitation by integrating BRB with DE, allowing the dynamic allocation of mutation and cross-over factors at each iteration. It ensured optimal exploitation and exploration. Since the offloading of computation either at mobile devices or at the MEC servers or at remote cloud servers is a clustering problem, it is not possible to obtain the labeled data. Consequently, a BRB based classifier, which uses labeled data should not be appropriate for this purpose. Moreover, this classifier uses DE as the optimization algorithm, which is not a balanced algorithm, necessary to obtain optimal learning parameter value. Therefore, the development of a novel BRB based unsupervised learning algorithm for clustering to handle non-labeled data of the types available in the MEC environment is necessary. This novel BRB based unsupervised learning algorithm should be a balanced one with the capability of optimal exploitation and exploration, so that increased accuracy of the distribution of software components in the MEC environment would be achieved. Eventually, this novel BRB based unsupervised learning algorithm will allow self-learning without the intervention of an expert in the MEC environment, which is called cognitive computation or the mind of the MEC environment can be computed. Such a novel approach will be applied in the three scenarios consisting of single user with single connectivity, single user with multi connectivity and multi user with multi connectivity and mobility.

III. Belief Rule Base Environment

Belief Rule Base is an extended version of traditional IF-Then rule with belief structure incorporated, which helps in addressing all types of uncertainty. Eq. 1 represents a Belief Rule. ER is used as inference methodology for generating result using the BRB.

$$R_k : \begin{cases} \text{IF } (A_1 \text{ is } V_1^k) \text{ AND / OR } (A_2 \text{ is } V_2^k) \text{ AND / OR} \\ \ldots \text{ AND / OR } (A_{T_k} \text{ is } V_{T_k}^k) \\ \text{THEN } (C_1, \beta_{1k}), (C_2, \beta_{2k}), \ldots, (C_N, \beta_{Nk}) \end{cases} \tag{1}$$

$$\text{where } \beta_{jk} \geq 0, \sum_{j=1}^{N} \beta_{jk} \leq 1 \text{ with rule weight } \theta_k,$$

$$\text{and attribute weights } \delta_{k1}, \delta_{k2}, \ldots \delta_{kTk}, k \in 1, \ldots, L$$

where $A_1, A_2, \ldots, A_{T_k}$ are the antecedent attributes of the k^{th} rule. $V_i^k (i = 1, \ldots, T_k, k = 1, \ldots, L)$ is the referential value of the i^{th} antecedent attribute. C_j is the j^{th} referential value of the consequent attribute. $\beta_{jk}(j = 1, ..., N, k = 1, ..., L)$ is the degree of belief for the consequent reference value C_j. If $\sum_{j=1}^{N} \beta_{jk} \leq 1$, then the k^{th} rule is considered as complete; otherwise, it is incomplete.

The inference procedure consists of four steps [11][32][33]. These are input transformation, rule activation, belief update and rule aggregation. The inputs are distributed over referential values of the consequent part of a rule, which is known as input transformation. Afterwards, matching degree and activation weight of the rules are calculated using the rule weights during the rule activation step. Henceforth, the belief update is performed to address any missing inputs. As the final step, the rule aggregation are carried out using either analytical or recursive reasoning algorithm, which produces fuzzy output. The fuzzy output can be converted to crisp value using utility function.

Attribute weights, rule weights, belief degrees, referential values of each input attribute influence the outcome, which is known as learning parameters. Therefore, a learning mechanism is needed to identify the proper values of the learning parameters. BRBaDE based joint optimization is used for conducting the learning mechanism [28].

IV. Proposed Solution

In our proposed solution, we consider efficient distribution of the software components among mobile devices, MEC server and remote cloud. G is a resource-intensive software which has V components. Each component can be executed either in the mobile device, MEC server or remote cloud. The relationship between components can be described as a weighted directed graph $G = (V, \varepsilon)$. The data dependency between the components are denoted by ε, while each edge $e_{v-1}, v(e_{v-1}, v \in \varepsilon)$ represents the data communication between two components. $e_{v-1,v}$ and $e_{v,v+1}, v$ represents the input and output data size respectively. The workload for each component $v \in V$ is denoted by θ_v. Execution time for a component in mobile device can be calculated using Eq. 2, where f_u is the number of CPU cycles of the mobile device, as shown below.

