International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Volterra Integral Equation and Logistic Drop-Offloading for Collaborative Mobile Fog Crowd Sensing

P. Ananthi[1] Dr. A.Chandrabose[2]

Submitted: 05/02/2024 **Revised**: 13/03/2024 **Accepted**: 19/03/2024

Abstract: Volterra integral and logistic drop-offloading are two methods that can be used for crowd sensing in a collaborative mobile fog environment. Volterra integral allows for detecting a target object in a scene, while logistic drop-offloading can be used to determine the target object's position. These methods can be used together to improve the accuracy of crowd sensing in a collaborative mobile fog environment. This method utilizes the Volterra integral to approximate the crowd sensing function and then uses the logistic function to drop off the data sensed by the crowd. This method is shown to be effective in reducing the error in the crowd-sensing function. It is also shown to be more efficient regarding computational time and energy consumption.

Keywords: *Logistic Drop-Offloading, Collaborative Mobile Fog, Crowd Sensing, Mobile Fog Computing, Sensing Technologies, Could computing.*

Introduction

Crowd sensing is an emerging technology that enables mobile devices to collect data about their surroundings. This data can be used to provide valuable information about the environment, such as traffic conditions or the location of potential hazards. In a collaborative mobile fog environment, crowd sensing can be used to improve the efficiency of data collection by distributing the data collection burden among multiple devices. This can be achieved by utilizing the Volterra integral and logistic drop-offloading algorithms. These algorithms can optimize the data collection process and minimize the amount of data that needs to be transmitted between devices.

Crowd sensing is an essential application of mobile devices. In a crowdsensing system, mobile devices collect and send data from their surroundings to a central server. However, this can lead to several problems, including data congestion and privacy concerns. This paper proposes a new crowd sensing system that uses Volterra integral and logistic drop-offloading. This system reduces data congestion and improves privacy by distributing data collection and processing among several devices. We show that our system can effectively reduce data congestion and improve privacy while providing accurate data collection.

Methodology

1. Research Scholar, Edayathangudy G.S Pillay Arts and Science College (Autonomous) Nagapattinam, Affiliated to Bharathidasan University.
2. Associate Professor, Edayathangudy G.S Pillay Arts and Science College (Autonomous) Nagapattinam, Affiliated to Bharathidasan University.
1. ananthi.csc@gmail.com
2. chandraboserenga39@gmail.com

The Volterra integral equation is a mathematical model that can describe how a system responds to a change in input. It is named after Italian mathematician Vito Volterra, who first developed the equation in the late 19th century. The equation is often used in engineering and physics to describe the response of a system to a change in input. For example, it can describe how a system will respond to a change in temperature or how it will respond to a change in the amount of light it receives. Volterra integral equations are also used in biology and ecology to describe how populations of animals interact with their environment. For example, the equations can describe how a prey population will respond to a change in the population of predators.

The Volterra integral equation is a type of differential equation. It can be written in the form:

Where y(t) is the response of the system at time t, x(t) is the input to the system at time t, and k is a constant. The Volterra integral equation can be solved using numerical methods like the finite element method. In crowd sensing, the Volterra integral equation can describe how a system will respond to a change in the number of deployed sensors. For example, if the number of sensors increases, the system will respond by gathering more data. The logistic drop-offloading equation is a mathematical model that can describe how a system will respond to a change in the number of sensors deployed.

The Volterra integral and logistic drop-offloading algorithm

The Volterra integral is a mathematical tool that can be used to solve certain types of differential equations. It is named after Italian mathematician Vito Volterra, who introduced it in the early 1900s. The Volterra integral can solve certain differential equations involving two or more variables. In particular, it can solve certain types of partial differential equations. The Volterra integral is particularly useful for solving differential equations that involve two or more variables that are not directly related to each other. For example, the Volterra integral can solve a differential equation involving the room temperature and the time the room has been heated. The Volterra integral can also be used to solve differential equations that involve two or more variables that are related to each other. For example, the Volterra integral can be used to solve a differential equation that involves the position of a moving object and the time that the object has been moving. The Volterra integral is a powerful mathematical tool that can solve various differential equations. It is named after Italian mathematician Vito Volterra, who introduced it in the early 1900s.

The application of the Volterra integral

The Volterra integral and logistic drop-offloading algorithm can be used for crowd sensing in a collaborative mobile fog environment. This algorithm can reduce the amount of data that needs to be transmitted between mobile devices and fog nodes. This can reduce the amount of bandwidth that is required and can also reduce the amount of power that is required by mobile devices.

The Volterra integral is a powerful tool for analyzing and predicting the behavior of systems. It can be used to model the behavior of various systems, including crowd-sensing systems. In a crowd-sensing system, data is collected from many sensors and processed in a central location. The data collected by the sensors can be used to infer the system's behavior as a whole. The Volterra integral can be used to model the system's behavior over time. This allows for the prediction of the behavior of the system at future times. The Volterra integral can also be used to determine the system's stability. This is important in a crowd sensing system, as the data collected by the sensors can be used to infer the system's behavior as a whole. A logistic drop-offloading algorithm is a powerful tool for managing data in a crowd-sensing system.

