Integrating Edge and Cloud Computing for Efficient Big Data Processing in IoT Environments: Enhancing Smart City Applications with Fog Computing

Karim Belkacem[1]

[1]University of El Oued, Faculty of Technology, Computer Science Department, Route de Touggourt, El Oued, Algeria

2024

Abstract

The seamless convergence of edge and cloud computing has emerged as a pivotal framework for managing large-scale data processing requirements in modern Internet of Things ecosystems. As smart city applications evolve, an increasing volume of real-time sensor data, high-resolution video streams, and contextual information must be analyzed with low latency to enable intelligent decision-making. At the same time, cloud infrastructures remain indispensable for long-term storage, historical analytics, and resource-intensive computational tasks. Fog computing offers an intermediate layer that augments the edge-to-cloud continuum and delivers context-sensitive intelligence at strategic points in the network. Balancing the location of services across the edge, fog, and cloud is critical for maximizing operational efficiency, improving scalability, and minimizing latency. A significant challenge lies in the design of dynamic task allocation and resource management strategies that are capable of adapting to fluctuating workloads and ensuring optimal quality of service. Recent developments in distributed systems engineering have demonstrated the potential of combining container-based virtualization with fine-grained orchestration techniques to yield robust and flexible deployment scenarios. Advanced data analytics frameworks capable of supporting machine learning, real-time streaming, and batch processing are increasingly integrated into this continuum. This work presents an extensive discussion of how edge and cloud architectures can be harmoniously blended with fog computing to enhance big data processing within IoT-enabled smart city environments. In particular, it addresses issues related to efficient system design, optimized resource utilization, and future research directions for large-scale deployments.

1 Introduction

The modern landscape of technological innovation is driven by the prolific growth of connected devices, sensors, and intelligent systems collectively forming the Internet of Things [1]. This widespread connectivity demands an infrastructure that can efficiently handle enormous data volumes and deliver real-time insights across diverse applications. In a typical smart city scenario, various subsystems such as traffic management, environmental monitoring, healthcare services, public safety, and energy management generate massive streams of data that must be analyzed to produce actionable intelligence. The ability to process these data streams in a timely and resource-efficient manner is integral to ensuring that urban services remain responsive, adaptive, and capable of meeting evolving demands. [2]

Edge computing has emerged as a vital paradigm by positioning computational and storage capabilities near the data source. This relocation reduces round-trip delays and alleviates bandwidth consumption, since critical analytical tasks can be performed locally before forwarding selective data to the cloud for more extensive processing. Such a strategy is particularly relevant in use cases where minimal latency is of high importance, or where bandwidth constraints could inhibit the transmission of large data sets. However, pushing substantial computational resources to the network edges alone does not suffice for holistic big data solutions [3]. The cloud remains necessary for large-scale analytics, advanced machine learning model training, archival of historical data, and collaboration across geographically distributed nodes. Hence, a balanced synergy between edge devices and remote cloud infrastructures is imperative for an effective big data processing pipeline in IoT ecosystems.

The advent of fog computing introduces an intermediate layer between the edge and the cloud. Instead of confining computation to one end of the network hierarchy, fog computing distributes resources along a spectrum of

possible deployment points [4]. A fog node, for instance, can be a local gateway or a micro data center strategically placed to handle near-edge computational tasks. This approach introduces new capabilities, including context-aware analytics and localized decision support, reducing the dependency on remote data centers when immediate responses are needed. By offloading specific functions from the cloud, the network can avoid bottlenecks associated with sending high-frequency sensor data over wide-area connections.

Despite the promise of this multi-layered approach, several challenges persist in designing an end-to-end architecture that leverages edge, fog, and cloud computing for big data applications [5]. One pressing issue lies in orchestrating service placement, where a continuum of possible nodes from local devices to distant data centers can host a given computation. Dynamic task allocation is complicated by variability in network conditions, unpredictable data arrival rates, and differences in hardware capacity. Another challenge emerges from the heterogeneity of IoT data sources, each with its own protocols, quality standards, and real-time requirements. A flexible solution must account for such diversity while ensuring reliability and interoperability across devices. [6]

These complexities are magnified when high-level analytics, such as machine learning or statistical modeling, become part of the real-time processing chain. Training data-intensive models often requires parallelization across substantial computing resources [7], which are typically found in the cloud. However, to enable rapid inference, edge nodes may host compressed or specialized versions of these models, allowing for local predictions and reducing transmission overhead. Coupling these tasks demands a careful division of labor between edge, fog, and cloud layers to avoid resource contention and yield predictable latency. [8]

Recent progress in virtualization and containerization technologies allows the packaging of services into modular components that can be deployed across heterogeneous platforms. These technologies provide a uniform environment to manage applications, facilitating load balancing and elastic scaling. Yet, the success of distributed resource management hinges on sophisticated scheduling algorithms that adapt to shifting workloads, energy usage patterns, and performance constraints. A well-designed scheduling algorithm also considers data locality by matching processing tasks with the location of relevant data, thereby reducing redundant transfers. [9]

Ensuring secure and private data handling further complicates the deployment. Different nodes in the network may have distinct trust levels, and sensitive information might not be suitable for processing outside controlled environments. Hence, encryption schemes and secure protocols must be employed, along with governance policies that define which data can be processed at which layers. This interplay between security considerations and performance objectives epitomizes the multifaceted nature of integrating edge, cloud, and fog computing for big data use cases in IoT. [10]

In what follows, a deeper exploration of these challenges and solutions is presented. A foundational discussion of edge, cloud, and fog computing sets the stage for understanding how each paradigm addresses the demands of big data processing. Advanced mathematical formulations guide the design of resource management and task scheduling, where intricate constraints and optimization goals define the operational parameters. Emphasis is placed on scalable analytics frameworks that can span distributed layers, and on how issues of security and privacy can be incorporated into these frameworks [11]. By harmonizing the best features of each layer within an IoT ecosystem, it becomes possible to realize robust and agile smart city services capable of turning raw sensor data into transformative insights.

2 Foundations of Edge, Cloud, and Fog Computing

The primary impetus behind edge computing lies in the need to reduce the latency associated with data transmission, particularly in real-time or near-real-time applications. By situating computational power closer to the data source, edge devices can preprocess sensor readings, perform localized analytics, and generate immediate responses. This approach diminishes the reliance on distant data centers for every aspect of data processing [12]. Instead, the most critical tasks are completed in situ, while non-urgent or large-scale computations are offloaded to larger infrastructures. The trade-off is that edge devices typically have limited computational and storage resources compared to dedicated servers. Consequently, these devices must employ lightweight models, resource-efficient algorithms, or on-the-fly data compression techniques to operate within their constraints.

In contrast, cloud computing provides virtually unlimited compute power and storage capacities at the expense of increased round-trip network latency [13]. The on-demand provisioning nature of cloud services allows them to handle extensive batch processing tasks, sophisticated machine learning model training, and large-scale data warehousing. Data centers leverage high-performance clusters and parallel processing architectures, enabling computationally expensive tasks to be completed efficiently. However, such efficiency at scale is contingent upon reliable and high-throughput network connections. This reliance can present critical bottlenecks in latency-sensitive scenarios or where network capacity is limited [14]. Moreover, transferring massive volumes of raw data to the cloud incurs bandwidth costs and could introduce delays detrimental to real-time applications.

Fog computing acts as a bridging layer between the edge and the cloud. Conceptually, it extends the cloud closer to the edge by introducing local nodes or micro data centers that operate at intermediate hierarchical levels.

These nodes possess more computational capabilities than typical edge devices, yet are located in the vicinity of data generation points [15], [16]. A fog node might reside at a cellular base station, a roadside unit for vehicular networks, or a local gateway within a building. By providing resource-rich but physically closer infrastructures, fog nodes can offload tasks that are too demanding for simple edge devices while eliminating the high latency associated with cloud-only solutions. The decision of what tasks to run on the fog layer is usually influenced by practical considerations such as data size, time sensitivity, and the availability of local storage and processing units.

