Heuristic and Bent Key Exchange Secured Energy Efficient Data Transaction for Traffic Offloading in Mobile Cloud

Nithya Rekha Sivakumar[1, *], Sara Ghorashi[1], Mona Jamjoom[1] and Mai Alduailij[1]

Abstract: In today's world, smart phones offer various applications namely face detection, augmented-reality, image and video processing, video gaming and speech recognition. With the increasing demand for computing resources, these applications become more complicated. Cloud Computing (CC) environment provides access to unlimited resource pool with several features, including on demand self-service, elasticity, wide network access, resource pooling, low cost, and ease of use. Mobile Cloud Computing (MCC) aimed at overcoming drawbacks of smart phone devices. The task remains in combining CC technology to the mobile devices with improved battery life and therefore resulting in significant performance. For remote execution, recent studies suggested downloading all or part of mobile application from mobile device. On the other hand, in offloading process, mobile device energy consumption, Central Processing Unit (CPU) utilization, execution time, remaining battery life and amount of data transmission in network were related to one or more constraints by frameworks designed. To address the issues, a Heuristic and Bent Key Exchange (H-BKE) method can be considered by both ways to optimize energy consumption as well as to improve security during offloading. First, an energy efficient offloading model is designed using Reactive Heuristic Offloading algorithm where, the secondary users are allocated with the unused primary users' spectrum. Next, a novel AES algorithm is designed that uses a Bent function and Rijndael variant with the advantage of large block size is hard to interpret and hence is said to ensure security while accessing primary users' unused spectrum by the secondary user. Simulations are conducted for efficient offloading in mobile cloud and performance valuations are carried on the way to demonstrate that our projected technique is successful in terms of time consumption, energy consumption along with the security aspects covered during offloading in MCC.

Keywords: Cloud computing, mobile cloud computing, heuristic, bent key exchange, reactive offloading.

[1] Computer Science Department, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia.

[*] Corresponding Author: Nithya Rekha Sivakumar. Email: NRRaveendiran@pnu.edu.sa.

Received: 12 May 2020; Accepted: 31 July 2020.

CMC. doi:10.32604/cmc.2020.011505 www.techscience.com/journal/cmc

1 Introduction

In Zaharia et al. [Zaharia, Ciobanu and Dobre (2019)], Lyapunov-based Dynamic Offloading Decision (LDOD) was designed for offloading process. This method executed flexible subtasks by the local terminals as per the offloading decision. Moreover, with the objective of enhancing the task offloading performance, an allocation scheme for computation resource was also designed that in turn allocated the computation resource of mobile computing server to vehicles with minimum energy consumption. Finally, a Lyapunov drift plus penalty minimization problem was also designed to reduce queue stability with the average length of transmission queue.

A Context-Sensitive Offloading System (CSOS) was introduced [Gnana and Maluk (2018)] to offer offloading decisions with high accuracy. First a decision engine was developed for decision-making. A profiling system was presented which transformed raw context elements to high-level context information at runtime, hence contributing to lesser runtime and energy consumption.

The key resources inspired by mobile applications are the energy and time consumption during offloading. In order to prevent attack against the offloaded data from potential attacks, security mechanisms need to be implemented. Hence, this work the main emphasis remains in constructing a new method that integrates the above-mentioned constraints, resulting in improved performance of mobile applications and their protection against attacks.

Inspired by the idea that both security and energy consumption aspects are considered during offloading in MCC environment, in this article, the joint energy and time consumption, and security aspect are considered during offloading as a constraint condition. The contribution of article is summarized as below:

(1) In this paper, the computation offloading for spectrum analysis in MCC is scrutinized. In cloud, the energy and time consumption are taken as the optimization objectives.

(2) Depends on theoretical analysis, a multi-objective optimization model lessen the memory usage when meeting time limit constraints and energy consumption during offloading in MCC given by workflow are established.

(3) We propose a model named Reactive Heuristic Search Optimization (RHSO) for MCC environment based on reaction heuristic optimization algorithm for solution. Some metrics in algorithm step are enhanced to suit necessitates of this issue.

(4) We designed an algorithm named, novel AES algorithm to ensure security while offloading primary user's unused spectrum to secondary user.

(5) Experiments and simulations have evident that H-BKE method is effective than the other methods and offer optimization offloading strategy with lesser energy and time consumption.

The rest of article is ordered as follows. Section 2 reviews the related work. In Section 3, the Design Methodology is described in detail. In section 4, the performance and simulation result analysis is discussed. The conclusion is presented in Section 5.

2 Related works

In computer network, cloud computing and Mobile Devices (MDs) have become an essential part of people's daily lives. In order to utilize the full benefit of CC, MCC brings new services and facilitations to Mobile Users (MUs). Despite the provisioning of full advantage, with the location of remote cloud far away from the MUs, during data transmission results in high network latency, compromising Quality of user Service (QoS).

A comprehensive investigation on energy and time consumption, and cost for cloudlet resource were analyzed [Peng, Zhu, Zhang et al. (2019)]. A multi-objective optimization model was introduced with offloading model depends on multi-objective on non-dominated sorting genetic algorithm to identify the optimal offloading strategy. In Akherfi et al. [Akherfi, Gerndt and Harroud (2016)], an energy efficient delay aware offloading scheme was designed to discover the offloading with lesser cost graph algorithm.

A genetic algorithm was introduced [Lan, Zhang, Liu et al. (2018)] with the objective of resolving the issue related to execution time constraints. To stimulate the opportunistic usage of unexploited Internet connections, a new and open market was designed [Nur Idawati and Maolin (2016)] that leased the bandwidth made available by third parties via access points to improve network capacity in a dynamic manner. In Ranji et al. [Ranji, Mansoor and Sani (2020)], an energy-aware task offloading mechanism is designed for the mobile devices to improve the energy efficiency.

An adaptive framework was introduced [Rajeswari and Ravi (2018)] with offloading capabilities in MEC. A complete offloading solution was designed [Al-Shayeji and Ebrahim (2019)] both saving the battery and gaining the data rate using machine learning. Yet another-aware multi-resource task scheduling algorithm was designed [Hoteit, Secci, Pujolle et al. (2015)] to attain optimal offloading.

An in-depth investigation of current offloading methods, computation offloading algorithms was analyzed and their critical issues were discussed [Rajeswari and Ravi (2018)]. In addition, several critical metrics based on which the methods were implemented were also discussed, but the security aspects were not covered. In Paris et al. [Paris, Martignon, Filippini et al. (2015)], an XOR mechanism was designed that enhanced both security and energy depending on the rate of sensing.

