Journal of Information Systems Engineering and Management
2025, 10(23s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Scalable and Secure AI Infrastructure for High-Impact Industries

Srinivasa Subramanyam Katreddy,
AI Solutions Architect, 328 Camelot Dr, City: Pittsburgh State: Pennsylvania, USA - 15028
srinivasa.katreddy@gmail.com

ARTICLE INFO

Received: 14 Dec 2024

Revised: 30 Jan 2025

Accepted: 18 Feb 2025

ABSTRACT

As industries increasingly adopt AI-driven solutions, scalable and secure infrastructures become essential to manage data-intensive operations. This paper presents a modular design for scalable AI infrastructure that integrates advanced security protocols with cloud-native technologies. The proposed architecture ensures data integrity, protects sensitive information, and adapts to evolving workload demands. Applications in healthcare and finance are analyzed to demonstrate the model's versatility, highlighting improvements in scalability, reliability, and compliance. This framework serves as a blueprint for deploying secure AI systems in high-stakes industries.

Keywords: Scalable AI Infrastructure, AI Security, Cloud-Native Technologies, High-Impact Industries, Data Integrity.

I. INTRODUCTION

In recent years, billions of smart devices such as Mobile Phones, Sensors etc. are connected to the internet in the form of Internet of Things (IoT) which are increasing in a very high rate. IoT combines with other Internet applications involving People, Services, Media and Business to support the development of Serverless edge based economy and digital society. Faster development of IoT devices and sensors make life easier and makes everything accessible from where we are. IoT is a method, which connects humans and peripheral devices to the internet through internet that is, Internet of Everything (IoE). Security is the most critical requirement in IoT environment because IoT is vulnerable to attacks for several reasons, because the volume of IoT devices are increasing day-by- day and these devices are located in both the managed and the unmanaged environment. The growth of IoT is predicted as, by the end of 2020 there will be 16 billion connected devices and an average of 6 devices per person on earth. Nearly 40 Zettabytes of data will be exchanged over the network (Vermesan et al, 2011). By 2023, the number of IoT connected device users will reach 5.3 billion(i.e.) 66 percent of global population. As per Cisco Forecast the data produced from several IoT devices will be in multiples of Zetta Bytes (Wang WY & Wang Y, 2020) and its economic impact of Serverless edge work automation will be in the range of $5 to $7 trillion per year by 2015 (Mujawar, Anjum, et al. 2018). This rapid development in data leads to the challenging of processing data, storage, computing, security and maintenance. IoT data will be generated as streams in most of the application. Some of the streaming data applications are traffic monitoring, health care predictions, temperature conditions etc. Data produced from IoT connected devices are large in volume, continuously produced over the time period and suddenly rises to very high speed in network and reaches the peak. So, IoT data can also be called as Real Time Big Data (Rathore et al 2015). This exponential growth of IoT provides a large set of opportunities to users and manufacturers through new business models, new revenue generation, increasing operational efficiency. It will enable a wide applicability in many application sectors such as healthcare, inventory, environmental monitoring to mention a few.

Internet of Things shares the existing networking infrastructures along with minor modifications. The IoT allows multiple off the shelf products to connect with its network and accept almost all kind of communication protocols, i.e., RF, Bluetooth, Zig Bee, and IEEE 802.11. IoT data processing consists of three different levels (Bhandari et al, 2017),

1. Data acquisition
2. Data Aggregation

3. Data Analysis

Data acquisition is the bottom-most layer in the hierarchy. This level consists of different sensors, which are deployed in the widespread area viz automotive utilities, health care, industrial control, power grid, climate observation, and oil mining. Sensor modules are used in this environment and role of these sensors are vital because the entry process is based on these sensor values only.

The data acquired by the sensors are aggregated at this stage. The values from the multiple sensors are merged, and the accumulated data will be transmitted to the centralized processing using any one of the wireless protocols.

This data analysis is the top layer in the reference architecture, which is a data-centric application interface to help the user by analyzing the aggregated data. Generally, data analytics is application specific. Few of them are Cloud analytics tools like Cloud storage, and web servers, Bigdata analytics tools such as Weka and Hadoop and Control systems like SCADA will analyze the data.