An important point in this continuum is the dynamic allocation of computational tasks [17]. One can conceptualize a function $f(x)$ representing a specific data processing task x. A variety of constraints might dictate whether $f(x)$ should be executed at the edge, in a fog node, or in the cloud.

Let c_i be the computational cost of executing $f(x)$ on node i, and let d_i be the delay associated with transferring data from its source to node i. A high-level objective could be to minimize total latency:

$$L = d_i + t_i,$$

where t_i denotes the execution time on node i [18]. If we extend this to a set of tasks X, each with its own resource demands, we obtain a multi-objective optimization problem. The problem might incorporate additional terms such as energy consumption e_i, financial cost p_i, or reliability metrics r_i, forming a composite cost function:

$$C(i) = \alpha_1 d_i + \alpha_2 t_i + \alpha_3 e_i + \alpha_4 p_i - \alpha_5 r_i.$$

The decision of which node i to assign to each task $x \in X$ becomes a matter of solving an assignment or scheduling problem subject to the constraints of available computing capacity, network bandwidth, and QoS expectations.

These foundational theories underpin how edge, cloud, and fog computing converge [19]. Successful deployment demands more than just architectural alignment; it necessitates a systemic approach to data lifecycle management. Data may originate in streaming form from IoT devices and require immediate transformation or filtering at the edge to remove irrelevant details. Important data subsets or aggregated features then propagate to fog nodes for additional processing, possibly involving batch or stream-based machine learning techniques. Long-term storage or advanced analytics occur in the cloud, where historical data sets are combined with data from multiple fog layers to derive global insights [20]. This cyclical process continuously refines edge models, which are updated based on cloud-derived inferences and subsequently redeployed closer to the source, perpetuating a closed-loop intelligence system.

In certain large-scale IoT deployments, the interplay among these layers becomes exceedingly complex. Consider a scenario in which multiple edge nodes, each with distinct specifications, interact with a fog layer composed of distributed micro data centers. Different subsets of tasks can be dynamically migrated between these nodes based on congestion, energy constraints, or time-varying user demand [21]. When the primary objective is low latency, heavier workloads might be distributed across multiple fog nodes to reduce per-node congestion. Conversely, when cost savings or energy efficiency dominate, tasks are consolidated to whichever nodes offer cheaper or more sustainable energy. Managing these multi-factor trade-offs in real time requires sophisticated models that track system state and reallocate tasks in response to fluctuations.

Another element of foundational importance is the communication protocol stack that supports edge, fog, and cloud integration [22]. High-performance, low-latency protocols might be required for edge communication, while more conventional or higher-level protocols are sufficient between fog nodes and the cloud. The entire system benefits from software-defined networking techniques that dynamically adjust routing paths, shaping network traffic to accommodate bandwidth-intensive tasks in an efficient manner. Cross-layer optimization becomes relevant, where decisions at the transport layer (e.g., congestion control) inform resource allocation at the application layer. Such interaction underscores the holistic nature of designing an integrated edge, fog, and cloud ecosystem. [23]

Ultimately, successful foundations for this computing paradigm hinge on the interplay of architectural layering, well-formulated optimization problems, appropriate protocols, and adaptive runtime environments. As this foundation supports advanced IoT applications, it becomes essential to integrate robust analytics pipelines. The subsequent sections explore how big data analytics frameworks scale across this continuum and how mathematical models can guide resource management to meet latency, energy, and reliability requirements.

3 Scalable Big Data Analytics in IoT

In large-scale IoT deployments, sensor data can be high-dimensional and originate from an immense number of devices [24], [25]. These data streams arrive continuously, often in unstructured or semi-structured formats, demanding real-time or near-real-time analysis. Traditional relational database systems, while effective for well-defined transactions, are less suitable for handling the velocity and variety characteristics of IoT data. Consequently, modern big data frameworks have emerged that accommodate distributed processing and horizontally scalable storage, providing a key foundation for analytics workflows across the edge-fog-cloud continuum.

A typical approach to scalable IoT analytics is to partition data streams in real time, applying a streaming engine that can ingest and process records with minimal latency [26], [27]. When edge devices undertake preliminary filtering or feature extraction, they essentially reduce the dimensionality and volume of data before forwarding it to fog or cloud layers. Fog nodes may apply more advanced analytics, such as windowed stream aggregations or anomaly detection models, to glean localized insights. The aggregated or intermediate results are then propagated to the cloud for global analyses, pattern recognition across different regions, or deeper historical context. This partitioned pipeline avoids redundant processing and ensures that each layer focuses on tasks well suited to its computational resources and latency requirements. [28]

The synergy between distributed data processing engines and containerization technologies has paved the way for flexible deployment strategies. Containers encapsulate analytics workloads together with their dependencies, streamlining the movement of analytics tasks among edge devices, fog nodes, and cloud clusters. This portability can be described by a function $M(d)$, representing a containerized microservice handling data type d. Suppose we have an array of microservices $M_1(d), M_2(d), \ldots, M_k(d)$, each specialized for a different analytics procedure, such as streaming aggregation, advanced classification, dimensionality reduction, or encryption [29].

Under dynamic conditions, an orchestration layer might relocate specific microservices to whichever physical or virtual node is most effective given the current data load or resource availability. In many cases, a load-balancing approach is implemented by measuring the real-time metrics of each node, such as CPU load, memory usage, and bandwidth capacity, and then redirecting tasks to maintain balanced utilization.

An intricate challenge in scalable IoT analytics is the coordination of event-time and processing-time semantics [30]. Some applications require that data be analyzed in the chronological order of their generation, even if the arrival times are skewed due to network latency. This synchronization necessitates buffering strategies and watermarking techniques that become increasingly complex in a distributed environment. If edge or fog nodes timestamp data based on local clocks, and the cloud merges streams from multiple locations, the system must cope with potential clock drift or out-of-order arrivals. Accurate analyses might mandate the periodic recalculation of partial results once delayed data arrives, implying that the underlying analytics framework must accommodate incremental and iterative computations. [31]

Machine learning plays a central role in extracting actionable insights from IoT data. Training large-scale models, such as deep neural networks, often occurs in cloud data centers equipped with specialized hardware like graphics processing units. However, these models can be pruned or compressed and then deployed at the edge for inference tasks. This workflow shortens the response time for critical decisions and conserves bandwidth [32].

The cost function for training a model could be formulated as:

$$\Omega(W) = L(W) + \lambda \|W\|^2,$$

where W represents the model parameters, $L(W)$ is the empirical loss over the training data, λ is a regularization coefficient, and $\|W\|^2$ is the squared norm. After iterative optimization in the cloud, a subset or approximation of the learned parameters might be distributed to fog or edge nodes for local predictions. Real-world data can subsequently be fed back into an online learning mechanism, reinforcing or adjusting the model to account for domain shifts or concept drifts.

Data redundancy and replication strategies are critical for fault tolerance in distributed analytics systems [33]. In an IoT context, sensor readings may be crucial for real-time applications such as public safety or healthcare monitoring. To mitigate the risk of data loss, systems frequently maintain multiple copies of data segments across nodes. A replication factor r signifies how many distinct nodes store the same data segment. This redundancy introduces overhead in both storage utilization and network traffic [34]. However, it ensures data availability despite node failures or network partitions.

Determining the optimal replication factor and placement policy can be viewed as a constrained optimization problem, balancing the benefit of high availability against increased resource costs. Let $A(r)$ measure availability as a function of r, and let $B(r)$ measure resource overhead. The system aims to maximize:

$$A(r) - \mu B(r),$$

where μ is a weight factor [35]. In a fog environment, replication policies might prioritize local copies for time-critical data, while the cloud stores complete archives for reliability and historical analysis.