A multi-site offloading problem was addressed [Rajeswari and Ravi (2018)] by introducing heuristic mechanism. The drawback of the mobile and cloud environment results in several limitations, like inadequate bandwidth, delay in time and energy, frequent disconnected network and so on. A strategy dynamic task optimization model was presented [Lin, Peng, Bian et al. (2019)] with offloading algorithm. Here, a coordinated management and optimized resource utilization model for offloading process was designed.

A novel communication model for Mobile Ad hoc Network (MANET) using heterogeneous secured reflection inducement e-state algorithm was presented [Huang, Xu, Lai et al. (2020)] using a similarity measure algorithm, therefore contributing to high-offloading ratio and low delay. Yet another pass point hotspots was analyzed [Duan, Akhtar and Wang (2015)] to evaluate capacity and energy saving based on signal quality information. However, end user quality of service was not ensured. To address this issue, a load balancing algorithm was designed [Chen, Chen, Ma et al. (2019)] to satisfy end user's quality of service.

The delay in performing tasks is considered to be one of the main limitations of cloud that is scarce in resources upon comparison with cloud server. Due to this, the tasks that are to be performed are split and are provided to respective mobile devices, different cloud servers and cloudlet. Therefore, to evaluate the integration of devices required to perform several tasks, deep learning algorithms were applied [Wang, Zhu, Hei et al. (2019)]. A time efficient offloading method was designed [Yunsik and Yangsun (2017)] for intelligent sensors to optimize both energy consumption and privacy entropy. Local computational resources were utilized [Xu, Liu, Jiang et al. (2019)] to optimize offloading.

Therefore, based on the detailed analysis of the research performed by the researchers in the area of offloading and user requests in the mobile cloud it is concluded that it's the need of the hour to propose an algorithm that will ensure both optimal offloading by reducing the energy, time consumption and provide security.

3 Methodology

Offloading indicates a procedure which migrate resource demanding from Mobile User 'MU' to Mobile Cloud 'MC'. The aim of Mobile Cloud-based computation offloading enhances the performance of concern application, reduces power consumption [Sivaram, Kaliappan, Shobana et al. (2020)] and execution time that failed to execute by MU owing to inadequate storage resources in mobile host or MU. It refers to the process where the intricate application is said to be executed in cloud and arrived results are communicated back to mobile host or MU.

This research whole work H-BKE method concentrates on secured energy efficient data transaction in mobile cloud for traffic overloading, i.e., reducing energy consumption, time and improving security. The block diagram of H-BKE method is depicted in Fig. 1.

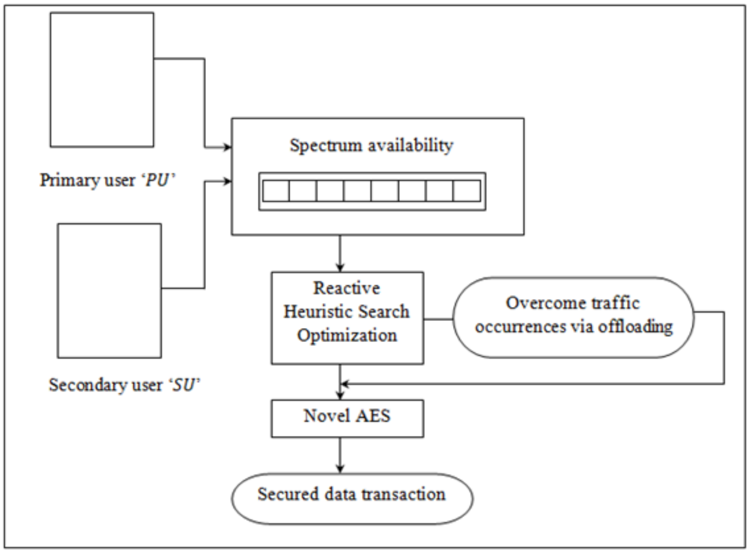


Figure 1: Block diagram of the H-BKE method

As shown in the above Fig. 1 when a greater number of requests are transmitted from mobile devices, it will result in traffic and hence degrading the network performance. In our work, traffic overloading or congestion is said to be minimized by using the concept of offloading. Here, primary user, secondary user and spectrum usage are analyzed. According to the users and spectrum availability the data request is scheduled and occurrences of traffic are avoided. Initially primary user is served with high priority. Hence, spectrum issues are not found and therefore, primary users request are processed in a smooth manner without traffic.

Next, secondary users request with minimum spectrum availability will leads to traffic and therefore causing delay in data delivery to the respective users. In our work, primary user's spectrum that is available as unused are allocated to secondary users via offloading. In this manner, traffic occurrence has been overcome, therefore reducing time, energy and memory usage, therefore resulting in better performance. Also, while using the primary user's spectrum, data transaction should take place in a secure manner. Therefore, AES encryption has been implemented in our work.

3.1 System model and problem formulation

The system model and problem formulation are presented in this section. At first, the MCC architecture is described. After that, the basic model is presented. The time consumption mode, energy consumption mode and cost mode are explained. Fig. 2 given below shows the MCC architecture.

Cloud includes of union of data centers. Mobile Device 'MD' in the figure indicates a mobile phone or tablet. Each 'MD' have one or more applications that are assumed to be time constrained and need to be processed. In this work for simple usage 'MD' is referred to as the 'MU'. These applications are executed in a direct manner or MU migrate a part of application or full application to CC environment via Local Area Network (LAN) to lessen user execution time, energy consumption and so on.

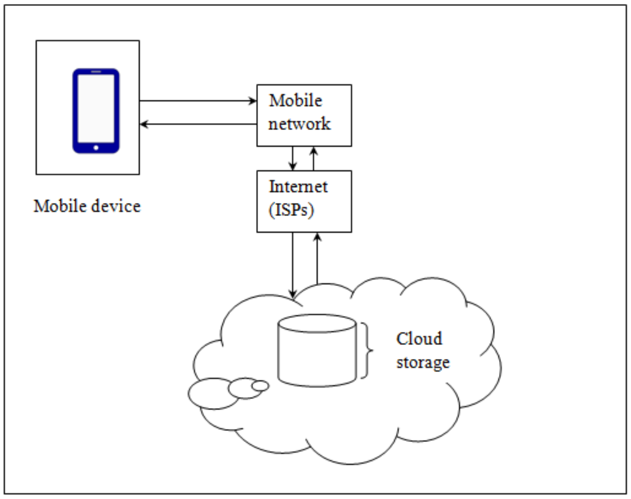


Figure 2: MCC architecture

As shown in the Fig. 2. The workflow application ‘WA’ is designed by a Direct Acyclic Graph ‘$G_i(V,E)$’ where ‘i’ represents the ‘ith’ ‘$WA(1 \leq f \leq F)$’ and ‘F’ denotes the total number of workflow application ‘WA’. Each application includes of multiple requests and any device or user in above Fig. 2 is considered as a request. Here, ‘$R = \{r_1 i, r_2 i, \dots, r_n i\}$’ represents the set of requests, and edge in ‘E’ denotes the set of dependency between any two requests. Each edge is in turn correlated with a weight ‘$w_{a,b}$’ that represents the size of data traffic between transmitted between request ‘r_{af}’ and request ‘r_{bf}’ respectively. The mobile cloud is configured as multiple virtual machines ‘VMs’ for providing parallel processing the workflow applications.

