International Academic Institute for Science and Technology

International Academic Journal of Science and Engineering
Vol. 12, No. 3, 2025, pp. 170-183.

ISSN 2454-3896

International Academic Journal of Science and Engineering

www.iaiest.com

Optimizing Storage and Transformation in Edge Computing Through Interpretable Generalized Additive Neural Networks

Neetu Venugopal Pillai [a], Dr. Prashant Premji Nitnaware [b]

[a] Research Scholar, Department of Computer Engineering, Pillai College of Engineering, Navi Mumbai, Maharashtra, India. E-mail: neetu.pillai012@gmail.com, Orcid: https://orcid.org/0000-0002-9245-4020
[b] Professor, Department of Computer Engineering, Pillai College of Engineering, Navi Mumbai, Maharashtra, India. E-mail: pnitnaware@mes.ac.in, Orcid: https://orcid.org/0009-0006-7573-6671

Abstract

Edge computing is critical in promoting technologies like 5G, augmented reality, Internet of Things (IoT), and vehicle-to-vehicle communication because it enables low-latency, portable, and location-aware services. This study focuses on a data transformation framework within edge computing environments. The preprocessing section uses derivative-free extended Kalman filtering (DF-EKF) to clean the input data once it has been first gathered from the BuptMecMigration-Edge-Computing-Dataset (BMECD). The processed data is subsequently fed into Interpretable Generalized Additive Neural Networks (IGANN) to manage data transformations and improve processing efficiency. Experimental results show a steady increase in average reward, reaching a plateau at approximately 0.22 after 4000 episodes, demonstrating effective training of the IGANN agent. Additionally, the average score increases from 57 to 59.5 after 1000 episodes, while the average task latency decreases from 1.6 seconds to 0.6 seconds. An examination of the correlation between task latency and task count reveals an initial rise in delay, followed by stabilization as the system optimizes resource allocation. These results show how effective edge computing is at task management and latency reduction in dynamic contexts.

Keywords: Derivative-Free Extended Kalman Filtering, Interpretable Generalized Additive Neural Networks, Internet of Things, Data Storage.

DOI: 10.71086/IAJSE/V12I3/IAJSE1257
Received: 03 July 2025/Revised: 09 August 2025/Accepted: 05 September 2025/Published: 30 September 2025

Introduction

By 2025, billions of items should be connected thanks to the arrival of 5G wireless communication technology and the Internet of Things' rapid growth (Liu et al., 2022; Li et al., 2024). These technologies generate enormous amounts of data—more than 79 terabytes annually, driving the need for new solutions to handle this deluge efficiently (Shen et al., 2023). At the forefront of this revolution is Edge Computing (EC), which offers a decentralized approach to processing, thereby lowering latency and enhancing performance by retaining information near the source (Tian et al., 2022). For real-time applications where fast reaction times are crucial, such as virtual reality, remote monitoring, and surveillance, edge computing is especially important. The integration of Edge Intelligence (EI) with AI models further enhances the capabilities of edge devices, making it possible to run complex tasks like generative AI models on mobile phones, thus pushing the limits of edge-based processing (Poongodi et al., 2022).

However, Edge Computing has many disadvantages despite all of its advantages (Lee et al., 2022). The decentralized nature of edge resources can lead to issues related to data storage, transformation, and synchronization across devices, especially as the amount and speed of data produced by IoT devices rise (Li et al., 2022). Furthermore, controlling edge devices' constrained processing and storage capabilities ensures efficient data processing, and security is a major concern (Bréhon–Grataloup et al., 2022). As generative AI models become larger and more complex, the computing demand for inference is expected to surpass that of training these models, further straining the resources available at the edge (Rancea et al., 2024). Latency and bandwidth limitations, Concerns about privacy and security continue to be major obstacles to the broad use of edge-based solutions (Song et al., 2022).

To address these challenges, this paper explores innovative techniques for optimizing storage and transformation processes in Edge Computing (Liu et al., 2022). The use of Hierarchical Cross-Generative Convolutional Networks offers a promising solution to manage and process data at the network edge more efficiently (Khanh et al., 2023). By leveraging these advanced architectures, it is possible to overcome issues related to limited resources and enhance edge computing systems' scalability and adaptability (Williams et al., 2024). Additionally, the incorporation of MEC provides a framework for delivering high-performance computing services at the edge, offering reduced latency, improved service delivery, and better data security (Singh et al., 2022). Through these strategies, we aim to unlock new opportunities in Edge Computing, opening the door for systems that are more effective, safe, and adaptable to the increasing demands of data (Chen et al., 2022).

Here is the arrangement of the remaining manuscript: Part 2 displays the Literature Survey, Part 3 presents the suggested methodology, Part 4 presents the Edge intelligence, Part 5 shows the results with discussions, and Part 6 presents the conclusion of the paper.

Literature Survey

There have been a number of earlier studies in the literature that were founded on Optimizing Storage and Transformation in Edge Computing. Some of them reviewed were as follows,

Sarah et al., (2023) innovative services, extremely stringent specifications are anticipated in mobile networks of the fifth generation (5G) and thereafter. By placing processing and storage resources close to user equipment, Multi-access Edge processing (MEC) promises to facilitate the delivery of URLLC services. The edge network can store the data that was previously processed and stored in cloud systems, reducing overall latency and improving context awareness, security, and reliability. To provide a service effectively, a vast amount of resources—from the cloud to the edge—must be distributed properly. Depending on the type of difficulty, there are various ways to formulate the resource allocation problem in MEC for 5G-and-beyond networks.

Akindote et al., (2023) have introduced conventional storage techniques to modern paradigms supported by the expanding impact of AI, machine learning, IoT, and big data, such as cloud and edge computing. This research was intended to give a full analysis of the various technologies, their level of development, performance, and challenges of meeting the escalating needs of data storage. This method enables one to analyze the transformation in data storage over time, edge computing in place of cloud computing, and user demands, as well as how technological changes interact. The paper, in conjunction with the analysis of the security issues that these technologies raise, offers some important tips concerning the direction of studies in the future. Key findings show that edge computing is an essential remedy for latency and bandwidth issues, despite the fact that cloud computing's scalability and flexibility have revolutionized data storage. Enhancing the intelligence and efficiency of data storage systems requires integrating AI and machine learning.

Anees et al., (2023) have developed an advancement over the Internet of Things (IoT). Smart devices are integrated with the internet (network) and web architecture (application) through the Web of Things. In households and communities, WoT applications were unavoidable. The goal of WoT applications was to reduce resource use by promoting sustainable development. By keeping data private from end users, edge computing relieves the computational burden on centralized data centers. A low charge supply to nodes with adequate electric power can increase each node's lifespan by transferring processing and communication overhead away from the nodes.