$$T_l = \frac{\theta_v}{f_u} \tag{2}$$

Eqs. 3 and 4 will be used to calculate maximum achievable rate (in bps) for uplink and downlink respectively [5]. The maximum achievable rate (in bps) for uplink is shown in Eq. 3

$$r_{ul} = n \frac{B}{N} \log_2 (1 + \frac{p_u \mid h_{ul} \mid^2}{T(g_{ul}) d^\beta N_o}) \tag{3}$$

The maximum achievable rate (in bps) for downlink is shown in Eq. 4

$$r_{dl} = n \frac{B}{N} \log_2 (1 + \frac{p_s \mid h_{dl} \mid^2}{T(g_{dl}) d^\beta N_o}) \tag{4}$$

Execution time for a component in MEC server or remote cloud server can be expressed as shown in Eqs. 5, 6 and 7.

$$T_{S_{ul}} = \frac{e_{v-1,v}}{r_{ul}} \tag{5}$$

$$T_{S_{dl}} = \frac{\theta_v}{m.f_s} + \frac{e_{v,v+1}}{r_{dl}} \tag{6}$$

$$T_S = T_{S_{ul}} + T_{S_{dl}} \tag{7}$$

Furthermore, n can be considered as communication resources used for the component out of N while p is the number of computational resources for the component out of P by the MEC server or remote cloud. Similarly, total memory and storage available in mobile device, MEC Server and remote cloud can be considered as H and M respectively, while h and m can be considered as the required memory and storage for the component respectively. Therefore, the cost of offloading a component on the mobile devices can be represented as shown in Eq. 8 and on MEC server and remote cloud can be presented using Eq. 9.

$$C_l(S) = T_l(v) + \frac{h}{H} + \frac{m}{M} \tag{8}$$

$$C_r(S) = T_s + \frac{p}{P} + \frac{n}{N} + \frac{h}{H} + \frac{m}{M} \tag{9}$$

Afterwards, a BRB framework will be constructed by taking into account the cost of software component offloading, memory requirement of the component and latency tolerance, which is shown in Fig. 1. Furthermore, an example of BRB is presented by Eq. 10.

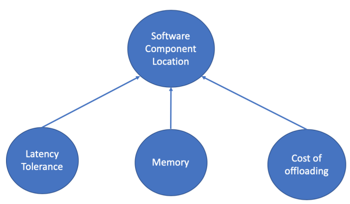

Fig. 1. BRB framework for software component offloading

$$R_k : \begin{cases} \text{IF X2 (Latency Tolerance) is Medium AND} \\ \text{X3 (Memory) is Low AND} \\ \text{X4(Cost of Offloading)is High} \\ \\ \text{THEN X1 (Software Component Location) is} \\ \{\text{(Mobile Device, 0), (MEC server, 0), (Remote Cloud, 1)}\} \end{cases} \tag{10}$$

Afterwards, the locations of software components deployment can be determined using the Chang et al. [29] proposed methodology. However, to improve the accuracy of clustering different deployment locations across the network a new optimization function (Eq. 11) needs to be used as there is a lack of labeled data.

$$\xi = \frac{1}{M} \sum_{m=1}^{M} (x_m - \mu_{c^i})^2 \tag{11}$$

Here, $x_m(m = 1 \ldots M)$ represents the input data, $\mu_{c^i}(i = 1 \ldots N)$ is the ith cluster centroid and N is the number of clusters or the numbers of software component deployment locations.

In summary, the proposed novel BRB unsupervised cluster algorithm will be used to distribute the software components optimally among mobile devices, MEC server and remote cloud based on the cost of deploying software components, latency tolerance and memory requirements. The BRB will play a significant role in addressing all types of uncertainty of different network components.

V. Evaluation Framework

Building upon the existing methods, a novel BRB based unsupervised learning algorithm will be developed, allowing optimal distribution of software components among mobile devices, MEC servers and remote cloud servers, while ensuring optimal usage of resources. In order to evaluate our solution, we are considering an evaluation framework with a Docker-based container implementation allowing for efficient distribution of micro-services in a networked system. The system will be set up using a three-tier architecture as shown in Fig. 2.