Each request ‘$r_{i,f}$’ in ‘R’ is modelled as a 2-tuple ‘$r_{i,f} = \left(AI_{i,f}, OS_{i,f}\right)$’ where ‘$AI_{i,f}$’ represents the average number of instructions of request ‘$r_{i,f}$’ and ‘$OS_{i,f}$’ represents the offloading strategy for request ‘$r_{i,f}$’ that in our work is denoted as a single dimensional vector ‘$OS = \left\{OS_{i,f} \mid i = 1,2,\dots,n, f = 1,2,\dots,F\right\}$’, where ‘$n$’ represents the number of requests in the ‘$fthWA$’ and ‘$OS_{i,f} = 0$’ represents the request ‘$r_{i,f}$’ is processed locally, ‘$OS_{i,f} = 0$’ represents the request ‘$r_{i,f}$’ is offloaded to the cloud.

3.2 Problem formulation

The aim of this article is to optimize energy and time consumption with minimum memory usage while meeting time limit constraint during offloading in MCC given by the workflow application. This is formulated as given below.

$$MinT_{wa,i}(OS), \forall f \in \{1,2,\dots,F\} \tag{1}$$

$$MinE_{wa,i}(OS), \forall f \in \{1,2,\dots,F\} \tag{2}$$

$$MinMU_{wa,i}(OS), \forall f \in \{1,2,\dots,F\} \tag{3}$$

$$SuchthatT_{wa,f}(OS) \leq TL(f), \forall f \in \{1,2,\dots,F\} \tag{4}$$

From the above Eqs. (1)-(4). ‘T’ Represents the time consumption, ‘E’ represents the energy consumption, ‘MU’ represents the memory usage and ‘TL’ refers to the time line constraint of the ‘$fthWA$’ given in advance or earlier.

3.3 Reactive heuristic search optimization

RHSO model support the request to offer solution by self-adapting a local search method based on previous history of search. Here, reactive refers to the ready response to events (i.e., request of data transaction by the primary and secondary users) during the search for the self-tuning of certain critical parameters (i.e., allocation of available spectrum). Fig. 3 shows the block diagram of RHSO model.

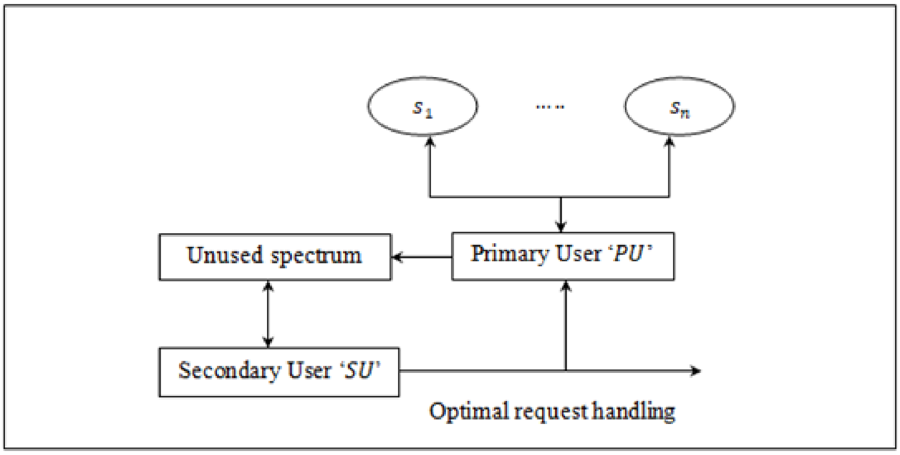


Figure 3: Block diagram of reactive heuristic search optimization model

As illustrated in the above Fig. 3 in our work, optimization between primary and secondary user according to spectrum availability for data transaction in MCC environment with high priority assumed for primary user and low priority assumed for secondary user for traffic offloading is presented.

Here, a local search is made for the available spectrum for data transaction in MCC environment ready to be used by the primary and secondary user and depending on the previous history of search spectrum allocation is made for the corresponding users. The RHSO model learns the spectrum availability '$S_{avail} = (s_1, s_2, \ldots, s_n)$' with '$\sum_{i=1}^{m} s_i = 1$', by optimizing the linear combination (i.e., allocating the available spectrum between the primary and secondary user via offloading) 'LC' of the objectives.

$$C(PU, s) = s_1(PU) + s_2(PU) + s_3(PU) + \cdots + s_m(PU) \tag{5}$$

From the above Eq. (5). '$LC(PU, s)$' refers to the linear combination of the primary user 'PU' with respect to spectrum 's'. With the above equation in a dense form is mathematically written as given below.

$$LC(PU_1, PU_2, PU_3, \ldots PU_n, s) = f(PU)^T . s \tag{6}$$

From the above Eq. (6). '$f(PU)^T$', represent the transpose function of a set of primary user in MCC environment with the probability of available overall spectrum denoted by 's'. Without generality loss, let us consider that 'LC' is optimized so that free spectrum are allocated to secondary user in case of traffic. The unused spectrum 'S_{uused}' of primary user 'PU' are offloaded to the secondary user 'SU'. Let us assume the primary user as '$PU = (PU_1, PU_2, PU_3, \ldots, PU_n)$' and the secondary user as '$SU = (SU_1, SU_2, SU_3, \ldots, SU_n)$' be the two types of users in MCC environment. Then, a 3-tuple offloading MU requests is mathematically expressed as given below.

$$R_i = \left(r_i, w_i, T_i^{arrival}\right) \tag{7}$$

From the above Eq. (7). The 3-tuple 'R_i' is measured based on the data size of request 'r_i' to be offloaded, 'w_i' denoted by CPU cycle needed to complete the request 'i' and

'$T_i^{arrival}$' denoted by the arrival time of request 'i' at the scheduler. Let '$r_{t,m}^{ST}$' and '$r_{t,m}^{FT}$' be the starting time and finishing time of user request 'i' executed by a secondary user 'SU' respectively. Then, the binary variable '$BV_{i,m}$' represents whether request 'i' is scheduled for secondary user or not-scheduled.

$$BV_{i,m} = \begin{cases} 1, ifuserrequestiisassignedtomachine \\ 0, Otherwise \end{cases} \tag{8}$$

Upon unsuccessful scheduling, the process is said to be optimized via offloading and represented as given below.

$$BV_{i,m} = \begin{cases} 1, ifuserrequestiisoffloaded \\ 0, Otherwise \end{cases} \tag{9}$$

From the above Eq. (9). The binary variable that denotes the used wireless medium for user request 'i' is set of '1'. On the other hand, it is set to 0. The pseudo code representation of Reactive Heuristic Offloading is given below.