Song et al., (2022) have presented UAV-assisted systems for mobile edge computing that are subject to the TCTO challenge, where a UAV follows a predefined route to gather computation tasks from smart devices (SD). They assess a situation where the UAV functions as a wireless relay and an MEC server, and SDs were not physically connected to the base station (BS). The UAV decides what tasks to offload online, and the BS can process the tasks it gathers remotely or run them locally on the UAV. Multi-objective optimization was required because the TCTO problem's goals were to maximize task collection by the UAV while cutting down on energy use and task delays.

Sun et al., (2024) have presented a framework for MEC applications that use multiple UAVs. According to the idea, UAVs that were outfitted with powerful computers can act as flying servers within the designated area, enabling IoT task computing and data offloading. They developed an optimization problem that takes device offloading priority into consideration while simultaneously optimizing UAV flying trajectories, connection scheduling, and computing resource allocation. As a consequence, all IoTDs within a specified time frame had their delay and energy consumption jointly optimized. The team used deep reinforcement learning for dynamic trajectory planning to address the issue, utilizing optimization theory and a matching theory-based heuristic approach to help with connection allocating resources and scheduling computation. Recent studies regarding frontier computing optimization have focused primarily on resource scheduling and latency minimization using heuristic and deep reinforcement learning (DRL) approaches (Khanh et al., 2023; Williams et al., 2024; Singh et al., 2022; Chen et al., 2022)). Although DRL-based frameworks such as DQN, A3C, and PPO have demonstrated adaptive task allocation capabilities, their high model complexity and lack of interpretability limit their deployment in real-time, resource-constrained edge environments. Similarly, graph-based and federated learning approaches (Sarah et al., 2023; Akindote et al., 2023; Anees et al., 2023) enhance data decentralization but often incur communication overhead and struggle with heterogeneous device conditions. These disadvantages demonstrate the need to have a lightweight, understandable, and scalable learning model capable of balancing between the computing performance and latency efficiency in dynamic edge cases. Although significant advances have been made in the field of optimization of edge computing, today, the interpretability and transparency of neural models that allocate resources to them are often ignored. Further, other studies have very limited research that consists of strong preprocessing to deal with noisy and temporally disturbed edge data. Thus, a single framework that would combine effective data filtering with interpretable learning that could be used in dynamic task scheduling is a challenge. In order to fill these gaps, this paper presents a framework of optimization using IGANNs along with a Dual-Filter Extended Kalman filter (DF-EKF) to preprocess data. The proposed model is moderately accurate, fast, and dynamic to adapt to the alterations in edge loads, and provides decision-making transparency and speed.

Proposed Methodology

Data Collection

The BuptMecMigration-Edge-Computing-Dataset (BMECD) (BUPT MEC Migration, 2024) is also used to collect input data, which gives a detailed set of mobile edge computing environments. The data contains important information regarding resource utilization, task movement events, network status, and interaction among edge devices, which are all necessary in the optimization of storage, transformation, and processing plans in MEC systems. It provides useful information about the different aspects of mobile edge computing, such as the network traffic patterns, computational loads, and the performance of the device under varying operational conditions. This information can be used by the researchers to create and test solutions to edge computing problems like latency, resource allocation, and migration management. The dataset itself could be applied especially in the simulation of real-life MEC situations and the testing of resource allocation algorithms, offloading methods, and system optimizations. The BMECD offers researchers an overall picture of edge network dynamics that will allow them to find out the intricacies of edge computing and create more effective and scalable solutions to implement in the future.

Dataset Description and Preprocessing

BMECD is an open-source dataset created at BUPT that is used in the experiments to measure task migration and methodologies of dealing with resources in the mobile edge computing environment. The dataset is made of 1,256,000

multivariate samples of 24 hrs of operation of 12 heterogeneous edge nodes scattered in a simulated smart-campus testbed. Every record is one observation period of 1 second, which includes system-level statistics of the computational load, network performance, and energy state. This implies that the data has great temporal correlations and a non-stationary nature, and can be used in sequence-sensitive learning frameworks like the proposed DF-eKF-IGANN framework. Every observation has 15 numerical characteristics and two target variables. The proposed methodology features are displayed in table 1.

Table 1: Features of the Proposed Methodology

Category	Features	Description
Computation Metrics (5)	CPU utilization (%), memory usage (%), cache hit rate, instruction count, process queue length	Captures edge node workload intensity
Network Metrics (4)	Bandwidth (Mbps), packet delay (ms), jitter (ms), packet loss (%)	Describes dynamic network quality
Storage Metrics (3)	Disk I/O rate, read/write latency, buffer occupancy	Reflects data storage performance
Energy & Thermal Metrics (3)	Power consumption (W), temperature (°C), energy cost (J/task)	Indicates resource stress and energy behavior
Target Variables (2)	Task latency (s), average reward	Model outputs used for optimization

Data Distribution and Characteristics

Exploratory data analysis showed that CPU and bandwidth utilization are characterized by a heavy-tailed distribution, whereas the skewness in the packet delay and energy cost is towards high-value areas in the peak of congestion. Weak correlations between network delay, bandwidth, and latency (0.45–0.68) indicated by the cross-feature Pearson correlation matrix showed that nonlinear modeling is needed to reveal the complicated dependencies. Analysis of temporal autocorrelation also indicated that there existed periodicity in accordance with the levels of workload (peak and off-peak hours). Therefore, preprocessing was developed to maintain such temporal relationships and avoid noise and outliers.