The interplay between stream processing, distributed storage, and machine learning exemplifies the core of scalable IoT analytics. Achieving smooth integration across edge, fog, and cloud layers requires architectural uniformity at the software level alongside robust scheduling, load balancing, and fault tolerance. Moreover, different applications impose unique workloads and performance criteria [36]. A public transportation monitoring system might generate moderately sized but continuous data streams focusing on location and occupancy, whereas a citywide network of video surveillance cameras produces high-bandwidth streams requiring efficient compression and distributed analytics for object detection. Each scenario necessitates a tailored pipeline reflecting the constraints and capabilities of the respective edge, fog, and cloud elements.

This diverse range of use cases underscores the necessity for flexible big data frameworks that adapt to heterogeneous IoT environments. The next logical step entails embedding rigorous mathematical modeling within the resource management strategies that govern these frameworks [37]. Only by codifying the operational constraints and performance objectives can a distributed system autonomously allocate resources across edge, fog, and cloud layers in a manner that ensures efficient, reliable, and scalable big data analytics. This mathematical perspective will also address uncertainties inherent in IoT workloads, such as variable data rates or unpredictable node failures, and provide a basis for robust decision-making under dynamic conditions. The following sections detail how advanced mathematical models and optimization techniques can be leveraged to meet these objectives.

4 Mathematical Formulations for Resource Management

Systematic allocation of computing, storage, and networking resources across edge, fog, and cloud layers can be captured by mathematically rigorous approaches [38]. A core element in such models is the formal definition of system states, decision variables, and objective functions. The problem of deciding where and when to execute tasks can be recast as a multi-stage decision process, often formulated using linear, mixed-integer, or non-linear programming frameworks.

Let the system be represented by a set N of nodes. Each node $i \in N$ has certain capacities: C_i for compute, S_i for storage, and B_i for network bandwidth [39]. There is a set T of tasks to be scheduled, where each task $t \in T$ requires a certain compute demand c_t, storage demand s_t, and generates traffic at a rate b_t. A binary decision variable $x_{t,i}$ can indicate whether task t is assigned to node i. Constraints arise from capacity limits:

$$\sum_{t \in T} c_t \, x_{t,i} \leq C_i, \quad \sum_{t \in T} s_t \, x_{t,i} \leq S_i.$$

These constraints reflect the maximum computational and storage resources that node i can allocate. Bandwidth constraints might be more complex, especially if each node is connected to a subset of other nodes by links with limited capacities [40]. Let $L_{i,j}$ represent the capacity of the link from node i to node j. If a fraction $f_{t,i,j}$ of the data generated by task t on node i needs to be sent to node j, then

$$\sum_{t \in T} f_{t,i,j} \, b_t \leq L_{i,j}.$$

Objective functions typically involve minimizing total execution cost, latency, or a weighted combination of such factors. For instance, if δ_i is the average latency to process a task at node i, and $\varepsilon_{i,j}$ is the latency to transmit data from node i to node j, the total latency for a task t assigned to node i with partial data flows to j might be...

$$L_t = \delta_i + \epsilon_{i,j} \, \mathbb{I}(f_{t,i,j} > 0),$$

where $\mathbb{I}(\cdot)$ is an indicator function. Minimizing the sum of latencies for all tasks requires integer or mixed-integer programming. However, such a formulation can be computationally prohibitive for large-scale IoT systems, leading to explorations of heuristic or metaheuristic solutions.

Stochastic modeling offers another powerful lens, especially when task arrivals and data rates are uncertain [41]. One can employ continuous-time Markov chains or semi-Markov decision processes to represent the arrival of tasks as a stochastic process with transition rates dependent on traffic intensities or external events. A state s in a Markov chain might encode which tasks are being processed at which nodes, along with remaining capacities and queue lengths. The system transitions to a new state s' upon arrival or completion of tasks. This approach yields a long-run average cost or reward measure, which can then be optimized [42]. Analytical solutions might only be tractable for simplified cases, necessitating approximate dynamic programming or reinforcement learning methods in practice [43].

A more granular perspective examines scheduling mechanisms at individual fog nodes, where tasks arrive according to a Poisson process with rate λ. If the service rate of a node is μ, a basic M/M/1 queueing model might be employed to approximate average waiting times and queue lengths. The average waiting time W in an M/M/1 queue is given by [44]:

$$W = \frac{1}{\mu - \lambda}, \quad \text{for } \lambda < \mu.$$

$$W = \frac{\rho}{\mu - \lambda}, \quad \rho = \frac{\lambda}{\mu}.$$

In edge-fog-cloud systems, queueing models can be extended into networks with multiple service stations, each with potentially different rates. These open queueing networks can be analyzed to predict how tasks flow through

the system, how congestion develops, and where bottlenecks emerge. Although these basic models provide insights, real-world IoT workloads can display burstiness and time-dependent arrival rates, warranting more advanced modeling techniques that incorporate general or hyperexponential service time distributions. Nonetheless, queueing theory remains a core tool for understanding fundamental performance trade-offs in latency-sensitive applications. [45]

Game-theoretic formulations can also be relevant for decentralized resource allocation. Different IoT devices or fog nodes might aim to optimize their own performance metrics, such as minimizing local energy consumption or latency. If these agents share limited network resources, they engage in a competition for those resources, leading to a game. Defining payoffs for each player based on resource usage and performance metrics, one can explore equilibrium strategies [46]. In a simple example, each node i tries to decide how many tasks to offload to a neighboring node j to reduce local congestion, balancing the cost of transmitting tasks over the network. A Nash equilibrium emerges when no individual node can improve its utility by unilaterally changing its offloading decision. Finding or approximating these equilibria can guide distributed solutions that scale without centralized coordination.

Beyond these conventional frameworks, advanced learning-based methods are being integrated into resource management [47]. Reinforcement learning agents can observe system states, such as current loads, queue lengths, and network latencies, and choose actions like migrating a subset of tasks to another fog node. Over time, the agent refines its policy to minimize a global cost function that might combine latency, energy consumption, and monetary expenses. Such data-driven methods are appealing in highly dynamic environments, where explicit analytical models may not capture all intricacies. However, they rely on extensive training data and careful tuning of hyperparameters to achieve stable and near-optimal performance. [48]

In practice, resource management solutions may combine multiple methods. A queueing model might provide an initial estimate of system performance, which serves as a baseline for more complex dynamic or game-theoretic optimization methods. Meanwhile, heuristics derived from domain-specific knowledge can handle real-time constraints. For instance, an IoT gateway might implement a rule-based approach to locally filter data during network congestion, while a global controller periodically adjusts task placement using a more comprehensive optimization procedure [49]. This layered approach takes advantage of theoretical rigor where feasible, yet remains adaptable to on-the-fly changes in system demands.

Mathematical modeling forms the underpinning of efficient resource management for integrated edge, fog, and cloud systems. By defining precise optimization goals and constraints, one can systematically explore different trade-offs, such as latency versus energy or local responsiveness versus global efficiency. These models also facilitate the integration of reliability and security constraints, ensuring that resource allocation meets not only performance requirements but also robustness criteria [50]. The next section delves into specific optimization schemes focused on minimizing latency and energy consumption, providing concrete examples of how these mathematical frameworks translate into improved system performance in real-world scenarios.

5 Optimization for Latency and Energy Efficiency

IoT systems with edge-fog-cloud integration frequently prioritize minimizing the overall response time for data processing and service delivery. Concurrently, there is a strong impetus to reduce energy consumption, particularly in resource-constrained edge devices and large-scale fog infrastructures. Balancing these two objectives often involves a compromise between retaining tasks at the edge for immediate processing and offloading them to resource-abundant nodes to alleviate local power draw [51]. Mathematical optimization techniques provide a structured means to achieve this balance.