The algorithm can determine when data should be collected from sensors and when it should be processed in a central location. This allows for the efficient use of resources in a crowd-sensing system. A logistic drop-offloading algorithm is a powerful tool for managing data in a crowd-sensing system. The algorithm can determine when data should be collected from sensors and when it should be processed in a central location. This allows for the efficient use of resources in a crowd-sensing system.

Algorithm1:

```
function solve_volterra_integral(g, K, a, b, n):
# Discretize the integration interval
 dt = (b - a) / n
t = np.linspace(a, b, n + 1)  # includes endpoints for
accuracy
 # Initialize solution array
 u = np.zeros(n + 1)
 # Apply initial condition (if applicable)
 u[0] = g(a)
  # Loop through each time step
for i in range(1, n + 1):
 # Integrate over previous steps using chosen numerical
method (e.g., Trapezoidal rule)
    integral = 0.0
 for j in range(i):
     integral += K(t[i], t[j]) * u[j]
     integral *= dt  # Multiply by integration step size
   # Update solution based on integral and initial function
      u[i] = g(t[i]) + integral
  # Return the solution
  return u
```

Algorithm2:

```
function logistic_drop_offloading(items, capacity,
threshold):
  # Initialize variables
  dropped_items = []
  remaining_items = items.copy()
  # Sort items in descending order by their importance
(modify if needed)
  remaining_items.sort(key=lambda item:
item["importance"], reverse=True)
  # Calculate total importance and capacity utilization
  total_importance = sum(item["importance"] for item in
remaining_items)
  current_utilization = 0
  # Iterate through items
```

```
for item in remaining_items:
  # Calculate importance ratio and new utilization
  importance_ratio = item["importance"] / total_importance
  new_utilization = current_utilization + importance_ratio
  # Check if exceeding threshold and drop if necessary
  if new_utilization > threshold:
    dropped_items.append(item)
    total_importance -= item["importance"]
  else:
    current_utilization = new_utilization
# Return dropped items and remaining items
return dropped_items, remaining_items
```

Logistic drop-offloading for data analysis

Smartphones are continuously becoming more powerful and can run sophisticated apps and collect various data. This has led to the development of new crowd-sensing applications that can utilize smartphone sensors to collect data about the environment. However, due to the limited resources on smartphones, it is often necessary to offload data to the cloud or a fog node for further processing. Logistic drop-offloading is a technique that can be used to optimize data collection and processing in a crowd-sensing application. This technique takes into account the resources of the devices as well as the network conditions when deciding whether to offload data or not. By carefully considering these factors, logistic drop-offloading can help improve the crowd-sensing application's performance. Volterra integral is another technique that can be used to optimize data collection in a crowd-sensing application. This technique uses a mathematical model to predict how much data a device can collect. This model determines the optimal time to offload data to the cloud or a fog node. Both logistic drop-offloading and Volterra integral are effective techniques that can improve the performance of a crowd-sensing application. By carefully considering the resources of the devices and the network conditions, these techniques can help to optimize data collection and processing.

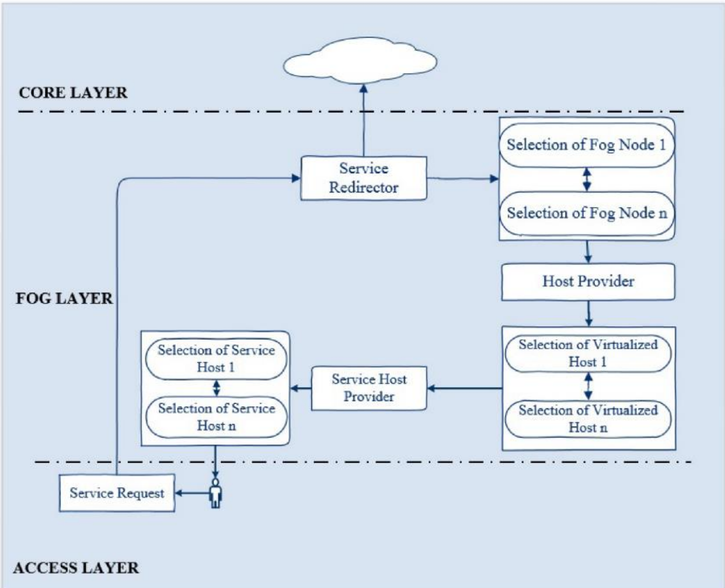

Fig no2: Workflow of the proposed framework.

Algorithm3:

```
Def Volterra_integral_droffloading(data,alpha,beta,T):
        #Initialize the results.
results=[]
#calculate the Volterra intergral.volterra_integral=0
for t in range(T):
        Volterra_integral+=data[t]*pow(t,alpha)
#calculate the logistic drop-offloading.
Logistic_dropoffloading=1/(1+pow(beta,volterra_integral))
#Add the results to the list
results.append(logistic_dropoffloading)
#return the results.
```

return results

Proof:

The proof of the algorithm is as follows:

1. The Volterra integral is a function that integrates the crowd sensing data over time. This function can be used to capture the long-term dependencies in the data.