Algorithm 1 Reactive heuristic offloading

Input: Primary User 'PU=(PU1, PU2, PU3, …, PUn)', Secondary User 'SU=(SU1, SU2, SU3, …, SUn)

Output: Optimized offloading

```
Initialize Spectrum availability 'Savail', Spectrum used by primary user 'Sused'
    Begin
        For each Primary User 'PU' and Secondary User 'US'
            Measure 'Suused=savail-sused'
            Evaluate optimized the linear combination using (5)
            Evaluate optimized linear combination in dense from using (6)
            Obtain 3- tuple offloading using (7)
            Obtain binary variable for scheduling using (8)
            Obtain offloading mechanism using (9)
            Return (optimal offloading)
        End for
    End
```

As given in the above Reactive Heuristic Offloading algorithm, three steps are performed. First, the number of primary users, secondary users and spectrums are initialized. Second, with high priority settings for primary users, the available spectrum is allocated to the primary user for performing data transaction in MCC according to the requests made. At last, the secondary users are assigned with unused spectrum of primary users to optimize energy consumption and execution time via reactive heuristic function.

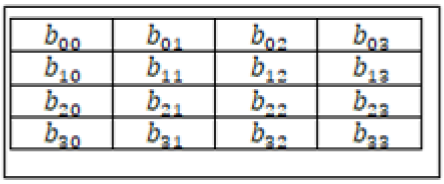
3.4 Novel AES secure data transaction

In addition to the offloading discussed above with the objective of safeguarding the offloaded data from potential attacks, a security model has to be incorporation in this work. Hence, this work concentrates on combining the above-mentioned constraints, leading towards improved performance of MCC and their protection against attacks. In other words, to provide secure data access from cloud [Gu, Wu, Yin et al. (2020)], through mobile devices AES encryption has been implemented. In this work, a novel AES algorithm is presented. The algorithm is called as novel as it incorporates two different functions in the conventional AES algorithm.

The conventional AES algorithm consists of four steps. They are, Substitution Bytes, Shift Rows, Mix Columns and Add Round Key. In this four step model, a Bent function is introduced in the Substitution Bytes step that are hard to approximate and Rijndael Variant Shift Row is used as it possesses the advantage of larger block size and hence hard to interpret. The key utilized in this algorithm includes of '128,192 or 256' bits. These bytes are rewritten as elements of definite field using below polynomial representation.

$$f(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdots + b_1 x^1 = \sum_{i=1}^{n-1} b_i x^i \tag{10}$$

Eq. (10). is represented in the form of state matrix 'S' of size '$4 * 4$' denoted by input variable 'b_{ij}' with 'i'and 'j' representing rows and the columns as given below.



b_{00}	b_{01}	b_{02}	b_{03}
b_{10}	b_{11}	b_{12}	b_{13}
b_{20}	b_{21}	b_{22}	b_{23}
b_{30}	b_{31}	b_{32}	b_{33}

Figure 4: State matrix representations

We have used key size of '128' bit concept for experimentation and hence number of rounds 'N_{round}' used in '10'. Via the several round transformations, the input state matrix is processed. State matrix as shown in Fig. 4 above progresses as it proceeds through different steps of cipher to obtain cipher text. In AES, every round follows the below steps.

The Substitute Bytes utilizes a Substitute and Shrink function where every byte of state matrixes attained from above Fig. 4 is replaced by its multiplicative inverse, followed by a Bent Function Mapping as below:

$$b_i^{'} = \frac{e^{b_i \oplus b_{(i+4)mod8} \oplus b_{(i+5)mod8} \oplus b_{(i+6)mod8} \oplus b_{(i+7)mod8} \oplus c_i}}{m} \tag{11}$$

From the above Eq. (11). The conventional Substitute Byte function given in the above numerator is included with a prime number 'm' along with an exponential value 'e' to produce finite results and hard to approximate. 'b_i' and 'c_i' represents the 'ith' bit of byte 'b' and 'ith' bit of byte 'c' respectively. The Bent Substitute Byte representation is given below.

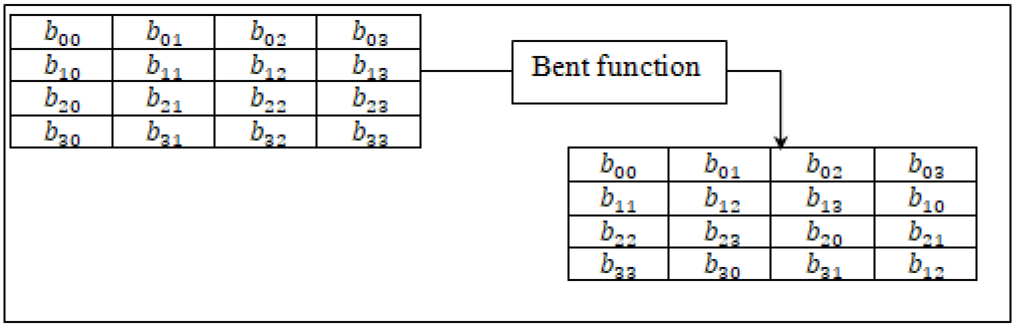

b_{00} b_{01} b_{02} b_{03}
b_{10} b_{11} b_{12} b_{13}
b_{20} b_{21} b_{22} b_{23}
b_{30} b_{31} b_{32} b_{33}
Bent function
b_{00} b_{01} b_{02} b_{03}
b_{11} b_{12} b_{13} b_{10}
b_{22} b_{23} b_{20} b_{21}
b_{33} b_{30} b_{31} b_{12}

Figure 5: Bent substitute byte representations

Followed by Bent Substitute Bytes representation, the next step is the Shift Row that operates on the rows of the state. In this work, Rijndael Variant Shift Row is used as it possesses the advantage of larger block size and hence hard to interpret. As we have used 128 bits, the first row remains unchanged, followed by the second row, shifted left circular by '$n-1\ bytes$' and so on. Each column of output state includes bytes from each column of input state.

$$S'_{r,c} = S_{(r,(c+shift(r+N_c))modN_c)} \tag{12}$$

From Eq. (12). The shift rows operation depends on number of columns 'N_c' in state matrix. In state matrix, each cell is denoted as 'S' with index of row 'r' and column 'c'. Fig. 6 given below shows the representation of Rijndael Variant Shift Row.