A typical formulation to reduce latency focuses on the sum of processing delays and communication delays. Let T_t denote the total processing time for task t across whatever sequence of nodes it traverses. Suppose T_t is composed of the local service delay at node i, D_i, plus any queueing or routing latency E_i,j between nodes i and j. If x_t,i remains the decision variable indicating that task t is processed by node i, and y_t,i,j indicates data transfer along link (i, j), the latency minimization can be expressed as

$$\min \sum_{t \in T} \Big(\sum_{i \in N} D_i \, x_{t,i} + \sum_{(i,j) \in E} E_{i,j} \, y_{t,i,j} \Big),$$

subject to capacity constraints ensuring that D_i does not exceed the node's resource limitations and that E_i,j does not exceed link bandwidth. One might incorporate S_i, the energy consumption for running tasks on node i, as an additional term in the objective [52]. Let P_i be the power usage function of node i, which might be modeled as P_i = P_idle,i + k_i U_i, where P_idle,i is the baseline power consumption, k_i is a scaling factor, and U_i is the utilization ratio. A combined objective becomes

$$\min\Big(\beta \sum_{t\in T} T_t + (1-\beta)\sum_{i\in N} P_i\Big),$$

where B is a weighting parameter determining the relative importance of latency versus energy. By tuning B, system designers can emphasize reduced response times or lower power consumption. This multi-objective function can be tackled via methods like weighted-sum approaches, epsilon-constraint techniques, or evolutionary algorithms designed for multi-objective optimization. [53]

Energy consumption in fog nodes often presents different characteristics than in edge devices or cloud data centers. Fog infrastructure typically supports moderate compute capacity with constraints on heat dissipation and power availability. In contrast, edge devices may be battery-operated or reliant on solar power, making power efficiency paramount. The cloud, on the other hand, can draw from large-scale electricity supplies but aims to optimize energy usage for cost and sustainability [54]. A hierarchical optimization can be employed, splitting the problem into subproblems at each layer with local objectives, and then reconciling these subproblems through global coordination. For example, each fog node might minimize its own power usage subject to local latency constraints, while a higher-level controller ensures that system-wide latency targets are met by distributing workloads among the fog layer and the cloud.

Sophisticated scheduling algorithms are necessary for near real-time operation, especially when tasks arrive unpredictably. A typical heuristic approach might sort tasks by urgency and size, then assign them to nodes in order of increasing available capacity [55]. Another approach uses a dynamic feedback mechanism, measuring queue lengths and estimated completion times at each node to shift tasks proactively. These heuristics can be informed by the underlying mathematical formulations. For instance, a Lagrangian relaxation technique might produce dual variables indicating the marginal cost of adding an additional unit of load to a node [56]. A scheduling policy could then offload tasks from nodes with high marginal costs to nodes with lower ones, approximating the global optimum.

One might consider scenarios where tasks can be partially processed at the edge, then handed off to a fog node for completion. This pipelined processing helps reduce local battery drain without incurring full network latency. A typical pipeline might break a task into multiple stages [57]. The local stage at the edge device processes an initial chunk of data or performs feature extraction, with the remainder transmitted to the fog node. Fog nodes further reduce or transform the data before sending a fraction of it to the cloud for advanced analytics. Such a pipeline can be captured by multi-stage optimization, where each stage i is associated with decision variables $x_{t,i}$ and constraints capturing the resource usage at that stage.

In addition to deterministic optimization, uncertainties in workload and energy availability, particularly for battery-operated devices, motivate robust optimization. A robust model might incorporate worst-case bounds on arrival rates or battery levels, guaranteeing that constraints are satisfied under adverse conditions [58]. Alternatively, chance-constrained optimization allows for a small probability of violation, balancing performance with the need to hedge against variability. Let W_t be a random variable representing the required workload for task t. A chance constraint might be

$$\mathbb{P}\left(\sum_{t\in T} W_t\, x_{t,i} \le C_i\right) \ge 1-\alpha,$$

where α is a small tolerance for capacity overload [59]. Solving such problems often necessitates approximations or bounding techniques, especially if the distribution of W_t is complex.

Another element is the synergy between workload scheduling and adaptive control of node states. To reduce energy usage, fog or edge nodes can enter low-power or sleep modes when idle. However, reactivating a node introduces a startup delay, which can affect latency [60]. An optimization framework might treat node states as decision variables, where z_i indicates whether node i is in active or standby mode. The system then weighs the energy savings from shutting down underutilized nodes against the potential latency penalty when those nodes need to be reactivated. The resulting problem might be formulated as

$$\min \sum_{t\in T} T_t\, x_{t,i} + \gamma \sum_{i\in N}\left(P_{\text{standby},i}\,(1-z_i) + P_{\text{active},i}\, z_i\right),$$

with constraints tying the availability of node i to z_i for resource scheduling. [61]

Ultimately, these optimization models form the backbone of design strategies aimed at achieving low-latency, energy-efficient operation in large-scale IoT systems. The integration of advanced techniques, including integer programming, robust and stochastic optimization, and multi-stage formulations, provides a robust methodology to handle the vast complexity inherent in distributed environments. While exact solutions can be challenging for large problem instances, approximation schemes, heuristics, and real-time adaptive algorithms can effectively guide resource allocation toward near-optimal outcomes. As the subsequent section highlights, security and privacy

considerations also feature prominently in these system designs, further complicating optimization frameworks by introducing new constraints and objectives related to data protection and trust management. [62]

6 Security and Privacy Considerations

In an IoT ecosystem encompassing edge, fog, and cloud nodes, security and privacy concerns must be addressed at all layers. Sensitive data originating from connected devices may traverse multiple networks and intermediate processing stages, each with its own vulnerabilities. These vulnerabilities are amplified by the heterogeneity of hardware and software platforms, as well as the distributed nature of resource allocation. Consequently, the risk of unauthorized data access or tampering escalates. [63], [64]

A pivotal challenge in such environments is ensuring end-to-end data confidentiality and integrity. Commonly, encryption schemes are implemented both in transit and at rest. However, the overhead of encrypting and decrypting data can negatively impact latency-sensitive applications, particularly on resource-constrained edge devices. To model this overhead, one might treat encryption and decryption as additional tasks that must be scheduled, each with a compute requirement proportional to the data volume [65]. Let E_t and D_t represent the computation times for encrypting and decrypting data associated with task t. The system must decide whether to encrypt at the edge, the fog, or the cloud to balance security with performance. This decision can be captured by introducing binary variables e_t,i or d_t,i indicating that encryption or decryption for task t occurs at node i. The total latency for task t might then be updated to include these encryption steps:

$$T\_t = \sum \_i \in N\big(D_i\, x_{t,i} + E_t\, e_{t,i} + D_t\, d_{t,i}\big) + \sum_{(i,j)\in E} E_{i,j}\, y_{t,i,j}.$$

A robust design must also safeguard against data leaks and unauthorized access. Fog nodes or edge devices lacking physical protection may be vulnerable to theft or tampering, which could expose encryption keys or stored data [66]. One approach involves distributing key management responsibilities so that no single node holds all the information needed to decrypt critical data. Secret sharing schemes can be employed, dividing a key into multiple shares stored across distinct fog nodes or even partially in the cloud. These methods complicate the mathematics of resource scheduling, as acquiring necessary key shares becomes an additional workflow step.

Access control policies further complicate matters [67]. Some tasks might require a high-security environment, restricting which nodes can process them. If we define a security level S_t for each task t, we can impose constraints that x_t,i = 0 unless the security level of node i meets or exceeds S_t. Let $\sigma(i)$ be the security clearance of node i. The constraint

$$x_{t,i} \leq \mathbb{I}(\sigma(i) \geq S_t)$$

enforces that task t can only be assigned to nodes with adequate clearance [68]. This can reduce the available nodes for high-security tasks, potentially increasing latency or resource usage. Nonetheless, it is crucial to ensure that sensitive tasks do not end up on untrusted nodes.