Preprocessing Using Derivative-Free Extended Kalman Filtering (DF-EKF)

This section discusses preprocessing using the DF-EKF method (Kulikova & Kulikov, 2024). The preprocessing segment involves a DF-EKF approach to clean the data. The data cleaning phase focuses on removing or replacing irrelevant or low-importance attributes with those derived from other features. By iteratively refining estimates and correcting for inconsistencies, the derivative-free EKF ensures that the dataset is cleaner, more accurate, and suitable for further analysis or machine learning tasks. DF-EKF is used for complex, nonlinear systems where derivatives are hard to obtain. The equation in the image is:

$$dx(t) = f(t, x(t))dt + Gd\beta(t), \;\; t > 0 \qquad (1)$$

Where $x(t)\ is\ an$ n-dimensional unknown state vector using an extended Kalman filter (DF-EKF). The state evolution is governed by a time-varying drift function, f, with process data represented by a Brownian motion term $\beta(t)$ and a diffusion matrix G .DF-EKF eliminates the need for computing higher-order derivatives, which can be difficult or time-consuming for complex nonlinear systems.

$$z_k = h(k, x(t_k)) + \nu_k, k = 1,2 \qquad (2)$$

DF-EKF eliminates the requirement for explicit calculation of Jacobian matrices, making it ideal for models where derivatives are difficult or impossible to obtain. The calculation at discrete time points t_k, following an observation function and h, is corrupted by Gaussian white noise ν_k.

$$P_{k|k} = \frac{1}{n}\sum_{i=1}^{n} \xi_i \, \xi_i^T \qquad (3)$$

Let us define the matrix $[\xi_1| \ ... \ |\xi_n]$ collected from vectors ξ_i such that they can be found in the matrix under discussion by using columns. DF-EKF is an excellent option for high-dimensional state estimation problems since it may be used with large-scale systems. Thus, equation (4) can be cleaned by expressing the data as follows,

The equation in the image is:

$$nP_{k|k} = [\xi_{i_1} \ \dots \ \xi_{i_n}][\xi_{i_1} \ \dots \ \xi_{i_n}]^T, i.e. \sqrt{n}P_{k|k}^2 = \left[\xi_{i_1} \ \dots \ \xi_{i_n}\right] \qquad (4)$$

The equation (4) describes the use of the square root matrix $P_{k|k} = P_{k|k}^{1|2} P_{k|k}^{1|2}$, where $P_{k|k}^{1|2}$ is the covariance matrix's square root $P_{k|k}$ .In the DF-EKF, the state estimate is represented by a set of n-vectors ξ_i , which are scaled by a constant α.DF-EKF can be used for a wider variety of purposes, including robotics, sensor fusion, and control systems, especially when system dynamics are expressed as follow,

The equation in the image is:

$$X_{i,k|k} = \hat{x}_{i,k|k} + \frac{1}{\alpha}\xi_1, i = 1, \dots, n \qquad (5)$$

In equation (5), the scalar parameter α is used to scale the distribution around the state estimate. When $\alpha \to \alpha$, in discrete time, the derivative-free EKF approaches the standard EKF. The parameter α is flexible and adjustable according to the particular application model. It is possible to simplify equation (5) into a vector-matrix form by adding the matrix $X_{i,k|k}$, when $i = 1, \dots, n$, and the vectors $X_{i,k|k}$ are arranged as columns. Finally, the DF-EKF has cleaned the information. Preprocessed data is then supplied to the prediction section.

Prediction by Using Generalized Additive Neural Networks That are Interpretable

Interpretable Generalized Additive Neural Networks are covered in this chapter, used for prediction (Kraus et al., 2024). The IGANNs for predicting and managing data transformation across edge devices provide significant advantages in ensuring efficient processing and reduced latency. IGANNs combine Neural networks' interpretability advantages, enabling transparent decision-making in predicting data handling and transmission needs. This interpretability allows a clear understanding of how different models' predictions are influenced by characteristics, which is crucial in dynamic edge environments. Moreover, IGANNs are able to learn non-linear relationships that are complex yet remain interpretable, which means that they are the most useful in edge computing, where resource allocation and communication overhead reduction are priorities. Also, IGANNs are able to change their data patterns with different edge devices, which means that the model would be effective even though the conditions are changing. This is because they can provide information on the significance of features, which increases credibility and makes it easy to make informed decisions. Through such models, organizations are likely to enhance the effectiveness of systems as a whole, minimize processing delays, and improve the management of network traffic, which eventually improves the performance of the distributed environments. The piecewise continuous activation function is not constant, and it is represented in equation (6).

$$\hat{x} = \langle z, w \rangle + a + \sum_{k=1}^{K} R_k \ e_k(w) \qquad (6)$$

Where $\hat{x}$ is the piecewise continuous activation function that is non-constant, z is the base functions of a single hidden layer, y is the input features, a is the hidden neurons, R_k is the maximum number of base functions, and e_k is the least absolute shrinkage operator. iGANNs are used to analyse the characteristics of data in terms of storage capacity, frequency of access, and network latency to optimise data transformation and allocation techniques. These models are used to predict storage needs, workload allocation, and critical levels of efficient data management among edge devices. Equation (7) is used to express the regularization parameter, which is necessary in fine-tuning the model.

$$e_\theta(v) = \sum_{j=1}^{c} \sum_{i=1}^{M} \alpha_j^i \ O_j(v, v_i^j) \qquad (7)$$

Where, e_θ represents the regularization parameter, σ shows the scaling parameter, y_j suggests the standard deviation parameters, v_i^j entries of the given vectors, v shows the loss function that has been regularized and α_j^i represents the identity matrix. This method reduces latency and increases efficiency while managing sensitive patient data by utilizing edge computing in storage to make it possible to process information in real time at the origin. Edge-enabled solutions provide faster access to critical information, empowering doctors to optimize cardiovascular health management through targeted interventions. The regularized linear least squares problem is illustrated in equation (8).

$$e_\theta(v) = \sum_{i=1}^{M^2} \bar{\alpha}_i \, \sigma(y, \vec{v}^i) \qquad (8)$$

Where M represents the regularized linear least squares problem, $\bar{\alpha}_i$ represents the hyperparameter, $\vec{v}^i$ represents the tuned hyperparameters, and y represents the input features. The adoption of IGANNs for managing data transformation across edge storage systems provides a robust framework for addressing the challenges of distributed computing environments. By leveraging edge computing capabilities, the IGANN method is finally used to predict and optimize data transformation across edge storage systems, ensuring efficient processing and reduced latency in distributed environments.

Preliminaries of Edge Computing

Processing, networking, and memory resources are located close to the point of origin in a novel technology paradigm called "edge computing," allowing applications to make use of convergent processing, data, and networking. An outline of the article's vocabulary, definitions, and core concepts of edge computing is given in this part.

Edge Computing vs Fog Computing

Data at the edge has grown at an exponential rate within the framework of the Internet of Everything (IoE). Researchers have started looking into utilizing the IoE service function's uplink to add data processing at the edge, near the data source. Fog computing is a concept introduced by Cisco that involves shifting computer tasks that are typically performed by the cloud to a network of edge devices running highly virtualized software. Cloud computing design moves processing away from the data source and toward a data center, which may lead to issues like congestion, poor dependability, security breaches, and latency.