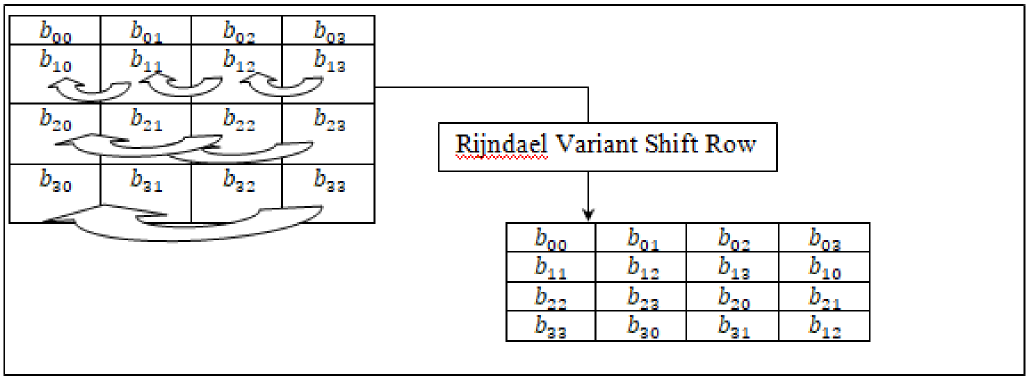
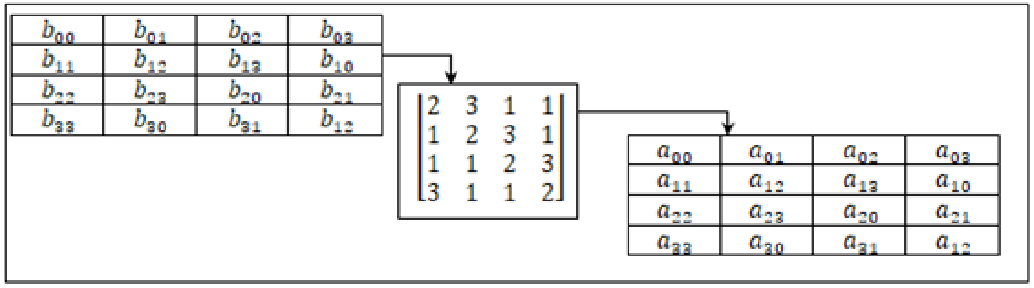

b_{00} b_{01} b_{02} b_{03}
b_{10} b_{11} b_{12} b_{13}
b_{20} b_{21} b_{22} b_{23}
b_{30} b_{31} b_{32} b_{33}
Rijndael Variant Shift Row
b_{00} b_{01} b_{02} b_{03}
b_{11} b_{12} b_{13} b_{10}
b_{22} b_{23} b_{20} b_{21}
b_{33} b_{30} b_{31} b_{12}

Figure 6: Rijndael variant shift row representation

This Mix Columns transformation functions on state matrix column-by-column. Each column is assumed as four-term polynomials over GF (i.e., Galois Field) and multiplied modulo 'x^4' with a fixed input polynomial 'b' represented as below.

$$a(x) = b_3x^3 + b_2x^2 + b_1x^1 + b_0 \tag{13}$$

Fig. 7 shows the Mix Columns representation with the resultant column vector being 'a'.



b_{00}	b_{01}	b_{02}	b_{03}
b_{11}	b_{12}	b_{13}	b_{10}
b_{22}	b_{23}	b_{20}	b_{21}
b_{33}	b_{30}	b_{31}	b_{12}

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

a_{00}	a_{01}	a_{02}	a_{03}
a_{11}	a_{12}	a_{13}	a_{10}
a_{22}	a_{23}	a_{20}	a_{21}
a_{33}	a_{30}	a_{31}	a_{12}

Figure 7: Mix column representations

Through the bitwise XOR operation, a round key 'RK' is added to state 'S'. Every round key is having size of 'N_c' columns in state matrix. 'N_c' columns in state matrix are added to rows of state matrix and written as given below.

$$\left[S'_{0,1}, S'_{1,0}, S'_{2,0}, S'_{3,0}\right] = \left[S_{0,1}, S_{1,0}, S_{2,0}, S_{3,0}\right] \oplus [N_c + N_r] \tag{14}$$

The pseudo code representation of Novel AES Secure Data Transaction is given below.

Algorithm 2 Novel AES secure data transaction

Input: input variable 'bij'

Output: robust secured key 'K'

```
Initialize state matrix 'S' of size '4×4' for input variable 'bij'
    Begin
        For each column
            Obtain polynomial expression using (10)
            Obtain bent function for substitution using (11)
            Evaluate shift row via Rjindeal variant using (12)
            Evaluate 'x4' with fixed input polynomial 'b' to obtain mix
            column using (13)
            Combine sub key with the state using (14)
            Return (robust secured key 'K')
        End for
    End
```

As given in the above Novel AES Secure Data Transaction algorithm, for each column or the secondary user who uses the primary user's spectrum, data transaction should take place in a secured manner. To ensure security, the secondary user possessing the key are said to use the primary users' spectrum, contributing to secured data transaction. With this objective, the conventional AES algorithm is modified and called as novel as it uses Bent function in the substitution byte step, therefore obtaining finite results and hard to approximate. Besides, with the application of Rijndael variant in shift row step, as the block size obtained are said to be larger, it is harder to interpret, therefore contributing the security.

4 Experimental evaluation

To evaluate the H-BKE method, a comprehensive simulation and experiment are conducted in this section. The comparison of H-BKE method is made with existing LDOD [Zaharia, Ciobanu and Dobre (2019)] and CSOS [Gnana and Maluk (2018)] with metrics energy consumption, time consumption and PDR. Simulation parameters are provided in Tab. 1.