Privacy considerations arise when certain data must not be processed outside a specific geographical region or beyond the domain of the data owner. Geo-fencing policies can be introduced to confine data to nodes physically located within approved areas [69]. If $\phi(i)$ denotes the geographical region of node i, we might require $\phi(i) \in \mathcal{R}_t$ for each task t, where $\mathcal{R}_t$ is the set of regions approved for that task's data. Such constraints further restrict possible schedules, making resource allocation even more intricate.

A crucial dimension of secure deployment is intrusion detection and anomaly detection at multiple layers. Machine learning models that analyze logs and traffic patterns can be hosted at either the fog or the cloud layer, depending on where data is most effectively aggregated. A dynamic approach to host anomaly detection might treat the anomaly detection service as a special task that periodically collects network traffic data [70]. This service must operate with minimal latency to promptly detect malicious activities. The system can incorporate the cost of real-time intrusion detection, such that tasks requiring heightened security are paired with anomaly detection services in the same fog node or cluster for immediate cross-checking. Let q_t,i be a variable indicating that an anomaly detection service is co-located with task t on node i. The resource constraints expand to account for the overhead of running these security services alongside normal tasks.

Multi-layer trust management mechanisms also influence how data and tasks move through the system [71]. Different trust policies may apply to edge devices, fog nodes, or cloud platforms. A trust level τ_i for node i can be modeled, and tasks might require a trust threshold η_t. Only nodes satisfying $\tau_i \geq \eta_t$ may process task t.

This requirement is similar to security clearance constraints, but it can also be dynamic: trust levels could evolve over time based on observed behavior or remote attestation processes [72]. If node i becomes compromised, τ_i might be reduced, and tasks reallocated accordingly.

These security and privacy requirements must be integrated with the latency and energy optimization frameworks discussed earlier. In many cases, there is a fundamental tension: more robust security procedures, such as comprehensive encryption or frequent intrusion checks, typically consume additional computing and networking resources, prolonging task turnaround times. Conversely, seeking to minimize latency might lead to shortcuts in security procedures or the placement of tasks on less secure nodes [73]. Mathematical formulations that incorporate these trade-offs often introduce penalty terms for security non-compliance or privacy violations. One might define a penalty function P(t, i) that represents the risk or potential damage of running a task t on node i. Minimizing the system-wide sum of latencies plus the risk penalty forms a multi-objective problem:

$$\min\Big(\beta \sum_{t \in T} T_t + (1-\beta) \sum_{t \in T} \sum_{i \in N} P(t,i)\, x_{t,i}\Big).$$

The nature of P(t, i) can be application-specific, potentially involving factors like node trust level, data sensitivity, and threat intelligence reports [74], [75]. Balancing these security considerations with performance and energy objectives requires a holistic optimization approach, potentially employing advanced solution techniques like robust multi-objective programming, iterative approaches that refine partial solutions, or distributed consensus methods.

In conclusion, security and privacy considerations cannot be addressed in isolation; they intertwine with latency, energy efficiency, and reliability across the entire edge-fog-cloud continuum. By encoding these requirements into unified mathematical or algorithmic frameworks, system designers can systematically explore and evaluate trade-offs. These integrated solutions form the basis for resilient, secure, and high-performing IoT infrastructures that serve as the backbone of advanced smart city applications, where data integrity and confidentiality are as important as operational effectiveness. [76]

7 Conclusion

The transformation of smart city applications through the integration of edge, cloud, and fog computing relies on orchestrating a wide spectrum of technologies, computational models, and system-level optimizations to handle the exponential growth of IoT data. By placing processing resources closer to data sources at the edge, near-edge gateways in the fog layer, and maintaining large-scale analytics engines in the cloud, it becomes possible to achieve a delicate balance between real-time responsiveness and massive-scale data processing. The intermediate fog layer offers a localized yet resource-rich environment, reducing latency while offloading significant burdens from the edge. Simultaneously, the cloud remains indispensable for globally aggregating data, conducting advanced data mining, and storing large historical repositories. [77]

A comprehensive methodology involves addressing system design from both architectural and algorithmic perspectives. Architecturally, multi-layer solutions distribute computing tasks based on workload characteristics, capabilities of individual nodes, and network topologies. Algorithmically, sophisticated mathematical models facilitate resource management by expressing constraints, objectives, and performance metrics. Whether framed as linear or non-linear programs, stochastic decision processes, or multi-objective optimizations, these models highlight fundamental trade-offs among latency, energy consumption, security, and robustness [78]. Solutions can take the form of approximation algorithms, heuristics, or dynamic allocation schemes driven by real-time metrics like node utilization and network throughput.

Resource allocation challenges are heightened by the inherent heterogeneity of IoT devices, each with diverse requirements in terms of processing power, memory, and battery life. Moreover, high-level analytics workflows and machine learning pipelines demand flexible scaling, often pushing the training of large models to cloud data centers while retaining inferential tasks near the data source. When tasks are partitioned across edge, fog, and cloud nodes, data life cycles intertwine with distributed storage, queueing networks, and partial offloading mechanisms [79]. Balancing these operational details requires careful coordination of protocols and intelligent scheduling algorithms that adapt in response to changing workloads and dynamic network conditions.

Security and privacy introduce a layer of complexity that intersects with all other considerations. Encryption, trust management, intrusion detection, and geo-fencing constraints become integral parts of the optimization framework. The overhead associated with these controls may affect system latency and energy usage, compelling a unified approach that accounts for both performance and risk mitigation [80]. As IoT data grows ever more sensitive and regulations around data privacy become more stringent, secure architectures assume critical importance in real-world implementations.

The discussion presented underscores the potential of a carefully orchestrated edge-cloud-fog continuum to elevate smart city services from passive data collection to active, real-time decision-making. Substantial research remains in refining robust solution frameworks, exploring distributed learning paradigms, and ensuring fault tolerance under unpredictable conditions. Although a variety of algorithms and models have been proposed, ongoing developments in hardware capabilities, virtualization technologies, and networking protocols will continue to shift the boundaries of what is achievable [81]. Future pathways may see increased adoption of machine learning for

predictive resource management, deeper integration of blockchain or other distributed ledgers for trust, and more advanced sensor fusion techniques to handle data of ever-growing variety and scale.

In sum, the synergy between edge computing, fog nodes, and cloud infrastructures is poised to meet the demands of modern IoT environments, transforming raw sensor streams into actionable intelligence. By leveraging robust architectures, sophisticated resource allocation, and end-to-end security measures, this tri-layered paradigm empowers smart city applications to achieve high performance, reliability, and adaptability. Continued exploration of mathematical modeling and optimization will be instrumental in guiding these deployments toward even greater levels of efficiency and resilience, ensuring that the promise of integrated computing for big data processing fulfills the evolving needs of diverse urban landscapes. [82]

References

[1] Q. Wang and D. Shen, "Computational medicine: A cybernetic eye for rare disease," *Nature biomedical engineering*, vol. 1, no. 2, pp. 0032–, Feb. 10, 2017. DOI: 10.1038/s41551-017-0032.

[2] L. Rompis, "A logic circuit simulation for finding identical or redundant files using counter and register," *Advances in Image and Video Processing*, vol. 6, no. 3, pp. 26–26, Jun. 30, 2018. DOI: 10.14738/aivp.63.4689.

[3] J. Ahmed, A. Malik, M. Ilyas, and J. S. Alowibdi, "Instance launch-time analysis of openstack virtualization technologies with control plane network errors," *Computing*, vol. 101, no. 8, pp. 989–1014, May 28, 2018. DOI: 10.1007/s00607-018-0626-5.

[4] S. T. Milan, L. Rajabion, A. M. Darwesh, M. Hosseinzadeh, and N. J. Navimipour, "Priority-based task scheduling method over cloudlet using a swarm intelligence algorithm," *Cluster Computing*, vol. 23, no. 2, pp. 663–671, Jun. 12, 2019. DOI: 10.1007/s10586-019-02951-z.