The following fig.1 shows the impact of IoT on various sectors in 2025. Since Machine learning and Machine Learning may be appropriate for parts of many applications, this indeed will lead to increase in demand of Machine Learning and Machine learning products in huge numbers. These approaches directly and indirectly have a strong impact on growth of industries on economic things.

The other reasons to move on to Serverless edge computing is as follows:

- Centralized cloud computing cannot makeup as data's are growing enormously in Serverless edge devices. (D. Boru et al, 2015)
- Long network delay- sudden increase in the cloud centre due to flow of massive data from Serverless edge device leads to network transmission bandwidth delay (Q. Fan, N. Ansari,2018).

Serverless edge computing based IoT protocols do not take into consideration the Serverless edge computing as IoT was conceived with Cloud as a source for performing analytics. IoT involves heterogeneous type of data which can be health, agriculture, energy etc., not a single standard routing mechanism has been developed so far for existing IoT. Handling such heterogeneous data from different sensors or end devices require minimal level of analytics for predicting the data pattern and autonomously coordinating among the Serverless edge computing routers for fast data processing and exchange among the objects. Further the data Serverless edge computing captures the IoT data, data analysis are playing a vital role in areas like Smart Factories, Smart Cities, Business Grid applications, Smart Health Care etc. Even then it has some shortfalls in processing the live data as a stream. Generally the data comes from IoT devices will be behaving like streams and it will be having following characteristics (i) it will be continuously produced with unlimited flows (ii) data characteristics and behaviours will not be constantresulting in non-identical and changing behaviour over period of time. Over investigating these characteristics IoT data characteristics can be referred as real-time big data, the IoT data processing Serverless edge computing architecture shown in figure 1.

II. RELATED WORKS

Zhao Zhuoran et al (2018) work focuses on surveying the relationship between IoT edge server and Machine learning, as well as applications of Machine learning models in edge Servers and Cloud structures and also work presented in by AlencarBrenno et al (2020). They proposed a new platform called FoT-Stream in IoT which can be used to process and analyze data streams from IoT in real time in Fog. This method uses the Wavelet transform and Concept Drift methods, both of these techniques are used to observe data behaviours and decomposition. This framework was applied in smart buildings and the outcome of this research gives low processing delay, no need of constant connection of internet, low network delay S. C. Patil (2024). If the behaviour of the data changes continuously then the system fails to stick to its policy and have to transmit all data to the cloud which affects the security performance of the framework.

The large development in IoTs technologies is used for the creation of huge raw data streams in big data environments. The raw data streams in big data systems improve the calculation complexity and resource usage in cloud enabled data mining systems. The pattern based data sharing ideas are introduced by Rehman et al (2015) in big data environments. This method allows local data processing at nearby data sources and converts raw data streams into Serverless edge patterns WADITWAR, P (2024). The Serverless edge patterns contain dual utility of local Serverless edge patterns for instant actions and for participatory data sharing in big data environments. A new

two stage method is designed by Thirumalai et al (2020) for IoT cloud data stored cisco based single stage encryption but not defined in algorithm analysis.

Kos, Anton, et al. (2015) discuss the Dataflow programming model in the Bigdata processing, and they also elaborated the model which improves the execution time, power and space. They claim that the data flow model systems perform very well when compared to the control flow systems with speed-up, less power consumption and lesser space requirements than the control flow. Researchers have proposed publish/subscribe broker based multitier architecture to reduce latency in the Serverless edge computing environment S. C. Patil (2024). In this method distributed topology are difficult to attached cloud it created unbalanced dataset.

Table 1 Serverless edge computing with Machine learning approaches for IoT data

S.No	Author	Methods	Remarks	Inference
1	Aydin et al. (2015)	Accessible & scattered design designed for IoT device	Iot sensors Data stored in cloud using distributed architecture and previous direct cloud storage parameters compared.	Reduce the storage time
2	Mulani, Nazneen, et al (2015)	Privacy issues while storing and recovering in sensors data in cloud.	Protocol based IoT data stored in cloud with COBWEB clustering security method.	Security issues high time duration comparatively direct methods.
3	Zhu, Yong, et al et al (2020)	design methodology of Tile-Architecture Cluster Computing Core	Cloud computing for cluster based architecture implemented also compared existing clustering methods.	power, memory and time savings are compared to existing clustering architecture.
4	Yi et al. (2016)	A distributed IoT topology aware unstructured peer-to-peer file caching infrastructure	Iot sensor data stored using distributed peer-to-peer topology based and compute storagetime.	minimum unbalanced dataset.
5	Thirumalai et al (2020)	Two stage security methods implemented for iot sensor data for cloud and compared to RSA and KESS methods.	Basic two stage duel RSA encryption method implemented and existing ESRKGS andENPKESS are compared.	Key Generation , encryption time, and decryption time more memory is drawbacks.
6	AlencarBrenno et al (2020).	FoT-Stream in IoT analyze data streams from IoT in real time in Fog computing.	FOG computing used receive the data form iotServerless edge devices. Analysis the data in before and after applying Haarwavelet transforms methods.	Produce better result in cloud.