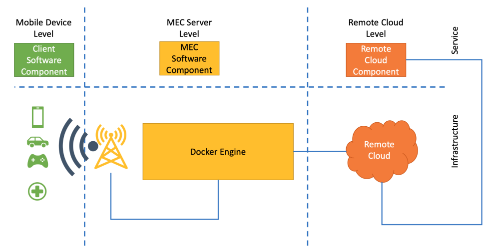

Fig. 2. Evaluation framework

As mentioned above, Docker is considered as the container implementation used at all three layers for the distribution of software components in the evaluation framework. EdgeCloudSim [34] is a new edge simulator based on CloudSim is considered as our simulation platform. It provides a mobility model, a network link model, a load generator model and an edge server model to evaluate various aspects of Edge Computing. This allows us to evaluate the proposed optimal distribution of software components using the novel BRB based unsupervised clustering algorithm in detail.

VI. Live Experiments

A 5G testbed is currently being installed at Luleå University of Technology, “5G Innovation Hub North” [35] as part of the Wireless Innovation Arena project [36]. Our work will use this testbed and perform experiments in a controlled environment.

VII. Conclusion

Edge computing is becoming more popular due to the demands of the new types of software such as autonomous driving, augmented reality based games, remote health services and many more. To cater the requirements, different components of the software need to be distributed among the network nodes. The proposed optimal software components distribution using the novel BRB based clustering algorithm will provide new opportunity to address the specific scenario.

References

[1] M. Chen, S. Mao, Y. Zhang, and V. C. Leung, “Big data: related technologies, challenges and future prospects,” 2014.

[2] ETSI, “Multi-access Edge Computing (MEC); Phase 2: Use Cases and Requirements,” ETSI GS MEC 002 V2.1.1, Tech. Rep., 2018.

[3] ——, “Multi-access Edge Computing (MEC); Framework and Reference Architecture,” ETSI GS MEC 003 V2.1.1, Tech. Rep., 2019.

[4] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 4924–4938, 2017.

[5] S. Yu, X. Wang, and R. Langar, "Computation offloading for mobile edge computing: A deep learning approach," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2017, pp. 1–6.

[6] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning," 2018.

[7] A. Asheralieva, "Optimal computational offloading and content caching in wireless heterogeneous mobile edge computing systems with hopfield neural networks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2019.

[8] R. U. Islam, M. S. Hossain, and K. Andersson, "A novel anomaly detection algorithm for sensor data under uncertainty," *Soft Computing*, vol. 22, no. 5, pp. 1623–1639, 2018.

[9] R. Ul Islam, K. Andersson, and M. S. Hossain, "A web based belief rule based expert system to predict flood," in *Proceedings of the 17th International conference on information integration and web-based applications & services*. ACM, 2015, pp. 1–8.

[10] M. R. S. Hridoy, M. S. Hossain, R. U. Islam, and K. Andersson, "A web based belief rule based expert system for assessing flood risk," in *Proc. iiWAS*, 2017, pp. 434–440.

[11] M. S. Hossain, K. Andersson, and S. Naznin, "A belief rule based expert system to diagnose measles under uncertainty," in *World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLDCOMP'15): The 2015 International Conference on Health Informatics and Medical Systems 27/07/2015-30/07/2015*, 2015, pp. 17–23.

[12] R. Karim, K. Andersson, M. S. Hossain, M. J. Uddin, and M. P. Meah, "A belief rule based expert system to assess clinical bronchopneumonia suspicion," in *Proc. FTC*, 2016, pp. 655–660.

[13] L. Dymova, P. Sevastjanov, and K. Kaczmarek, "A forex trading expert system based on a new approach to the rule-base evidential reasoning," *Expert Systems with Applications*, vol. 51, pp. 1–13, 2016.

[14] K. Andersson and M. S. Hossain, "Smart risk assessment systems using belief-rule-based dss and wsn technologies," in *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*. IEEE, 2014, pp. 1–5.

[15] J.-B. Y. J. Liu, J. Wang, H.-S. Sii, and H.-W. Wang, "Belief rule-base inference methodology using the evidential reasoning approach-rimer," *IEEE Transactions on systems, Man, and Cybernetics-part A: Systems and Humans*, vol. 36, no. 2, pp. 266–285, 2006.

[16] M. S. Hossain, S. Rahaman, R. Mustafa, and K. Andersson, "A belief rule-based expert system to assess suspicion of acute coronary syndrome (acs) under uncertainty," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 22, no. 22, pp. 7571–7586, 2018.