[5] P. Singh, Y. K. Dwivedi, K. S. Kahlon, R. S. Sawhney, A. A. Alalwan, and N. P. Rana, "Smart monitoring and controlling of government policies using social media and cloud computing," *Information Systems Frontiers*, vol. 22, no. 2, pp. 315–337, Apr. 16, 2019. DOI: 10.1007/s10796-019-09916-y.

[6] B. Foreman, I. A. Lissak, N. Kamireddi, D. Moberg, and E. Rosenthal, "Challenges and opportunities in multimodal monitoring and data analytics in traumatic brain injury," *Current neurology and neuroscience reports*, vol. 21, no. 3, pp. 1–9, Feb. 2, 2021. DOI: 10.1007/s11910-021-01098-y.

[7] R. Avula, "Strategies for minimizing delays and enhancing workflow efficiency by managing data dependencies in healthcare pipelines," *Eigenpub Review of Science and Technology*, vol. 4, no. 1, pp. 38–57, 2020.

[8] N. Gupta, S. Gupta, Khosravy, *et al.*, "Economic iot strategy: The future technology for health monitoring and diagnostic of agriculture vehicles," *Journal of Intelligent Manufacturing*, vol. 32, no. 4, pp. 1117–1128, Jul. 19, 2020. DOI: 10.1007/s10845-020-01610-0.

[9] X. Chen, H. H. Wang, and B. Tian, "Multidimensional agro-economic model with soft-iot framework," *Soft Computing*, vol. 24, no. 16, pp. 12 187–12 196, Jan. 4, 2020. DOI: 10.1007/s00500-019-04657-1.

[10] R. R. Thangudu, P. A. Rudnick, M. Holck, *et al.*, "Abstract lb-242: Proteomic data commons: A resource for proteogenomic analysis," *Cancer Research*, vol. 80, no. 16$_S$*upplement*, LB–242-LB-242, Aug. 15, 2020. DOI: 10.1158/1538-7445.am2020-lb-242.

[11] A. B. Nellippallil, Z. Ming, J. K. Allen, and F. Mistree, "Cloud-based materials and product realization—fostering icme via industry 4.0," *Integrating Materials and Manufacturing Innovation*, vol. 8, no. 2, pp. 107–121, Jun. 6, 2019. DOI: 10.1007/s40192-019-00139-2.

[12] S. Murphy, V. Castro, and K. Mandl, "Grappling with the future use of big data for translational medicine and clinical care," *Yearbook of Medical Informatics*, vol. 26, no. 1, pp. 96–102, Aug. 19, 2017. DOI: 10.1055/s-0037-1606487.

[13] B. S. Rawal, V. Vijayakumar, G. Manogaran, R. Varatharajan, and N. Chilamkurti, "Secure disintegration protocol for privacy preserving cloud storage," *Wireless Personal Communications*, vol. 103, no. 2, pp. 1161–1177, Jan. 19, 2018. DOI: 10.1007/s11277-018-5284-6.

[14] E. F. Z. Santana, A. P. Chaves, M. A. Gerosa, F. Kon, and D. Milojicic, "Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture," *ACM Computing Surveys*, vol. 50, no. 6, pp. 78–37, Nov. 22, 2017. DOI: 10.1145/3124391.

[15] X. Xie, Y. Fang, Z. Jian, Y. Lu, T. Li, and G. Wang, "Blockchain-driven anomaly detection framework on edge intelligence," *CCF Transactions on Networking*, vol. 3, no. 3, pp. 171–192, Nov. 24, 2020. DOI: 10.1007/s42045-020-00044-9.

[16] M. Kansara, "A comparative analysis of security algorithms and mechanisms for protecting data, applications, and services during cloud migration," *International Journal of Information and Cybersecurity*, vol. 6, no. 1, pp. 164–197, 2022.

[17] L. Yang, J. Li, N. Elisa, T. Prickett, and F. Chao, "Towards big data governance in cybersecurity," *Data-Enabled Discovery and Applications*, vol. 3, no. 1, pp. 1–12, Dec. 2, 2019. DOI: 10.1007/s41688-019-0034-9.

[18] B. Dorneanu, S. Zhang, H. Ruan, *et al.*, "Big data and machine learning: A roadmap towards smart plants," *Frontiers of Engineering Management*, vol. 9, no. 4, pp. 623–639, Aug. 31, 2022. DOI: 10.1007/s42524-022-0218-0.

[19] S. A. P. Kumar, R. Madhumathi, P. R. Chelliah, L. Tao, and S. Wang, "A novel digital twin-centric approach for driver intention prediction and traffic congestion avoidance," *Journal of Reliable Intelligent Environments*, vol. 4, no. 4, pp. 199–209, Oct. 16, 2018. DOI: 10.1007/s40860-018-0069-y.

[20] J. Zhou, J. Xia, N. Wei, *et al.*, "A traceability analysis system for model evaluation on land carbon dynamics: Design and applications," *Ecological Processes*, vol. 10, no. 1, pp. 1–14, Jan. 29, 2021. DOI: 10.1186/s13717-021-00281-w.

[21] J. Qin, Z. Li, R. Wang, *et al.*, "Industrial internet of learning (iiol): Iiot based pervasive knowledge network for lpwan—concept, framework and case studies," *CCF Transactions on Pervasive Computing and Interaction*, vol. 3, no. 1, pp. 25–39, Jan. 5, 2021. DOI: 10.1007/s42486-020-00050-2.

[22] J. Zhao, S. Ou, L. Hu, Y. Ding, and G. Xu, "A heuristic placement selection approach of partitions of mobile applications in mobile cloud computing model based on community collaboration," *Cluster Computing*, vol. 20, no. 4, pp. 3131–3146, Jul. 1, 2017. DOI: 10.1007/s10586-017-1011-4.

[23] G. Casale, M. Artac, W.-J. van den Heuvel, *et al.*, "Radon - rational decomposition and orchestration for serverless computing," *SICS Software-Intensive Cyber-Physical Systems*, vol. 35, no. 1, pp. 77–87, Aug. 26, 2019. DOI: 10.1007/s00450-019-00413-w.

[24] A. Jindal, A. K. Marnerides, P. Spachos, and A. Dvir, "Guest editorial: Smart computing for smart cities," *IET Smart Cities*, vol. 4, no. 1, pp. 1–2, Feb. 18, 2022. DOI: 10.1049/smc2.12024.

[25] M. Kansara, "A framework for automation of cloud migrations for efficiency, scalability, and robust security across diverse infrastructures," *Quarterly Journal of Emerging Technologies and Innovations*, vol. 8, no. 2, pp. 173–189, 2023.

[26] C. Brunsdon and A. Comber, "Opening practice: Supporting reproducibility and critical spatial data science," *Journal of Geographical Systems*, vol. 23, no. 4, pp. 477–496, Aug. 8, 2020. DOI: 10.1007/s10109-020-00334-2.

[27] R. Avula, "Addressing barriers in data collection, transmission, and security to optimize data availability in healthcare systems for improved clinical decision-making and analytics," *Applied Research in Artificial Intelligence and Cloud Computing*, vol. 4, no. 1, pp. 78–93, 2021.

[28] E. B. Sifah, Q. Xia, K. O.-B. O. Agyekum, *et al.*, "Chain-based big data access control infrastructure," *The Journal of Supercomputing*, vol. 74, no. 10, pp. 4945–4964, Mar. 14, 2018. DOI: 10.1007/s11227-018-2308-7.

[29] Y. Zhao, Y. Feng, Y. Wang, R. Hao, C. Fang, and Z. Chen, "Quality assessment of crowdsourced test cases," *Science China Information Sciences*, vol. 63, no. 9, pp. 1–16, Aug. 10, 2020. DOI: 10.1007/s11432-019-2859-8.

[30] P. Radanliev and D. D. Roure, "Advancing the cybersecurity of the healthcare system with self-optimising and self-adaptative artificial intelligence (part 2).," *Health and technology*, vol. 12, no. 5, pp. 923–929, Aug. 12, 2022. DOI: 10.1007/s12553-022-00691-6.