Table 1: Simulation parameters

Parameters	Values
Mobile user	50,100,150,200,250,300,350,400,450,500
Channel bandwidth	1 MHz
Packet length	500 b
Transmission power	100 mw
Local execution power	300 mw
CPU frequency	1-3 GHz

The performance evaluation of energy efficient and secured data traffic offloading method in mobile cloud is measured based on the following metrics.

- Energy consumption
- Time consumption
- Packet drop rate

4.1 Evaluation results of energy consumption

Initially, the energy consumption is measured. First foremost parameter analysis to be made during offloading is energy consumption. Though offloading is said to be performed by several literature works, a significant amount of energy is said to be consumed during offloading. The energy consumption is measured as given below.

$$E = \sum_{i=1}^{n} MU_i * EMU_i(T) \tag{15}$$

From the above Eq. (15). The energy consumption 'E', is measured based on the frequency of 'MU_i' and the energy consumption for the 'MU' at time 'T'. The energy consumption is measured in terms of joules (J). The average energy consumption is shown in Tab. 2 with number of MUs in the range from 50 to 500 with three different methods. The average energy consumed by 50 MUs is 3.5J with existing LDOD [Zaharia, Ciobanu and Dobre (2019)] and 4.5J with the existing CSOS [Gnana and Maluk (2018)] and 2.5J using H-BKE. Comparing with the existing methods [Zaharia, Ciobanu and Dobre (2019); Gnana and Maluk (2018)], the proposed H-BKE method achieves the purpose of energy saving through task offloading.

Table 2: Energy consumption

Mobile Users	Energy Consumption (J)		
	H-BKE	LDOD	CSOS
50	2.5	3.5	4.5
100	3	4	6.5
150	3.5	5.5	8
200	5	7	9.5
250	5.5	8.5	10
300	6	9	11.5
350	6.5	10.5	12
400	8	11	13.5
450	8.8	12.5	15
500	9	14	15.5

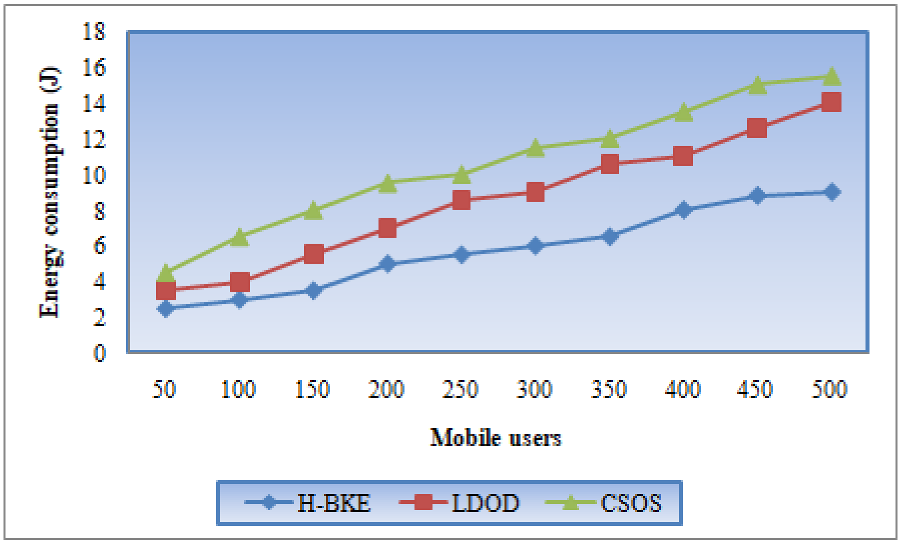


Figure 8: Performance measure of energy consumption

When considering 50 MUs, the energy consumption of three methods are 2.5J, 3.5J and 4.5J. With the increased number of MUs, the energy consumption of MU enhances to 9J, 14J, and 15.5J. This is due to, the several MUs are access the same spectrum to execute the task offloading in a simultaneous manner, which results in higher interference. Hence, with 500 MUs, more secondary users are select spectrum in a greedy manner and average energy consumption of MU enhances. In contrast, the H-BKE method can save at least 32% and 46% of energy consumption as compared to Zaharia et al. [Zaharia, Ciobanu and Dobre (2019); Gnana and Maluk (2018)]. This is due to the fact that the reactive search optimization model based offloading mechanism presents task offloading decision in a global long-term perspective, reasonably allocating unused primary users' spectrum to meet the QoS requirements. Hence, it provides minimal energy consumption when number of MUs is small. Though, with the increase in number of MUs the performance degrades due to enhance in the traffic patterns. Evidently, the H-BKE method provides better energy saving solution than existing [Zaharia, Ciobanu and Dobre (2019); Gnana and Maluk (2018)].

4.2 Evaluation results of time consumption

The next parameter that has to be analyzed is the time consumption. The time consumption here refers to the time consumed during offloading between the MU and the cloud. Lower the time consumption more the number of secondary users is being allocated with the respective spectrum and hence ensuring optimal offloading. The time consumption is measured as given below.

$$T = \sum_{i=1}^{n} MU_i * TMU_i(T) \quad (16)$$

From the above Eq. (16). The time consumption 'T', is measured based on the frequency of mobile users 'MU_i' and the time consumed for the 'MU' at time 'T' for the corresponding offloading. The time consumption is measured in terms of milliseconds (ms). The average time consumption of H-BKE method and existing [Zaharia, Ciobanu and Dobre (2019); Gnana and Maluk (2018)] is compared.

Table 3: Time consumption

Mobile Users	Time Consumption (ms)		
	H-BKE	LDOD	CSOS
50	1.75	2.5	3
100	3.25	5.5	7
150	5.5	8	9.5
200	5.9	9.5	12
250	6.3	11	13.5
300	6.5	11.5	15
350	8	12	17.6
400	9.5	13	18
450	12	14.5	21
500	12.5	15	23.5