Ji, Changqing, et al (2012),

- Dataflow programming model intheBigdata processing is proposed
- Map reduced optimization method used in Big data environment for bioinformatics data.

- Data Transfer Bottlenecks, Iterative Optimization, Join Query Optimization andOnline
- Performance of proposed system produce better result. Storage management privacy and computation analysisare main challenges.

Tuli, Shreshth, et al (2020),

- Health Fog topology and Convolutional Neural Network algorithm used in medical dataset in fogcomputing environment
- Heart patients data set are stored in cloud and analysis the Ensemble Machine learning and Convolutional Neural Network classification methods data are used.
- Bandwidth, accuracy, time, latency and power consumed are calculated in training, test correct and test incorrect datasets. Main challengeslatency configuration.

Fotiou, Nikos, et al (2018),

- Automatic IoT data analysis using big data analytics.
- IoT sensor datas are stored using FIESTA IoT platform.
- Time, privacy, accuracy and energy driver in strategy layer performance are discussed. Noise data and temperature cause the problem in datastore in cloud.

Zhao Zhuoran et al (2018),

- Relationship between IoT Serverless edge server and Machine learning Approaches
- Machine learning approaches DNNs/CNNs with Fused Tile Partitioning methods used cloud in image data.
- Deep learning approachesreduce memorywithout accuracy sacrifice but Computation timeconsumedmore maindrawbacks.

Rehman, Muhammad Habib, and Aisha Batool (2015),

- Pattern based data sharing in cloud computing based big data environments
- In image data pattern classificationfor Serverless edge Discovery approach implemented in bigdataremote environments.
- Proposed approach effectively Handles six v's toreducedata complexity, Serverless edge availability and Complete personal datacontrol. Implementationfor MobileSocial Networkapplication difficult.

III. PROBLEM FORMULATION:

Processing of Streaming data delay: Most of the IoT applications work on real time environment and needs quicker response time for decision making. Since Cloud is connected to many Serverless edge devices and placed somewhere far away from the end devices, it can't react instantly for real time applications. Latency will be high and there will be a delay in response time. Some of the scenarios like traffic monitoring, autonomous driving cars will come under this shortfall (Li et al, 2018).

WorkLoad: Many devices are trying to connect to the Cloud at a time, which generates large amount of data at the Serverless edge which in turn have high speed transfer of data to cloud infrastructure which becomes bottleneck for Cloud infrastructure (Chen Jiasi and Xukan Ran, 2019).

Bandwidth: The transfer of huge amount of records from serverless scheme into the cloud makes network transmission to increase the bandwidth load which results in long network delay. Other shortcomings such as privacy, security is there in processing of IoT data in cloud apart from the mentioned things. Data processing as its in high need, Serverless edge computing emerged as a platform for data processing. Serverless edge computing migrates Cloud services such as networking, computing, storage capabilities and resources nearer to the end devices i.e., Serverless edge network (Wang Xiaofei, et al, 2020).

Data scalability technique: the suitable machine learning approach is need to provide intelligent decision making to activate the suitable response in the IoT environment. The intelligence isdistributed on the fog nodes

which are closer to the location of end devices, to select appropriate response at a faster rate. It not only selects appropriate response but also the timely response (TolbaAmr, and Zafer Al-Makhadmeh, 2020).

IV. RESEARCH OBJECTIVES

There are numerous issues we faced in designing mobile applications for efficient execution in serverless edge computing settings, and summarize our approaches tackling the numerous difficulties.