[31] R. K. L. Kennedy, T. M. Khoshgoftaar, F. Villanustre, and T. L. Humphrey, "A parallel and distributed stochastic gradient descent implementation using commodity clusters," *Journal of Big Data*, vol. 6, no. 1, pp. 16–, Feb. 14, 2019. DOI: 10.1186/s40537-019-0179-2.

[32] F. Ye, Y. Qian, and R. Q. Hu, *Big data analytics and cloud computing in the smart grid*, Jun. 15, 2018. DOI: 10.1002/9781119240136.ch8.

[33] B. Li, X. Wang, A. K. Singh, and T. Mak, "On runtime communication and thermal-aware application mapping and defragmentation in 3d noc systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 12, pp. 2775–2789, Dec. 1, 2019. DOI: 10.1109/tpds.2019.2921542.

[34] J. V. Lacey, N. T. Chung, P. Hughes, *et al.*, "Insights from adopting a data commons approach for large-scale observational cohort studies: The california teachers study.," *Cancer epidemiology, biomarkers & prevention : a publication of the American Association for Cancer Research, cosponsored by the American Society of Preventive Oncology*, vol. 29, no. 4, pp. 777–786, Apr. 1, 2020. DOI: 10.1158/1055-9965.epi-19-0842.

[35] A. Midekisa, F. Holl, D. J. Savory, *et al.*, "Mapping land cover change over continental africa using landsat and google earth engine cloud computing.," *PloS one*, vol. 12, no. 9, e0184926–, Sep. 27, 2017. DOI: 10.1371/journal.pone.0184926.

[36] M.-P. Hosseini, D. Pompili, K. Elisevich, and H. Soltanian-Zadeh, "Optimized deep learning for eeg big data and seizure prediction bci via internet of things," *IEEE Transactions on Big Data*, vol. 3, no. 4, pp. 392–404, Dec. 1, 2017. DOI: 10.1109/tbdata.2017.2769670.

[37] P. Gupta, C. Krishna, R. Rajesh, *et al.*, "Industrial internet of things in intelligent manufacturing: A review, approaches, opportunities, open challenges, and future directions," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, Oct. 21, 2022. DOI: 10.1007/s12008-022-01075-w.

[38] G. Saunders, M. Baudis, R. Becker, *et al.*, "Leveraging european infrastructures to access 1 million human genomes by 2022," *Nature reviews. Genetics*, vol. 20, no. 11, pp. 693–701, Aug. 27, 2019. DOI: 10.1038/s41576-019-0156-9.

[39] D. Mytton, "Assessing the suitability of the greenhouse gas protocol for calculation of emissions from public cloud computing workloads," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 1–11, Aug. 8, 2020. DOI: 10.1186/s13677-020-00185-8.

[40] C. R. Harrison, S. Keles, R. Hudson, S. Shin, and I. Dutra, "Ipdps workshops - atsnpinfrastructure, a case study for searching billions of records while providing significant cost savings over cloud providers," *IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum : [proceedings]. IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum*, vol. 2018, pp. 497–506, Aug. 6, 2018. DOI: 10.1109/ipdpsw.2018.00086.

[41] L. Abualigah, A. Diabat, and M. A. Elaziz, "Intelligent workflow scheduling for big data applications in iot cloud computing environments," *Cluster Computing*, vol. 24, no. 4, pp. 2957–2976, May 27, 2021. DOI: 10.1007/s10586-021-03291-7.

[42] K. Iregbu, A. Dramowski, R. Milton, *et al.*, "Global health systems' data science approach for precision diagnosis of sepsis in early life.," *The Lancet. Infectious diseases*, vol. 22, no. 5, e143–e152, Dec. 13, 2021. DOI: 10.1016/s1473-3099(21)00645-9.

[43] A. Sharma and K. D. Forbus, "Graph-based reasoning and reinforcement learning for improving q/a performance in large knowledge-based systems," in *2010 AAAI Fall Symposium Series*, 2010.

[44] R. (Ray, Z. Agar, P. Dutta, S. Ganguly, P. Sah, and D. Roy, "Mengo: A novel cloud-based digital healthcare platform for andrology powered by artificial intelligence, data science & analytics, bio- informatics and blockchain," *Biomedical Sciences Instrumentation*, vol. 57, no. 4, pp. 476–485, Oct. 15, 2021. DOI: 10.34107/kszv7781.10476.

[45] F. khan Fasuludeen Kunju, N. Naveed, M. N. Anwar, and M. I. U. Haq, "Production and maintenance in industries: Impact of industry 4.0," *Industrial Robot: the international journal of robotics research and application*, vol. 49, no. 3, pp. 461–475, Dec. 16, 2021. DOI: 10.1108/ir-09-2021-0211.

[46] M. Sudmanns, H. Augustin, B. Killough, *et al.*, "Think global, cube local: An earth observation data cube's contribution to the digital earth vision," *Big Earth Data*, vol. 7, no. 3, pp. 831–859, Jul. 21, 2022. DOI: 10.1080/20964471.2022.2099236.

[47] A. Ahmed, J. Allen, T. Bhat, *et al.*, "Design considerations for workflow management systems use in production genomics research and the clinic.," *Scientific reports*, vol. 11, no. 1, pp. 21 680–, Nov. 4, 2021. DOI: 10.1038/s41598-021-99288-8.

[48] H. Wang, C. Yu, L. Wang, and Q. Yu, "Effective bigdata-space service selection over trust and heterogeneous qos preferences," *IEEE Transactions on Services Computing*, vol. 11, no. 4, pp. 644–657, Jul. 1, 2018. DOI: 10.1109/tsc.2015.2480393.

[49] F. Dong, C. Wu, and S. Gao, "Cloud computing–based big data processing and intelligent analytics," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 24, Sep. 15, 2019. DOI: 10.1002/cpe.5531.

[50] I. Awan, M. Younas, and S. Benbernou, "Convergence of cloud, internet of things, and big data: New platforms and applications," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 23, Oct. 6, 2021. DOI: 10.1002/cpe.6668.

[51] S. Elmougy, S. Sarhan, and M. Joundy, "A novel hybrid of shortest job first and round robin with dynamic variable quantum time task scheduling technique," *Journal of Cloud Computing*, vol. 6, no. 1, pp. 12–, Jun. 15, 2017. DOI: 10.1186/s13677-017-0085-0.

[52] R. K. Barik, H. Dubey, K. Mankodiya, S. A. Sasane, and C. Misra, "Geofog4health: A fog-based sdi framework for geospatial health big data analysis," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 2, pp. 551–567, Feb. 21, 2018. DOI: 10.1007/s12652-018-0702-x.

[53] J. Yang, "Big data and the future of urban ecology: From the concept to results," *Science China Earth Sciences*, vol. 63, no. 10, pp. 1443–1456, Aug. 20, 2020. DOI: 10.1007/s11430-020-9666-3.

[54] S. Kumar, S. K. Madria, and M. Linderman, "M-grid: A distributed framework for multidimensional indexing and querying of location based data," *Distributed and Parallel Databases*, vol. 35, no. 1, pp. 55–81, Mar. 13, 2017. DOI: 10.1007/s10619-017-7194-0.

[55] null R Senthil Prabhu, null D SabithaAnanthi, null D Umamaheswari, and null S Rajasoundarya, "Internet of nanothings (iont) and machine learning (ml) – innovations in drug discovery and healthcare system," *International Journal of Frontiers in Science and Technology Research*, vol. 2, no. 1, pp. 1–8, Jan. 30, 2022. DOI: 10.53294/ijfstr.2022.2.1.0021.

[56] Z. Tong, H. Chen, X. Deng, K. Li, and K. Li, "A novel task scheduling scheme in a cloud computing environment using hybrid biogeography-based optimization," *Soft Computing*, vol. 23, no. 21, pp. 11 035–11 054, Nov. 28, 2018. DOI: 10.1007/s00500-018-3657-0.