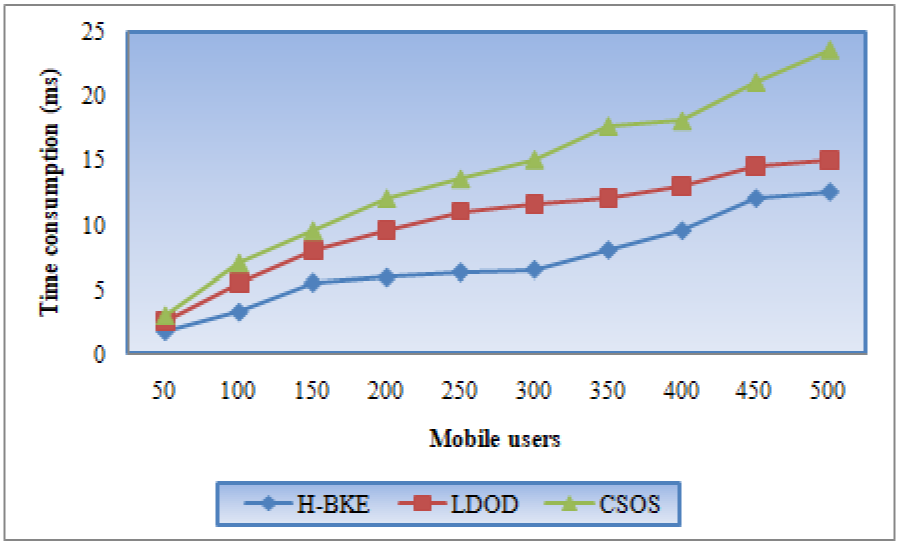


Figure 9: Performance measure of time consumption

Fig. 9 given above shows the graphical representation of time consumption with respect to 500 mobile users. Here, MUs refer to both the primary and secondary users. From Fig. 9 the time consumption of MUs for task offloading according to availability of spectrum with primary users' for performing a task is 1.75 ms for 50 mobile users using H-BKE, 2.5 ms using LDOD [Zaharia, Ciobanu and Dobre (2019)] and 3 ms using CSOS [Gnana and Maluk (2018)]. Comparedwith the LDOD [Zaharia, Ciobanu and Dobre (2019)] and CSOS [Gnana and Maluk (2018)], at least 32% and 49% of thetime consumption in offloading is saved. When considering 500 mobile users, the time consumption of three methods are 12.5 ms, 15 ms and 23.5 ms. The improvement is due to the application of Reactive Heuristic Offloading algorithm. By applying this algorithm, two important aspects are covered. They are optimal initialization of spectrum and appropriate allocation of available spectrum for secondary user for performing data transaction in MCC according to the requests made. This is performed by applying the reactive heuristic function.

4.3 Evaluation results of packet drop rate

Finally, one of the important parameters to be measured is the PDR through which security aspect is said to be inspected. During offloading, the primary users' unused spectrum is said to be utilized by the secondary user. As the data transaction is being performed by the secondary user in the primary users' spectrum, the security aspect has to be analyzed. This is evaluated in our work by means of PDR. This is mathematically expressed as given below.

$$PDR = \frac{P_{lost}}{P_{sent}} * 100 \tag{17}$$

From the above Eq. (17). The packet drop rate 'PDR' is measured based on the percentage ratio of the packets lost 'P_{lost}' to the packet sent 'P_{sent}'. It is measured in terms of percentage (%). Tab. 3 shows the PDR of H-BKE and conventional methods.

Table 4: Packet drop rate

Mobile Users	Time consumption (ms)		
	H-BKE	LDOD	CSOS
50	8	12	16
100	15	25	40
150	25	30	60
200	40	50	70
250	60	75	90
300	80	89	110
350	90	130	150
400	100	140	190
450	110	160	210
500	120	180	230

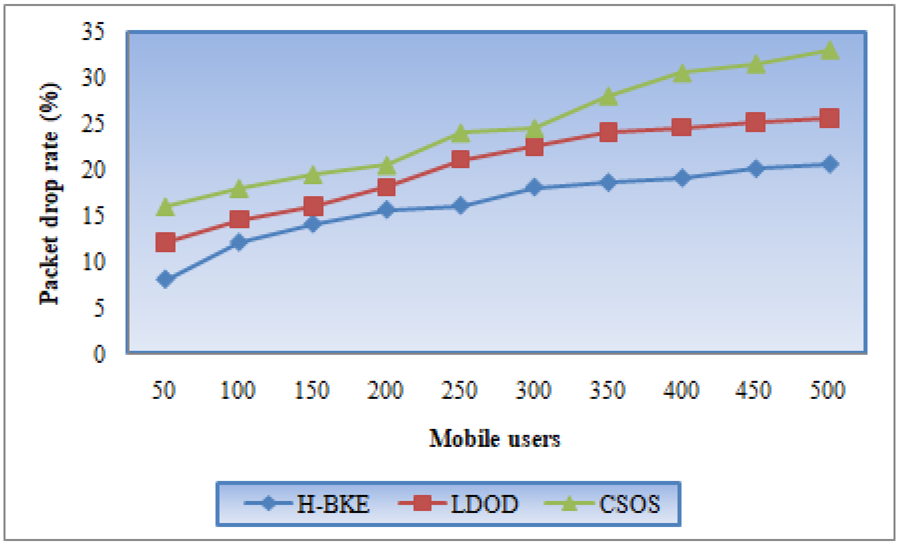


Figure 10: Performance measure of packet drop rate

PDR is depicted in Fig. 10 with number of MUs in the range of 50 to 500. When considering 50 mobile users, the PDR of H-BKE method is 8% while the existing [Zaharia, Ciobanu and Dobre (2019); Gnana and Maluk (2018)] are 12% and 16%. Although the PDR of MUs for H-BKE method is lower than the other two methods, when the number of MUs is between 200 and 350. With the growth of MUs, the PDR of conventional method is higher. With 500 MUs, the three method exhibits PDR of 20.5%, 25.5% and 33% respectively. Compared with the [Zaharia, Ciobanu and Dobre (2019); Gnana and Maluk (2018)] method, the PDR is lower using H-BKE method. This is because of the application of novel AES algorithm. By applying this algorithm, a function called Bent function is used in the Substitution Bytes step. With the use of this function, the approximation of key determination is said to be hard. Besides a variant called Rijndael Variant Shift Row is used that possesses the advantage of larger block size and hence is found to be hard to interpret. With both these concepts being used in the conventional AES algorithm, the PDR is reduced and hence found to be more secured.

5 Conclusion

The H-BKE method is proposed to provide offloading decision with optimization on reducing the energy and time consumption of MUs with improvement in security aspect. Assume interference threshold in every spectrum, the task execution delay and energy consumption, task offloading decision issue is formulated as a triplet optimization. First, offloading mechanism is designed based on the RHSO model that reduces both energy and time using 3-tuple. Next, a novel AES algorithm is designed to provide secured data transactions by designing keys that are hard to interpret and determine. The performances of H-BKE method is compared with other two methods and evaluated with performance metrics. Simulation results evident that H-BKE method provides enhanced performances in terms PDR time and energy consumption.

Funding Statement: This Research was funded by the Deanship of Scientific Research at Princess Nourah Bint Abdulrahman University, Riyadh, Kingdom of Saudi Arabia through the Fast-track Research Funding Program.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Akherfi, K.; Gerndt, M.; Harroud, H. (2016): Mobile cloud computing for computation offloading: issues and challenges. *Applied Computing and Informatics*, vol. 14, no. 1, pp. 1-16.