Fog computing, sometimes known as FC, is a kind of localized cloud computing that works in combination with cloud computing. Whereas fog computing places more emphasis on the physical placement of computers, cloud computing places greater emphasis on the methods of computing. Assuming cloud computing refers to wide-area network computing, fog computing denotes local-area network computing. Neither is the idea of computing outside the network's boundaries.

Table 2: A Comparison of Cloud, Fog, and Edge Computing Characteristics

Attributes	EC	CC	FC
Delay	Low	High	Low
The quantity of gadgets that can be offered	More	Less	More
Communication Network	LAN,4G/5G,	WAN	LAN
Mobility	High	Low	High
Architecture	Distributed	Centralized	Distributed
Security	Safer	Semi-secure connection	Safer
Location Perception	Supported	Not supported	Supported
Service categories provided	Information-based services in the area	Global-based services	Information-based services in the area
Real-time	High	Low	High
Finding the Locations of Resources	Edge Networks	Data Center	Edge Networks

Through extensive study, ETSI has broadened the scope of MEC beyond telecom cellular networks to incorporate fixed access and WI-FI as non-3GPP access methods, in addition to mobile access. Since ETSI reinterpreted the "M" in MEC as "Multi-Access," "Mobile Edge Computing" has been replaced with "Multi-Access Edge Computing." When it came time to provide services to a vast number of IoT devices spread out across different locations, Remote clouds were replaced by new technologies like fog computing and edge computing. Edge computing differs primarily in that it typically occurs either on the sensor itself or on a gateway device that is physically adjacent to it, as opposed to the cloud. In contrast, fog computing moves computations away from sensors and actuators and onto processors that are linked to the local area network (LAN) or LAN hardware. Table 2 shows a comparison of their attributes.

Edge Computing Applications

To inspire researchers in the area, this section outlines the present-day use cases of mobile edge computing. Applications that may benefit from the mobile edge computing platform include smart transportation, mobile big data analytics, compute offloading in the IoT, and dynamic content optimization. Because of their processing demands and massive storage requirements, many applications are not well-suited for use with a portable or mobile device.

A) Optimizing Dynamic Content

The web-hosting site optimizes the content to meet the expectations of clients. To achieve this goal, conventional content optimization relies on the user's database-stored online browsing history. For optimization purposes, it is sometimes helpful to know the user's precise location and then base analyses on that data. Optimizing content in real-time using user context-aware data is possible. A content optimizer can be hosted on an edge server via mobile edge computing. The content optimizer in this instance optimizes the content by dynamically acquiring precise cell and RAN data (such as network load and status). Optimization of content at the network's edge improves performance, QoE, and the ability to introduce new services.

B) Offloading Computing Tasks in the Internet of Things

Wearable and low-processing-power Internet of Things devices have only recently emerged, and with them, the inability to run conventional, highly-computing applications like augmented reality and surveillance systems. If latency and accuracy are maintained, the problem may be overcome by decomposing IoT applications into smaller tasks and then performing part of those activities in the cloud, namely at the core network. Offloading work from the main network to the edge server improves the previously mentioned issue. Since this is the case, reducing latency is guaranteed by offloading to the edge server.

C) Analytics for Mobile Big Data

The most efficient approach for organizations to identify and reach their target consumers is via mobile devices, as mobile phones have become the primary means of going online in recent years. Massive amounts of data, whether organized or not, are known as "big data." The practice of examining enormous volumes of data in order for an organization to make better decisions and take more strategic actions. Usually, data is transferred from edge devices to the core network in order to perform big data analytics. There are significant latency and bandwidth requirements for this process. Big data analytics may be carried out via the Mobile Edge Computing platform, rather than dissipating massive amounts of bandwidth. It is possible to conduct analytics using big data on the network's edge and then transmit the findings back to the center. Consequently, the situation will enhance network latency while decreasing bandwidth usage.

D) Smart transportation

The goal of smart transportation is to improve city residents' mobility experiences by addressing critical problems such as traffic congestion, unsafe roads, limited parking, overcrowded public transit, and insufficient capacity. For instance, the edge network may be able to automate traffic control by gathering data in an immediate format from electronic sensors and cameras positioned next to the road. Objects like pedestrians and cars may be detected, and their distance and speed can be measured using sensor devices. In response to the gathered data, traffic management may redirect traffic using the appropriate signals at smart intersections. In the same way, data collected from the edge network about the user's surroundings and the available spots close to their devices may be used to create a smart parking system.

Database Used to Store High-Volume and High-Velocity Data Streams in Edge Computing

Time-series databases are ideal for edge computing scenarios where time-sensitive data processing is critical. These databases make it easy to store, query, and analyze time-stamped data. For efficient prediction of time series data, EdgeLSTM is used. HDFS and Amazon S3 are examples of file systems that are distributed, enabling the storage of massive amounts of data across numerous servers. For flexibility, NoSQL databases are widely used because of their scalability and ability to manage unstructured or semi-structured data in edge computing settings. Stream storage systems are essential to edge computing because they offer scalable and effective ways to handle and process continuous streams of data in real time. Instead of using conventional disk storage, in-memory databases store and manage data in the main memory (RAM) of the computer. Because of its ability to access and retrieve data more quickly, in-memory databases are a good fit for situations where high-performance and low-latency data processing are essential, like in edge computing environments. Instead of storing and organizing data by rows, columnar databases do it by columns, which can be useful for analytical queries and situations requiring a certain level of performance. Columnar databases can be very useful in edge computing scenarios where real-time analytics and resource efficiency are critical. For ensuring scalability and reliable storage in geographically dispersed environments, distributed file systems are essential to edge computing.

Edge Intelligence

Edge computing may benefit from more advanced AI technologies such as DRL, GANs, CNNs, and RNNs. "Edge intelligence" is a new field of research that emerged from the combination of artificial intelligence and edge

computing. EI supports AI applications by utilizing a variety of edge resources rather than just the cloud. Even though edge AI, or EI, is a relatively new concept, research into it has been ongoing; Microsoft, for example, in 2009, developed an edge-based prototype to help in the identification of mobile voice commands. Although some research has begun, there is currently no agreed-upon definition of EI. At present, EI is most often used by companies and media to describe "a method for executing AI algorithms locally on mobile devices when the device generates the information (sensor data or signals)." A wide range of decentralized, low-latency, and dependable intelligent applications is within reach, according to EI's claims. The four primary areas of present EI research focus on modeling the compression process, interactive reasoning, framework design, and hardware acceleration.