A. Scaling of application micro services:

Multiple instances of application micro services must be generated in order to handle processing of data from this set of cameras when a developer asks DataXe to deploy a certain application pipeline on a set of sensors, such as cameras. Poor estimates can result in over-provisioning or under-provisioning of microservices, reducing the accuracy of insights derived from analytics processing. In contrast to managing state across several cameras, auto-scaling stateless micro services are simpler than auto-scaling stateful micro services. As a result, DataXe creates a domain specific micro service replica for every reported stream generated by the AU (one-to-one mapping), while there is no each representation of registered streams to stateless micro services. DataXe dynamically scales up or scales down the number of instances of stateless microservices to effectively process the camera streams based on the processing pace of the input data streams.

B. Enhancing application-level effectiveness on a large scale:

An application consists of several microservices, and an efficient application execution necessitates both efficient communication and processing. Efficiency in communication and processing both refer to the speed at which data is transferred within and between microservices. The simultaneous handling and management of both is a challenging endeavour. By ensuring that the majority of time is spent on meaningful tasks rather than waiting for I/O, DataXe helps to monitor and improve application-level performance. The microservices are never idle or busy waiting because to the high performance communication techniques deployed. Such careful data management within and across microservices enables excellent microservice performance, which leads to increased application-level performance.

C. Scalable increasing system efficiency.

It makes sense to assume that enhancing application-level performance would also improve system-level performance. However, application-level performance and system-level performance are distinct concepts. Take into account two distinct applications that are performing a task that is shared by the two applications. Application-level performance enhancement would concentrate on getting the processing and communication correct in order to complete this typical task. But on a systemic level, we are exerting twice as much effort to complete the same activity. Such duplication obviously wastes processing power, but it can be disastrous for apps that use video analytics. For example, video analytics apps feature various AI engines that make considerable use of GPUs to perform deep learning-based AI models. These models need a substantial amount of memory resources in a GPU. Since GPU memory is limited, it may be impossible to load two copies of an AI model (such object identification) if it is used by more than one analytics pipeline processing more than one video stream. One of the analytics pipelines will malfunction in such situations.

Thus, it is crucial for DataXe to identify common AI models across several application pipelines and load the model on the GPU just once. DataXe must manage the GPU-processing of the AI model across multiple distinct application pipelines. Therefore, managing system-level performance is more difficult than managing application-level performance. DataXe carefully monitors individual applications, including numerous microservices that are a part of the application, in order to enhance system-level performance. DataXe automatically establishes a common resource of microservice instances for programs that can benefit from reuse and sharing, all while maintaining the processing pace required for each video input stream. By creating a restricted number of microservice instances and spreading them among applications, the overall utilisation of system resources is maximised, hence increasing system-level efficiency. As an example, when microservices use common deep learning models, only a single model is loaded into GPU memory and shared across applications. This keeps applications from running out of resources or crashing, which improves system-level performance. The security and privacy of each application pipeline are appropriately taken into account while enabling such sharing.

A. Proposed Trusted key based Secure Communication (TK-SCom) Model:

Based on Fig. 3, feature selection and data classification are performed in predicting the DDoS attack to classify the data as normal / abnormal. In the data classification model, several feature subsets are segregated with the association of extracting the feature. Then cross validation model is applied to improve the data accuracy. Finally the classifier of modified XG Boost algorithm is performed as it improve the computational speed and data reliability. Then the cloud based knowledge discovery is deployed to store the large volume of data and then explore the data used to perform the feature extraction.