[17] M. S. Hossain, S. Rahaman, A. L. Kor, K. Andersson, and C. Pattinson, "A belief rule based expert system for datacenter pue prediction under uncertainty," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 140–153, 2017.

[18] M. S. Hossain, F. Ahmed, F.-T. Johora, and K. Andersson, "A belief rule based expert system to assess tuberculosis under uncertainty," *Journal of Medical Systems*, vol. 41, no. 3, pp. 1–11, 2017.

[19] M. M. Islam, T. Mahmud, and M. S. Hossain, "Belief-rule-based intelligent decision system to select hospital location," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 1, no. 3, pp. 607–618, 2016.

[20] T. Mahmud and M. S. Hossain, "An evidential reasoning-based decision support system to support house hunting," *International Journal of Computer Applications*, vol. 57, no. 21, pp. 51–58, 2012.

[21] J.-B. Yang, J. Liu, D.-L. Xu, J. Wang, and H. Wang, "Optimization models for training belief-rule-based systems," *IEEE Transactions on systems, Man, and Cybernetics-part A: Systems and Humans*, vol. 37, no. 4, pp. 569–585, 2007.

[22] L. Chang, J. Sun, J. Jiang, , and M. Li, "Parameter learning for the belief rule base system in the residual life probability prediction of metalized film capacitor," *Knowledge-Based Systems*, vol. 73, pp. 69–80, 2015.

[23] L. Chang, Y. Zhou, J. Jiang, M. Li, and X. Zhang, "Structure learning for belief rule base expert system: A comparative study," *Knowledge-Based Systems*, vol. 39, pp. 159–172, 2013.

[24] Y.-M. Wang, L.-H. Yang, Y.-G. Fu, L.-L. Chang, and K.-S. Chin, "Dynamic rule adjustment approach for optimizing belief rule-base expert system," *Knowledge-Based Systems*, vol. 96, no. C, pp. 40–60, 2016.

[25] Z.-J. Zhou, C.-H. Hu, J.-B. Yang, D.-L. Xu, M.-Y. Chen, and D.-H. Zhou, "A sequential learning algorithm for online constructing belief-rule-based systems, expert systems with applications," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1790–1799, 2010.

[26] L.-H. Yang, Y.-M. Wang, J. Liu, and L. Mart?nez, "A joint optimization method on parameter and structure for belief-rule based systems," *Knowledge-Based Systems*, vol. 142, pp. 220–240, 2018.

[27] L.-L. Chang, Z.-J. Zhou, Y.-W. Chen, T.-J. Liao, Y. Hu, and L.-H. Yang, "Belief rule base structure and parameter joint optimization under disjunctive assumption for nonlinear complex system modeling," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1542–1554, 2018.

[28] R. U. Islam, M. S. Hossain, and K. Andersson, "Belief rule based adaptive differential evolution algorithm under uncertainty," *Submitted for review*.

[29] L. Chang, Z. Zhou, Y. You, L. Yang, and Z. Zhou, "Belief rule based expert system for classification problems with new rule activation and weight calculation procedures," *Information Sciences*, vol. 336, pp. 75–91, 2016.

[30] R. Storn and K. Price, "Differential evolution a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341?–359, 1997.

[31] R. D. Al-Dabbagh, F. Neri, N. Idris, and M. S. Baba, "Algorithmic design issues in adaptive differential evolution schemes: Review and taxonomy," *Swarm and Evolutionary Computation*, vol. 43, pp. 284–311, 2018.

[32] T. Mahmud, K. N. Rahman, and M. S. Hossain, "Evaluation of job offers using the evidential reasoning approach," *Global Journal of Computer Science and Technology*, 2013.

[33] M. S. Hossain, S. Rahaman, R. Mustafa, and K. Andersson, "A belief rule-based expert system to assess suspicion of acute coronary syndrome (acs) under uncertainty," *Soft Computing*, vol. 22, no. 22, pp. 7571–7586, 2018.

[34] C. Sonmez, A. Ozgovde, and C. Ersoy, "Edgecloudsim: An environment for performance evaluation of edge computing systems," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 11, p. e3493, 2018.

[35] LTU, "5G Innovation Hub North," https://www.5ginnovationhubnorth.se/, 2019 (accessed April 22, 2019).

[36] ——, "Wireless Innovation Arena," https://www.wirelessinnovationarena.se/, 2019 (accessed April 22, 2019).