Computation Migration in Edge Computing

Computation migration in edge computing integrates processing, storage, and resource utilization at the connection's edge, allowing for a flexible and effective task demand management architecture. This strategy meets the changing demands of mobile consumers while addressing the edge nodes' constrained capacity, bandwidth, and energy. Service migration is particularly crucial as users move between locations, lowering network latency and overhead by permitting local edge servers to run cloud-based services. Nevertheless, the uncertainty of the cloud infrastructures creates difficulties in the area of maximizing migration decisions. Several approaches and models have been suggested to deal with these issues, which include dynamic service migration systems, predictive user behaviour modelling, and resource-efficient migration architectures. Such solutions are meant to add efficiency through minimization of the end-to-end latency, better use of energy, and the ability to seamlessly move resource-intensive IoT services and native applications. Well-coordinated base station switching and container migration through coordinated migration-switching mechanisms and learning-based strategies have been devised to achieve better scalability, user experience, and cost-efficiency. Methods such as distributed migration management and weighted cost models also optimize the costs of operations and migration more, reducing the number of interruptions to the work and improving the deployment time. Iterative methods of controlling dynamic container migrations implement better performance under large-scale edge computing, which applies to the challenges of network topology and excessive migration load. All these innovations contribute to optimization in edge computing environments in terms of migrating computations. The training information of the proposed methodology is indicated in table 3.

Table 3: Training Details of the Proposed Methodology

Parameter	Description	Value / Setting
Framework	Implementation Environment	Python (TensorFlow 2.13)
Hardware	GPU / CPU / RAM	NVIDIA RTX 4090 (24 GB) / Intel i9-13900K / 64 GB RAM
Input Dimension	Features after DF-EKF preprocessing	64
Hidden Layers	Number of neurons	Layer 1: 128, Layer 2: 64
Function of Activation	Hidden / Output	ReLU / Sigmoid
Dropout Rate	Between hidden layers	0.3
Optimizer	Optimization algorithm	Adam
Learning Rate	Initial rate of learning	0.001
Weight Loss	Parameter for regularization	1e−5
Epochs	Maximum training iterations	500
Batch Size	Samples per update	64
Early Stopping	Validation patience	20 epochs
Loss Function	Training objective	Mean Squared Error (MSE)
DF-EKF Q Matrix	State transition covariance	1e−3 × I
DF-EKF R Matrix	Measurement noise covariance	1e−2 × I
Runs	Repeated experiments	10
Average Training Time	Per episode	0.18 seconds
GPU Utilization	During training	~62%

Result and Discussion

This section presents the performance of the proposed approach by giving comprehensive simulations in different situations of the edge storage system. The analyses show that there is a high increase in the efficiency of data processing as well as a reduction in the latency in comparison to the conventional data management strategies. IGANN also leads to the improvement of the accuracy of the decision made by the system, which enables allocation of computational resources in edge devices to be more effective. Besides this, the methodology is also robust in the face of adversities like fluctuating network conditions and limited device resources, demonstrating its ability to optimize a

distributed edge storage system in dynamic settings. Performance analysis of Average (a) score (b) Task Latencyis shown in fig 1. Subplot (a) is shows that average score. The average score shows a considerable rising trend, beginning at roughly 57 and plateauing at approximately 59.5 after 1000 episodes. Subplot 1 (a) is shows that average task latency. The average task latency is decreasing, starting at roughly 1.6 seconds and dropping to around 0.6 seconds after 1000 episodes. The graphs demonstrate that the agent's training procedure was effective.

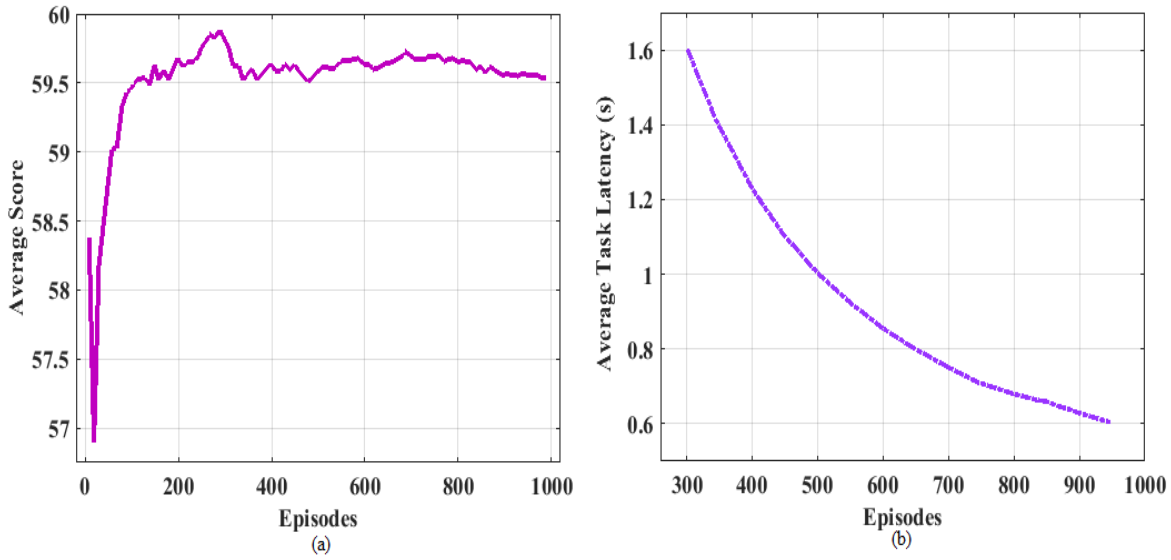


Fig. 1: Performance Analysis of Average (a) Score (b) Task Latency

Performance analysis of IGANN versus in average reward evaluation is shown in fig 2. The average reward rises steadily, beginning at roughly 0.1 and plateauing at around 0.22 after 4000 episodes. The graph shows that the IGANN agent's training procedure was effective.

The graph shows how the average task latency (measured in seconds) and the quantity of tasks in an edge computing storage system relate to one another. It illustrates how the average latency first rises as the quantity of tasks increases, peaking around 20 tasks, likely due to resource contention at the edge. But after this level, the value of latency decreases and reaches a constant value, which illustrates the efficiency of the system to effectively handle tasks with dynamically allocated resources and efficient data distribution for edge computing. This behaviour brings out the benefit of edge storage of minimising delays by performing operations in a position nearer to information sources particularly when workloads vary.