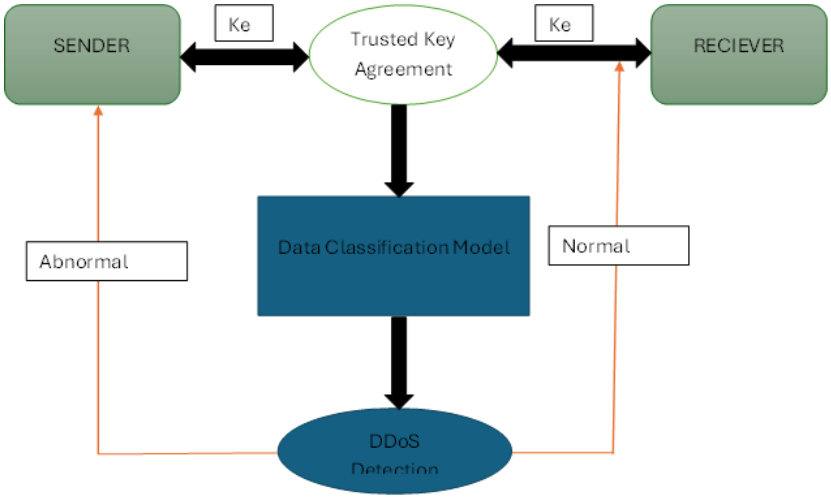

Fig. 3. Trusted key based Secure Communication (TK-SCom) Model

B. Trusted key based Secure Communication (TK-SCom) Model:

In the proposed Trusted key based Secure Communication (TK-SCom) Model, which act as the anomaly detection model as it associated with results generated form the learning classifier and perform prediction by applying best feature subsets. For the classifier, modified XG Boost algorithm is deployed as it helps to predict the DdoS attack to categorized normal or abnormal data. Then the above setup is deployed between the sender and receiver and the sender initiate the connection with some data entity, which is send to the receiver i.e. server. Between the client and server, the data classification model is deployed associated with feature extraction to predict the DdoS attack whether the data is malicious or not as represented in Fig. 3.

1. To verify the key-based agreement between the two point ‘A’ and ‘B’ where A ☐ Sender and B☐ Receiver.
2. Initially, ‘A’ request for the connection establishment with authentication along with One Time Pad (OTP) and Nonce, so that ‘A’ can initiate the establishment with session information.
3. Then after initiating the connection request along with data, then trust is ensured as it added into the data classification as it helps to predict the threat detection to categorize the data whether it is normal or abnormal.
4. The mask ID is generated for ‘A’ and ‘B’ along with the digital signature as the sender performs encryption along with the private key and decrypted by the receiver.
5. Receiver also uses same procedure to establish the connection with the sender as it uses asymmetric type of algorithm.
6. By using the integration of digital signature, verification is done based on the public and private key of ‘A’ and ‘B’.
7. Using the digital signature as it performs encryption and decryption, the verification time will be reduced as the digital signature is applied with batch verification and multiply dot operation.

Algorithm: 1 Trusted key-based Secure Communication (TK-Scom) Model

Input: Sender and Receiver ‘A’ and ‘B’ along with private and public key

Output: Connection established with data exchange

1. Initialize 'A' and 'B' with private and public key along with mask ID.
2. Establish the authentication connection.
3. 'A' sends the login establishment with certain key 'K'.
4. Classifier {Feature 1,2, ...n} = Identification of threat.
5. If (Threat = Normal Data)
6. {
7. 'B' receives data along with digital signature 'n'
8. Digital Signature {Encryption; Decryption}
9. Hash Function and information batch
10. Secure verification
11. Connection establishment completes for initiate the information exchange between 'A'and 'B'.
12. Else
13. Terminate the connection establishment
14. }

V. PERFORMANCE ANALYSIS

In the analysis, the data classification and feature extraction are performed based on the input datasets. The modifier classifier is used to detect whether the data is abnormal or normal based on a DDoS attack. In order to analyze the performance of the learning classifier, the detection rate is considered.

In Fig. 4, the operational time is determined for the proposed TK Scom Model as it performs reduced in the operational time in terms of milliseconds. The proposed TK Scom Model analyze the decrease in the value as the test data size varies from 0 to 60. The proposed model outperforms the existing model such as Identity Auth Model, ECC Auth Model, and Cloud MW Model.