[57] J. Xiao, W. Liu, M. Zhao, W. Zhang, and R. Xu, "Research on smart energy system technology based on cloud computing platform," *IOP Conference Series: Earth and Environmental Science*, vol. 619, no. 1, pp. 012 010–, Dec. 1, 2020. DOI: 10.1088/1755-1315/619/1/012010.

[58] G. D. Samaraweera and J. M. Chang, "Security and privacy implications on database systems in big data era: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 1, pp. 239–258, Jan. 1, 2021. DOI: 10.1109/tkde.2019.2929794.

[59] M. Maabreh, B. Qolomany, I. Alsmadi, and A. Gupta, "Bibm - deep learning-based msms spectra reduction in support of running multiple protein search engines on cloud," *Proceedings. IEEE International Conference on Bioinformatics and Biomedicine*, vol. 2017, pp. 1909–1914, Dec. 18, 2017. DOI: 10.1109/bibm.2017.8217951.

[60] N. Gupta, Khosravy, N. Patel, *et al.*, "Economic data analytic ai technique on iot edge devices for health monitoring of agriculture machines," *Applied Intelligence*, vol. 50, no. 11, pp. 3990–4016, Jul. 7, 2020. DOI: 10.1007/s10489-020-01744-x.

[61] R. Connor, R. Brister, J. P. Buchmann, *et al.*, "Ncbi's virus discovery hackathon: Engaging research communities to identify cloud infrastructure requirements.," *Genes*, vol. 10, no. 9, pp. 714–, Sep. 16, 2019. DOI: 10.3390/genes10090714.

[62] S. Namasudra, R. Chakraborty, S. Kadry, G. Manogaran, and B. S. Rawal, "Fast: Fast accessing scheme for data transmission in cloud computing," *Peer-to-Peer Networking and Applications*, vol. 14, no. 4, pp. 2430–2442, Aug. 28, 2020. DOI: 10.1007/s12083-020-00959-6.

[63] C. Kern, D. J. Fu, K. Kortuem, *et al.*, "Implementation of a cloud-based referral platform in ophthalmology: Making telemedicine services a reality in eye care.," *The British journal of ophthalmology*, vol. 104, no. 3, pp. 312–317, Jul. 18, 2019. DOI: 10.1136/bjophthalmol-2019-314161.

[64] M. Kansara, "A structured lifecycle approach to large-scale cloud database migration: Challenges and strategies for an optimal transition," *Applied Research in Artificial Intelligence and Cloud Computing*, vol. 5, no. 1, pp. 237–261, 2022.

[65] M. Cui and D. Y. Zhang, "Artificial intelligence and computational pathology.," *Laboratory investigation; a journal of technical methods and pathology*, vol. 101, no. 4, pp. 412–422, Jan. 16, 2021. DOI: 10.1038/s41374-020-00514-0.

[66] S. A. Moqurrab, N. Tariq, A. Anjum, *et al.*, "A deep learning-based privacy-preserving model for smart healthcare in internet of medical things using fog computing.," *Wireless personal communications*, vol. 126, no. 3, pp. 2379–2401, Aug. 30, 2022. DOI: 10.1007/s11277-021-09323-0.

[67] H. Zhuge and X. Sun, "Semantics, knowledge, and grids at the age of big data and ai," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 3, Oct. 29, 2018. DOI: 10.1002/cpe.5066.

[68] R. Rashidifar, H. Bouzary, and F. F. Chen, "Resource scheduling in cloud-based manufacturing system: A comprehensive survey," *The International Journal of Advanced Manufacturing Technology*, vol. 122, no. 11-12, pp. 4201–4219, Aug. 21, 2022. DOI: 10.1007/s00170-022-09873-y.

[69] J. McLeod and B. Gormly, "Using the cloud for records storage: Issues of trust," *Archival Science*, vol. 17, no. 4, pp. 349–370, Sep. 1, 2017. DOI: 10.1007/s10502-017-9280-5.

[70] R. K. Runting, S. R. Phinn, Z. Xie, O. Venter, and J. E. M. Watson, "Opportunities for big data in conservation and sustainability," *Nature communications*, vol. 11, no. 1, pp. 2003–2003, Apr. 24, 2020. DOI: 10.1038/s41467-020-15870-0.

[71] L. C. Harwell, D. N. Vivian, M. D. McLaughlin, and S. F. Hafner, "Scientific data management in the age of big data: An approach supporting a resilience index development effort.," *Frontiers in environmental science*, vol. 7, no. Article 72, pp. 1–13, Jun. 4, 2019. DOI: 10.3389/fenvs.2019.00072.

[72] M. Mahdianpari, B. Salehi, F. Mohammadimanesh, S. Homayouni, and E. W. Gill, "The first wetland inventory map of newfoundland at a spatial resolution of 10 m using sentinel-1 and sentinel-2 data on the google earth engine cloud computing platform," *Remote Sensing*, vol. 11, no. 1, pp. 43–, Dec. 28, 2018. DOI: 10.3390/rs11010043.

[73] X. You, C.-X. Wang, J. Huang, *et al.*, "Towards 6g wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Science China Information Sciences*, vol. 64, no. 1, pp. 110 301–, Nov. 24, 2020. DOI: 10.1007/s11432-020-2955-6.

[74] G. Kougka, A. Gounaris, and A. Simitsis, "The many faces of data-centric workflow optimization: A survey," *International Journal of Data Science and Analytics*, vol. 6, no. 2, pp. 81–107, Mar. 6, 2018. DOI: 10.1007/s41060-018-0107-0.

[75] R. Avula, "Assessing the impact of data quality on predictive analytics in healthcare: Strategies, tools, and techniques for ensuring accuracy, completeness, and timeliness in electronic health records," *Sage Science Review of Applied Machine Learning*, vol. 4, no. 2, pp. 31–47, 2021.

[76] R. Sangarsu, "Analysing market trends with ai," *Journal of Artificial Intelligence & Cloud Computing*, pp. 1–3, Sep. 30, 2022. DOI: 10.47363/jaicc/2022(1)123.

[77] M. R. Kamdar, J. D. Fernández, A. Polleres, T. Tudorache, and M. A. Musen, "Enabling web-scale data integration in biomedicine through linked open data.," *NPJ digital medicine*, vol. 2, no. 1, pp. 1–14, Sep. 10, 2019. DOI: 10.1038/s41746-019-0162-5.

[78] R. Zhu, L. Liu, H. Song, and M. Ma, "Multi-access edge computing enabled internet of things: Advances and novel applications," *Neural Computing and Applications*, vol. 32, no. 19, pp. 15 313–15 316, Aug. 3, 2020. DOI: 10.1007/s00521-020-05267-x.

[79] L. Wang and C. A. Alexander, "Big data analytics in medical engineering and healthcare: Methods, advances and challenges.," *Journal of medical engineering & technology*, vol. 44, no. 6, pp. 267–283, Jun. 5, 2020. DOI: 10.1080/03091902.2020.1769758.

[80] M. Hosseini, M. Wieczorek, and B. Gordijn, "Ethical issues in social science research employing big data.," *Science and engineering ethics*, vol. 28, no. 3, pp. 29–, Jun. 15, 2022. DOI: 10.1007/s11948-022-00380-7.

[81] R. Itten, R. Hischier, A. S. G. Andrae, *et al.*, "Digital transformation—life cycle assessment of digital services, multifunctional devices and cloud computing," *The International Journal of Life Cycle Assessment*, vol. 25, no. 10, pp. 2093–2098, Aug. 12, 2020. DOI: 10.1007/s11367-020-01801-0.

[82] Z. Zheng, A. Kind, and P. M. Chen, "Guest editorial: Special issue on blockchain-based services computing," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 200–202, Mar. 1, 2020. DOI: 10.1109/tsc.2020.2975678.