Al-Shayeji, M.; Ebrahim, F. (2019): A secure and energy-efficient platform for the integration of wireless sensor networks and mobile cloud computing. *Computer Networks*, vol. 165, 106956.

Chen, X.; Chen, S.; Ma, Y.; Liu, B.; Zhang, Y. et al. (2019): An adaptive offloading framework for Android applications in mobile edge computing. *Science China, Information Sciences*, vol. 62, pp. 1-17.

Duan, X.; Akhtar, A. M.; Wang, X. (2015): Software-defined networking-based resource management: data offloading with load balancing in 5G HetNet. *Eurasip Journal on Wireless Communications and Networking*, vol. 1, pp. 181.

Gnana, A. N.; Maluk, M. A. (2018): DyTO: dynamic task offloading strategy for mobile cloud computing using surrogate object model. *International Journal of Parallel Programming*, pp. 1-17.

Gu, K.; Wu, N.; Yin, B.; Jia, W. J. (2020): Secure data query framework for cloud and fog computing. *IEEE Transactions on Network and Service Management*, vol. 17, pp. 332-345.

Hoteit, S.; Secci, S.; Pujolle, G.; Wolisz, A.; Ziemlicki, C. et al. (2015): Mobile data traffic offloading over pass point hotspots. *Computer Networks*, vol. 84, pp. 76-93.

Huang, X.; Xu, K.; Lai, C.; Chen, Q.; Zhang, J. (2020): Energy-efficient offloading decision-making for mobile edge computing in vehicular networks. *Eurasip Journal on Wireless Communications and Networking*, vol. 1, pp. 35.

Junior, W.; Oliveira, E.; Santos, A.; Dias, K. (2019): A context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment. *Future Generation Computer Systems*, vol. 90, pp. 503-520.

Lan, L.; Zhang, X.; Liu, K.; Fu, J.; Jun, P. (2018): An energy-aware task offloading mechanism in multiuser mobile-edge cloud computing. *Hindawi, Mobile Information Systems*, vol. 2018, no. 4, pp. 1-12.

Li, H. X.; Li, W. J.; Zhang, S. G.; Wang, H. D.; Pan, Y. et al. (2019): Page-sharing-based virtual machine packing with multi-resource constraints to reduce network traffic in migration for clouds. *Future Generation Computer Systems*, vol. 96, pp. 462-471.

Li, W. J.; Chen, Z. Y.; Gao, X. Y.; Liu, W.; Wang, J. (2018): Multimodel framework for indoor localization under mobile edge computing environment. *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4844-4853.

Liao, Z. F.; Liang, J. B.; Feng, C. C. (2017): Mobile relay deployment in multihop relay networks. *Computer Communications*, vol. 112, no. 1, pp. 14-21.

Lin, W.; Peng, G.; Bian, X.; Xu, S.; Chang, V. et al. (2019): Scheduling algorithms for heterogeneous cloud environment: main resource load balancing algorithm and time balancing algorithm. *Journal of Grid Computing*, vol. 17, no. 4, pp. 699-726.

Medhane, D. V.; Sangaiah, A. K.; Hossain, M. S.; Muhammad, G.; Wang, J. (2020): Blockchain-enabled distributed security framework for next generation IoT: an edge-cloud and software defined network integrated approach. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2020.2977196.

Nur Idawati, M. E.; Maolin, T. (2016): A heuristic algorithm for multi-site computation offloading in mobile cloud computing. *The International Conference on Computational Science, Elsevier, Procedia Computer Science*, vol. 80, pp. 1232-1241.

Paris, S.; Martignon, F.; Filippini, I.; Chen, L. (2015): An efficient auction-based mechanism for mobile data offloading. *IEEE Transactions on Mobile Computing*, vol. 14, no. 8, pp. 1573-1586.

Peng, K.; Zhu, M.; Zhang, Y.; Liu, L.; Zhang, J. et al. (2019): An energy-and cost-aware computation offloading method for workflow applications in mobile edge computing. *Eurasip Journal on Wireless Communications and Networking*, vol. 1, pp. 207.

Rajeswari, P.; Ravi, T. N. (2018): He-SERIeS: an inventive communication model for data offloading in MANET. *Egyptian Informatics Journal*, vol. 19, no. 1, pp. 11-19.

Ranji, R.; Mansoor, A. M.; Sani, A. A. (2020): EEDOS: an energy-efficient and delay-aware offloading scheme based on device to device collaboration in mobile edge computing. *Telecommunication Systems*, vol. 73, no. 2, pp. 171-182.

Sivaram, M.; Kaliappan, M.; Shobana, S. J.; Viju Prakash, M.; Porkodi, V. et al. (2020): Secure storage allocation scheme using fuzzy based heuristic algorithm for cloud. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-020-02082-z.

Tang, Q.; Chang, L.; Yang, K.; Wang, K. Z.; Wang, J. et al. (2020): Task number maximization offloading strategy seamlessly adapted to UAV scenario. *Computer Communications*, vol. 151, pp. 19-30.

Wang, Y.; Zhu, H.; Hei, X.; Kong, Y.; Ji, W. et al. (2019): An energy saving based on task migration for mobile edge computing. *EURASIP Journal on Wireless Communications and Networking*, vol. 1, pp. 133.

Xiong, B.; Yang, K.; Zhao, J. Y.; Li, K. Q. (2017): Robust dynamic network traffic partitioning against malicious attacks. *Journal of Network and Computer Applications*, vol. 87, pp. 20-31.

Xu, Z.; Liu, X.; Jiang, G.; Tang, B. (2019): A time-efficient data offloading method with privacy preservation for intelligent sensors in edge computing. *Eurasip Journal on Wireless Communications and Networking*, vol. 1, pp. 1-12.

Yao, J.; Zhang, K. M.; Dai, Y. X.; Wang, J. (2018): Power function-based signal recovery transition optimization model of emergency traffic. *Journal of Supercomputing*, vol. 74, no. 12, pp. 7003-7023.

Yunsik, S.; Yangsun, L. (2017): Offloading method for efficient use of local computational resources in mobile location-based services using clouds. *Mobile Information Systems, Hindawi*, vol. 2017, pp. 1-9.

Zaharia, G. E.; Ciobanu, R. I.; Dobre, C. (2019): Machine learning-based traffic offloading in fog networks. *Simulation Modelling Practice and Theory*, vol. 101, 102045.