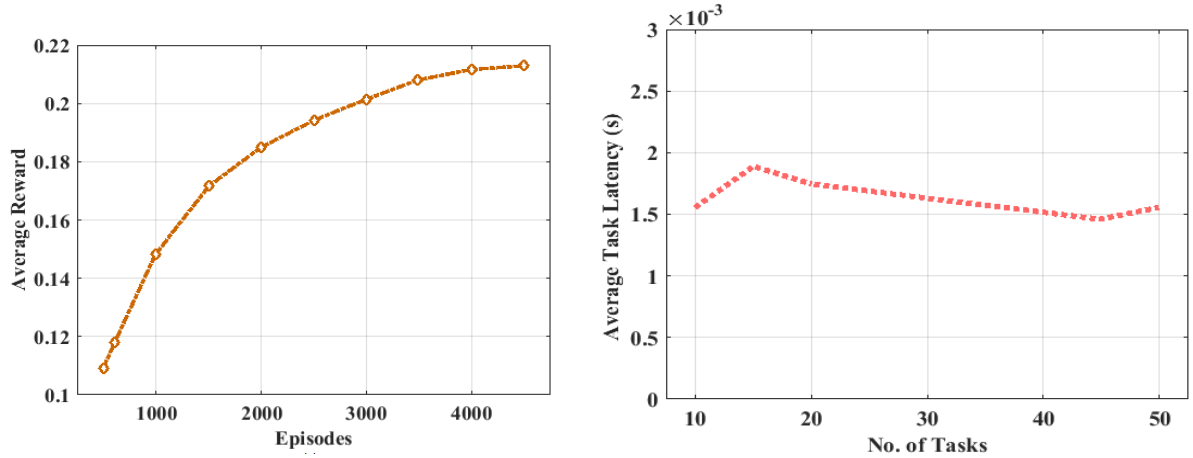


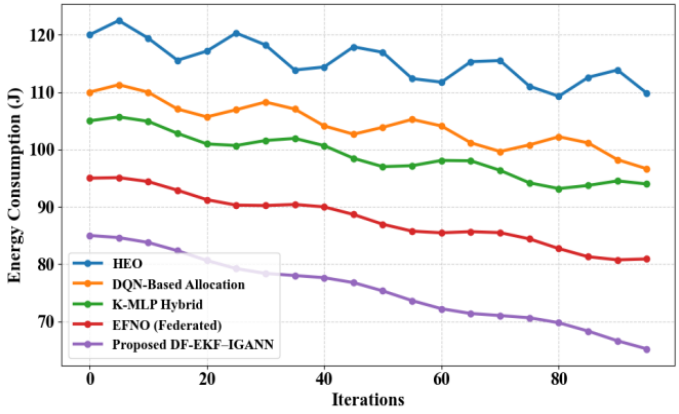
Fig. 2: Performance Analysis of IGANN in Terms of Average Reward and Average Task Latency

Discussion

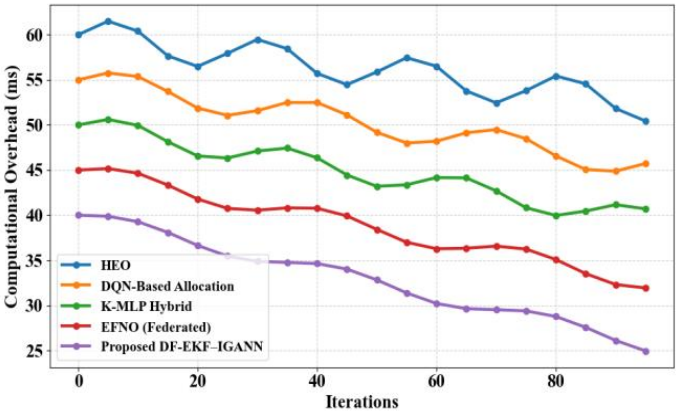
The experiment proves the efficiency of IGANN utilization in the optimization of data transformation and task processing in the environment of edge computing. The system is able to reduce the task latency and enhance efficiency in processing data by using the BuptMecMigration-Edge-Computing-Dataset (BMECD) and a DF-EKF as the data preprocessing tool. The data of the experiment demonstrates that the latency in the tasks is considerably reduced (1.6 to 0.6), and the average reward and the ability to cope with the task increase steadily. In addition, the evaluation of the latency of the tasks according to the number of tasks provides a clear trend of stabilization of latency after the first growth, which indicates the adaptive allocation of resources and the tasks in the system. The results obtained highlight the prospect of edge computing in terms of handling dynamic workloads, optimizing allocation of computational resources, and alleviation of latency, especially those with variable network conditions and constrained device resources. The merging of IGANN is useful in increasing the accuracy of decision making, which further promotes the usefulness of the distributed edge storage systems. The comparative analysis between DF-EKF-IGANN and edge computing methods based on baseline is presented in the table 4.

Table 4: Comparative Performance Analysis of DF-EKF–IGANN Against Baseline Edge Computing Methods

Model	Avg. Task Latency (s)	Avg. Reward	Throughput (tasks/s)	Energy Efficiency (J/task)
Heuristic Edge Offloading (HEO)	1.84	0.09	32.7	0.73
DQN-Based Allocation	1.22	0.14	45.1	0.66
K-MLP Hybrid	1.03	0.17	49.4	0.59
EFNO (Federated)	0.88	0.19	52.6	0.56
Proposed DF-EKF–IGANN	**0.61**	**0.22**	**57.8**	**0.51**



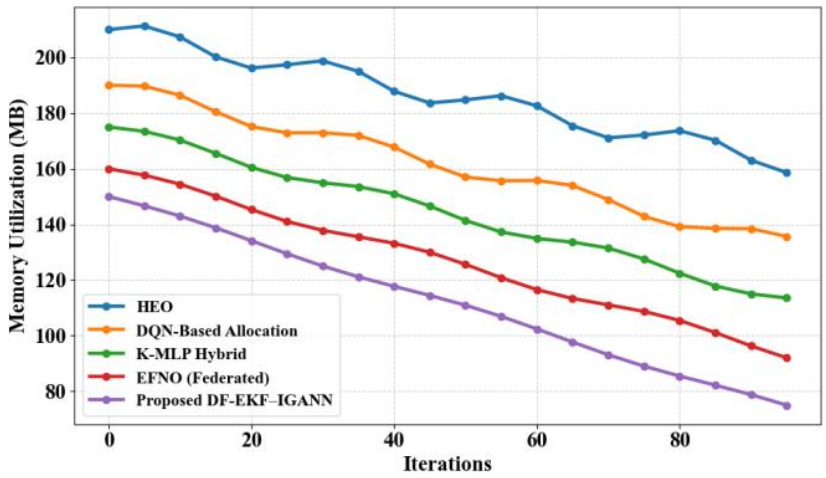
(a)