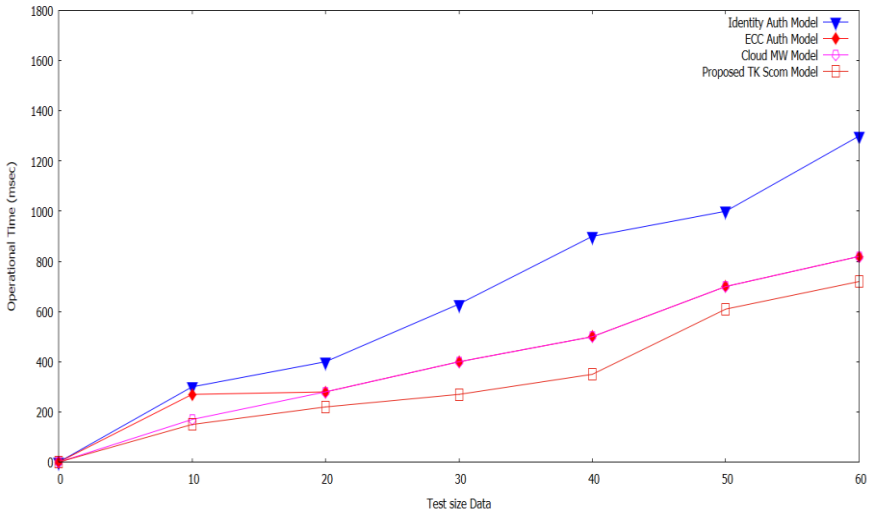

Fig. 4. Test Size Data Vs. Operational Time

In Fig. 5, the Computational cost is determined for the proposed TK Scom Model as it performs reduced in the operational time in terms of KiloBytes. The proposed TK Scom Model analyzes the decrease in the value as the test data size varies from 0 to 60. The proposed model outperforms the existing model such as Identity Auth Model, ECC Auth Model, and Cloud MW Model.

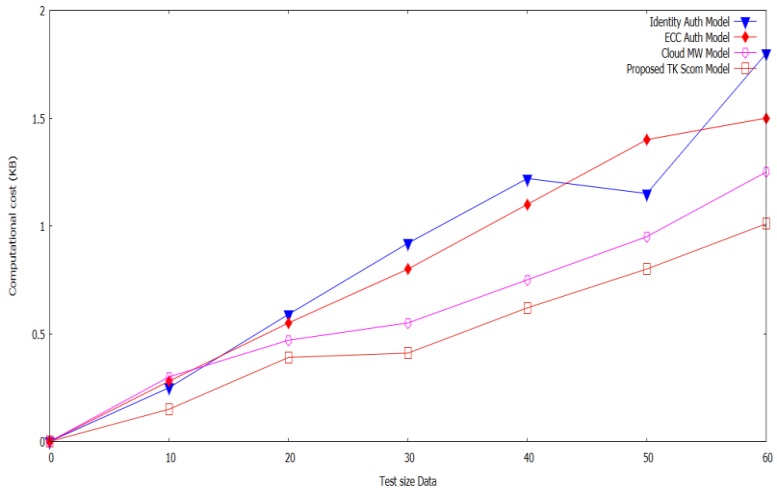

Fig. 5. Test Size Data Vs. Computational Cost

VI. CONCLUSION

Services like feature extraction and data classification are performed as it contributes to the modified learning classifier it combines the objective function. It helps to optimize the process to obtain improved classification accuracy and check whether the data is normal or abnormal based on DDoS attacks. The modified classifier performance is analyzed based on the detection rate and classification accuracy. To establish secure communication based on cryptography of encryption/decryption and between the sender and receiver, a data classification process is performed. Based on the classification of data and DDoS verification, the data is sent to the receiver to ensure secure communication based on data confidentiality and authentication. The proposed model is analyzed in terms of operational cost and computational cost is determined.

REFERENCES:

[1] G. Coviello, K. Rao, M. Sankaradas and S. T. Chakradhar, "DataX: A system for Data eXchange and transformation of streams", The 14th International Symposium on Intelligent Distributed Computing (IDC 2021), 2021

[2] S. Ranger, "What is the IoT? Everything you need to know about the Internet of Things right now", Feb. 2020

[3] E. van Eyk, L. Toader, S. Talluri, L. Versluis, A. Uţă and A. Iosup, "Serverless is more: From paas to present multi-cloud", IEEE Internet Computing, vol. 22, no. 5, pp. 8-17, 2018.

[4] Cristina L. Abad, Edwin F. Boza, and Erwin Van Eyk. 2018. Package-aware scheduling of FaaS functions. In Companion of the 2018 ACM/SPEC International Conference on Performance Engineering. 101–106.

[5] Gojko Adzic and Robert Chatley. 2017. Serverless computing: Economic and architectural impact. In Proceedings of the 11th Joint Meeting on Foundations of Software Engineering. ACM, 884–889.

[6] Alexandru Agache, Marc Brooker, Alexandra Iordache, Anthony Liguori, Rolf Neugebauer, Phil Piwonka, and DianaMaria Popa. 2020. Firecracker: Lightweight virtualization for serverless applications. In 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI'20). 419–434.