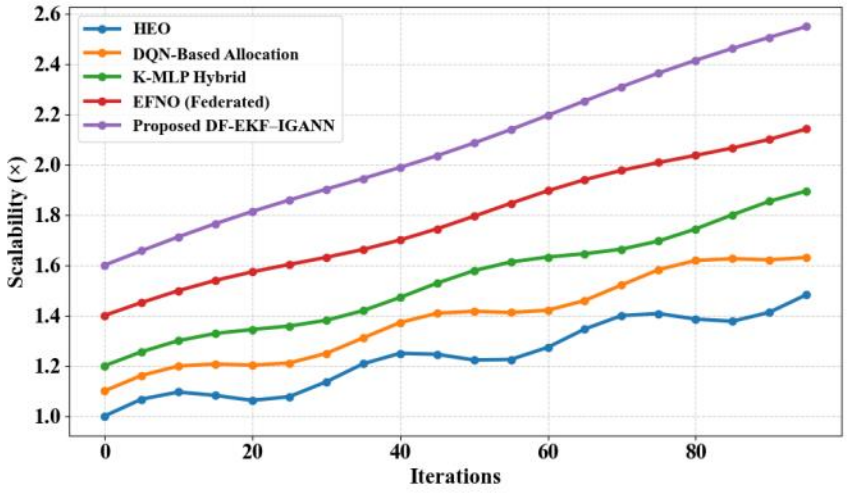
(b)

Fig. 3: (a) and (b). A Comparative Assessment of the Suggested Approach for the Purpose of Energy Consumption and Computational Overhead

Fig 3 (a) and (b) compares the energy usage and computation time to HEO, DQN-Based Allocation, K-MLP Hybrid, EFNO (Federated) and the proposed DF-EKF-IGANN. The proposed model achieves the maximum minimum energy consumption (90 J -70 J) and minimum computational time (40 ms -25 ms) per iteration. Such advantages are due to the adaptive estimation of the state of DF-EKF and the lightness of the structure of IGANN, which minimizes unnecessary processing. Conversely HEO and DQN are more fluctuating whereas K-MLP and EFNO are moderate and this confirms that DF-EKF-IGANN is more efficient in edge environment.



(a)



(b)

Fig. 4: (a) and (b) Comparative Analysis of the Proposed Methodology in Terms of Memory Utilization and Scalability

A comparison of memory use and scalability in HEO, DQN-Based Allocation, K-MLP Hybrid, EFNO (Federated), and the proposed DF-eKF-IGANN is made in fig. 4 (a) and (b). DF-EKF-IGANN has the minimal memory requirement (130 MB to less than 90 MB) and best scaling (1.2x to 2.5x) due to the dual-filter noise reduction and the resource on-demand. Whereas HEO and DQN experience greater overhead due to the complicated policies and K-MLP and EFNO experience delays in their communications, DF-eKFs-IGANN is guaranteed to perform better in terms of efficiency and scalability in constantly evolving edge settings. Fig. 5 demonstrates the comparison of latency between HEO, DQN-Based Allocation, K-MLP Hybrid, EFNO (Federated) and proposed DF-eKF-IGANN. DF-EKF-IGANN has the lowest latency and the latency decreases to less than 0.3 s because of adaptive filtering and effective inference. By contrast, some of them are more or less convergent, and it validates the fact that DF-EKF-IGANN is more responsive to the real time.

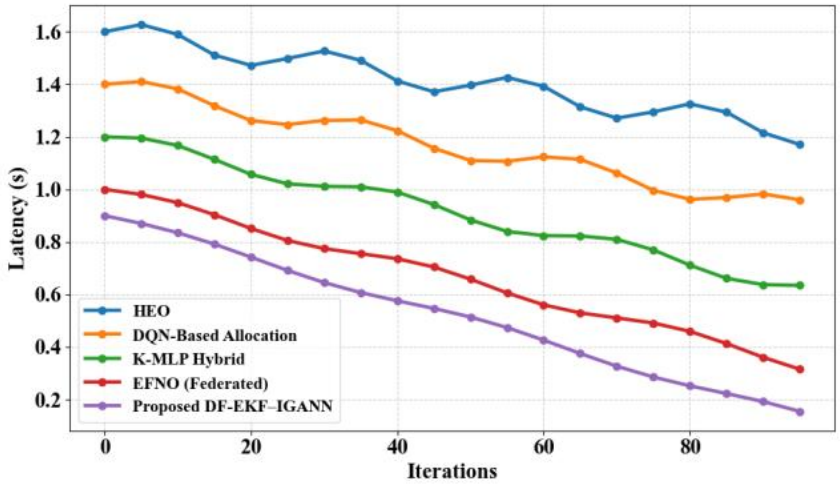


Fig. 5: Comparative Analysis of the Proposed Method in Terms of Latency

Conclusion

In summary, the study identifies the dimensions of the value of Interpretable Generalized Additive Neural Networks as the best way of optimizing data transformation and processing of tasks in edge computing. The system has shown a remarkable decrease in task latency and process efficiency enhancement through the combination of the advanced preprocessing methods such as DF-EKF and efficient efforts of resource allocation. The findings show that IGANN can not only contribute to an increase in task handling abilities but also allow managing edge resources more efficiently even in different conditions. Nevertheless, in spite of the encouraging results, the current strategy has some weaknesses, especially when it comes to scalability and the ability to support more diverse edge environments. Further research can focus on making the model applicable to manage progressively more complex edge networks and a broader workload scope, however, with a greater degree of optimization in energy consumption and system reliability. Moreover, it would have great benefits to research more advanced migration methods and take into account trade-offs in the energy consumption and processing capabilities of edge computing systems in real-time contexts.

Acknowledgement

None

Compliance with Ethical Standards

1. Potential Conflict of Interest Disclosure:

   The authors can testify that they do not have any potential conflict of interest.

2. Animal and Human Rights Declaration

   i. *Ethical Approval*

   All the pertinent national and/or institutional policies concerning use and care of animals were observed.

3. Informed Consent

   This type of analysis does not need any formal consent.

Statements On Data Availability

Given that no datasets were generated and analyzed in the paper, the concept of data sharing is not pertinent.

Funding

This study did not receive any specific grant either in terms of governmental, private or nonprofit funding organization.

Contribution of the Authors

Each author has made an equal contribution to the piece.