[7] Nabeel Akhtar, Ali Raza, Vatche Ishakian, and Ibrahim Matta. 2020. COSE: Configuring serverless functions using statistical learning. In IEEE Conference on Computer Communications (INFOCOM'20). IEEE, 129–138.

[8] Eyhab Al-Masri, Ibrahim Diabate, Richa Jain, Ming Hoi Lam Lam, and Swetha Reddy Nathala. 2018. A serverless IoT architecture for smart waste management systems. In IEEE International Conference on Industrial Internet (ICII'18). IEEE, 179–180.

[9] May Al-Roomi, Shaikha Al-Ebrahim, Sabika Buqrais, and Imtiaz Ahmad. 2013. Multi-cloud pricing models: A survey. International Journal of Grid and Distributed Computing 6, 5 (2013), 93–106.

[10] Kalev Alpernas, Cormac Flanagan, Sadjad Fouladi, Leonid Ryzhyk, Mooly Sagiv, Thomas Schmitz, and Keith Winstein. 2018. Secure serverless computing using dynamic information flow control. arXiv preprint arXiv:1802.08984 (2018).

[11] Sundar Anand, Annie Johnson, Priyanka Mathikshara, and R. Karthik. 2019. Low power real time GPS tracking enabled withRTOSandserverlessarchitecture. In4th IEEEInternational Conference on Computer andCommunication Systems (ICCCS'19). IEEE, 618–623.

[12] Sundar Anand, Annie Johnson, Priyanka Mathikshara, and R. Karthik. 2019. Real-time GPS tracking using serverless architecture and ARM processor. In 11th International Conference on Communication Systems & Networks (COMSNETS'19). IEEE, 541–543.

[13] Lixiang Ao, Liz Izhikevich, Geoffrey M. Voelker, and George Porter. 2018. Sprocket: A serverless video processing framework. In Proceedings of the ACM Symposium on Multi-cloud. ACM, 263–274.

[14] Dulcardo Arteaga, Jorge Cabrera, Jing Xu, Swaminathan Sundararaman, and Ming Zhao. 2016. CloudCache: Ondemand flash cache management for multi-cloud. In 14th USENIX Conference on File and Storage Technologies (FAST'16). 355–369.

[15] Gabriel Aumala, Edwin Boza, Luis Ortiz-Avilés, Gustavo Totoy, and Cristina Abad. 2019. Beyond load balancing: Package-aware scheduling for serverless platforms. In 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID'19). IEEE, 282–291.

[16] Arda Aytekin and Mikael Johansson. 2019. Harnessing the power of serverless runtimes for large-scale optimization. arXiv preprint arXiv:1901.03161 (2019).

[17] Ting Bai, Jian-Yun Nie, Wayne Xin Zhao, Yutao Zhu, Pan Du, and Ji-Rong Wen. 2018. An attribute-aware neural attentive model for next basket recommendation. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 1201–1204.

[18] Ioana Baldini, Paul Castro, Kerry Chang, Perry Cheng, Stephen Fink, Vatche Ishakian, Nick Mitchell, Vinod Muthusamy, Rodric Rabbah, Aleksander Slominski, et al. 2017. Serverless computing: Current trends and open problems. In Research Advances in Multi-cloud. Springer, 1–20.

[19] Priscilla Benedetti, Mauro Femminella, Gianluca Reali, and Kris Steenhaut. 2021. Experimental analysis of the application of serverless computing to IoT platforms. Sensors 21, 3 (2021), 928.

[20] S. C. Patil, B. Y. Kasula, V. A. Mohammed, K. Gupta and T. Thamaraimanalan, "Utilizing Genetic Algorithms for Detecting Congenital Heart Defects," 2024 International Conference on E-mobility, Power Control and Smart Systems (ICEMPS), Thiruvananthapuram, India, 2024, pp. 1-6, doi: 10.1109/ICEMPS60684.2024.10559358.

[21] K. J. Rolla, S. C. Patil, S. Madasu, R. Gupta and T. Kiruthiga, "Leveraging Machine Learning for Early Detection of Brain Tumors," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-6, doi: 10.1109/ICCCNT61001.2024.10726259.

[22] WADITWAR, P. The Intersection of Strategic Sourcing and Artificial Intelligence: A Paradigm Shift for Modern Organizations. Open Journal of Business and Management, v. 12, n. 6, p. 4073-4085, 2024.