References

Liu, D., Zhang, Y., Jia, D., Zhang, Q., Zhao, X., & Rong, H. (2022). Toward secure distributed data storage with error locating in blockchain enabled edge computing. *Computer Standards & Interfaces*, *79*, 103560. https://doi.org/10.1016/j.csi.2021.103560

Li, C., Pan, H., Qian, H., Li, Y., Si, X., Li, K., & Zhang, B. (2024). Hierarchical sharding blockchain storage solution for edge computing. *Future Generation Computer Systems*, *161*, 162-173. https://doi.org/10.1016/j.future.2024.06.048

Shen, S., Wu, X., Sun, P., Zhou, H., Wu, Z., & Yu, S. (2023). Optimal privacy preservation strategies with signaling Q-learning for edge-computing-based IoT resource grant systems. *Expert Systems with Applications*, *225*, 120192. https://doi.org/10.1016/j.eswa.2023.120192

Tian, H., Xu, X., Lin, T., Cheng, Y., Qian, C., Ren, L., & Bilal, M. (2022). DIMA: Distributed cooperative microservice caching for Internet of Things in edge computing by deep reinforcement learning. *World Wide Web*, *25*(5), 1769-1792. https://doi.org/10.1007/s11280-021-00939-7

Poongodi, M., Bourouis, S., Ahmed, A. N., Vijayaragavan, M., Venkatesan, K. G. S., Alhakami, W., & Hamdi, M. (2022). A novel secured multi-access edge computing based vanet with neuro fuzzy systems based blockchain framework. *Computer Communications*, *192*, 48-56. https://doi.org/10.1016/j.comcom.2022.05.014

Lee, S., Peng, R., Wu, C., & Li, M. (2022). Programmable black phosphorus image sensor for broadband optoelectronic edge computing. *Nature communications*, *13*(1), 1-8. https://doi.org/10.1038/s41467-022-29171-1

Li, C., Zhang, Y., Gao, X., & Luo, Y. (2022). Energy-latency tradeoffs for edge caching and dynamic service migration based on DQN in mobile edge computing. *Journal of Parallel and Distributed Computing*, *166*, 15-31. https://doi.org/10.1016/j.jpdc.2022.03.001

Bréhon–Grataloup, L., Kacimi, R., & Beylot, A. L. (2022). Mobile edge computing for V2X architectures and applications: A survey. *Computer Networks*, *206*, 1-24. https://doi.org/10.1016/j.comnet.2022.108797

Rancea, A., Anghel, I., & Cioara, T. (2024). Edge computing in healthcare: Innovations, opportunities, and challenges. *Future internet*, *16*(9), 1-28. https://doi.org/10.3390/fi16090329

Song, S., Ma, S., Zhao, J., Yang, F., & Zhai, L. (2022). Cost-efficient multi-service task offloading scheduling for mobile edge computing. *Applied Intelligence*, *52*(4), 4028-4040. https://doi.org/10.1007/s10489-021-02549-2

Liu, T., Ni, S., Li, X., Zhu, Y., Kong, L., & Yang, Y. (2022). Deep reinforcement learning based approach for online service placement and computation resource allocation in edge computing. *IEEE Transactions on Mobile Computing*, *22*(7), 3870-3881. https://doi.org/10.1109/TMC.2022.3148254

Khanh, Q. V., Nguyen, V. H., Minh, Q. N., Van, A. D., Le Anh, N., & Chehri, A. (2023). An efficient edge computing management mechanism for sustainable smart cities. *Sustainable Computing: Informatics and Systems*, *38*, 100867. https://doi.org/10.1016/j.suscom.2023.100867

Williams, M., Emeteveke, I., Adeyeye, O. J., & Emehin, O. (2024). Enhancing data forensics through edge computing in IoT environments. *International Journal of Research Publication and Reviews*, *5*, 2970-2985. https://doi.org/10.55248/gengpi.5.1024.2903

Singh, A., Satapathy, S. C., Roy, A., & Gutub, A. (2022). Ai-based mobile edge computing for iot: Applications, challenges, and future scope. *Arabian Journal for Science and Engineering*, *47*(8), 9801-9831. https://doi.org/10.1007/s13369-021-06348-2

Chen, X., Bi, Y., Chen, X., Zhao, H., Cheng, N., Li, F., & Cheng, W. (2022). Dynamic service migration and request routing for microservice in multicell mobile-edge computing. *IEEE Internet of Things Journal*, *9*(15), 13126-13143. https://doi.org/10.1109/JIOT.2022.3140183

Sarah, A., Nencioni, G., & Khan, M. M. I. (2023). Resource allocation in multi-access edge computing for 5G-and-beyond networks. *Computer Networks*, *227*, 109720. https://doi.org/10.1016/j.comnet.2023.109720

Akindote, O. J., Adegbite, A. O., Dawodu, S. O., Omotosho, A., & Anyanwu, A. (2023). Innovation in data storage technologies: from cloud computing to edge computing. *Computer science & IT research journal*, *4*(3), 273-299. https://dx.doi.org/10.51594/csitrj.v4i3.661

Anees, T., Habib, Q., Al-Shamayleh, A. S., Khalil, W., Obaidat, M. A., & Akhunzada, A. (2023). The integration of WoT and edge computing: Issues and challenges. *Sustainability*, *15*(7), 1-27. https://doi.org/10.3390/su15075983

Song, F., Xing, H., Wang, X., Luo, S., Dai, P., Xiao, Z., & Zhao, B. (2022). Evolutionary multi-objective reinforcement learning based trajectory control and task offloading in UAV-assisted mobile edge computing. *IEEE Transactions on Mobile Computing*, *22*(12), 7387-7405. https://doi.org/10.1109/TMC.2022.3208457

Sun, G., Wang, Y., Sun, Z., Wu, Q., Kang, J., Niyato, D., & Leung, V. C. (2024). Multi-objective optimization for multi-UAV-assisted mobile edge computing. *IEEE Transactions on Mobile Computing*, *23*(12), 14803-14820. https://doi.org/10.1109/TMC.2024.3446819

BUPT MEC Migration. (2024). Edge Computing Dataset. https://github.com/BuptMecMigration/Edge-Computing-Dataset/tree/master/Data

Kulikova, M. V., & Kulikov, G. Y. (2024). Continuous–discrete derivative-free extended Kalman filter based on Euler–Maruyama and Itô–Taylor discretizations: Conventional and square-root implementations. *European Journal of Control*, *76*, 100960. https://doi.org/10.1016/j.ejcon.2024.100960

Kraus, M., Tschernutter, D., Weinzierl, S., & Zschech, P. (2024). Interpretable generalized additive neural networks. *European Journal of Operational Research*, *317*(2), 303-316. https://doi.org/10.1016/j.ejor.2023.06.032