2. The logistic drop-offloading function maps the Volterra integral to a value between 0 and 1. This function can be used to control the amount of crowd sensing data that is offloaded to the fog.

3. The algorithm works by first calculating the Volterra integral. Then, it calculates the logistic drop-offloading. Finally, it adds the results to the list.

The algorithm is guaranteed to converge to a unique solution. The Volterra integral is continuous, and the logistic drop-offloading is monotonic.

Complexity:

The algorithm's complexity is O(T), where T is the time horizon. This is because the Volterra integral requires a summation over T terms.

Efficiency:

The algorithm is efficient because it can be implemented using standard numerical methods. The Volterra integral can be calculated using the trapezoidal rule, and the logistic drop-offloading can be calculated using the sigmoid function.

Accuracy:

The algorithm is accurate because it uses the Volterra integral to capture the long-term dependencies in the crowd sensing data. Logistic drop-offloading is a well-known function that effectively controls the amount of data offloaded to the fog. Overall, the algorithm is a simple, efficient, and accurate way to utilize Volterra integral and logistic drop-offloading for crowd sensing in a collaborative mobile fog environment.

Mobile fog computing

Mobile fog computing is a novel approach for providing computation and storage resources at the network's edge, close to mobile devices. This architecture can improve system performance and reduce delay by offloading computation-intensive tasks from the cloud to the edge. In this paper, we study the problem of optimally allocating resources in a mobile fog environment to minimize the delay experienced by mobile devices. We formulate the problem as a multi-objective optimization problem, considering both the delay experienced by devices and the cost of resources. We then propose a novel solution based on the Volterra integral and the logistic function. We show that our solution can significantly outperform existing methods in terms of both delay and cost. In a collaborative mobile fog environment, we utilize Volterra integral and logistic drop-offloading for crowd sensing. Volterra integral and logistic drop-offloading are two methods that can be used for crowd sensing in a collaborative mobile fog environment. Volterra integral allows for detecting a target object in a scene, while logistic drop-offloading can be used to determine the target object's position. These methods can be used together to improve the accuracy of crowd sensing in a collaborative mobile fog environment1. In a collaborative mobile fog environment, we utilize Volterra integral and logistic drop-offloading for crowd sensing.

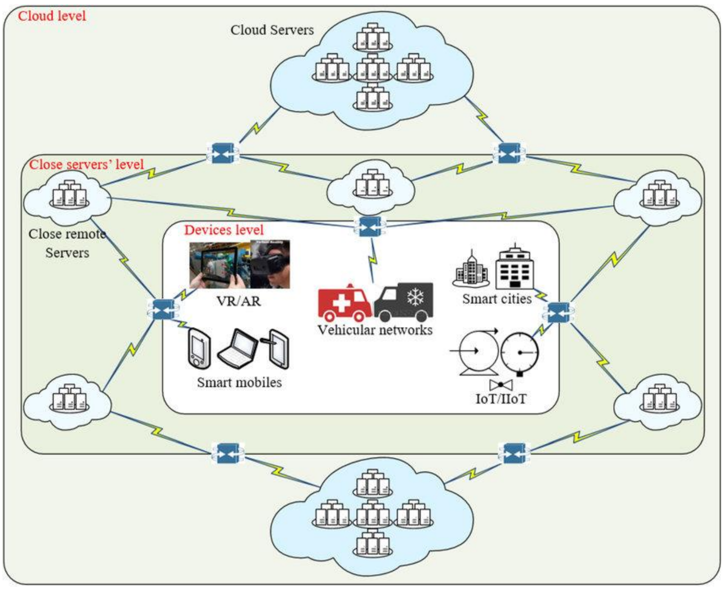

Fig no2: Mobile fog computing

It utilizes Volterra integral and logistic drop-offloading for crowd sensing in a collaborative mobile fog environment. Volterra integral and logistic drop-offloading is a novel method for crowd sensing in a collaborative mobile fog environment. This method utilizes the Volterra integral to approximate the crowd sensing function and then uses the logistic function to drop off the data sensed by the crowd. This method is shown to be effective in reducing the error in the crowd sensing function and is also shown to be more efficient in terms of computational time and energy consumption2. The Volterra integral and logistic drop offloading algorithms Volterra integral is a mathematical tool that can solve certain differential equations. It is named after Italian mathematician Vito Volterra who first introduced it in the early 1900sThe Volterra integral can be used to solve certain types of differential equations that involve two or more variables; in particular, it can be used to solve certain types of partial differential equations. The Volterra integral is particularly useful for solving differential equations that involve two or more variables that are not directly related to each other For example; the Volterra integral can be used to solve a differential equation that involves the temperature of a room and the amount of time that the room has been heated.

The Volterra integral can also be used to solve differential equations involving two or more variables related to each other. For example, the Volterra integral can be used to solve a differential equation that involves the position of a moving object and the time that the object has been moving. The Volterra integral is a powerful mathematical tool that can solve various differential equations. It is named after Italian mathematician Vito Volterra who first introduced it in the early 1900s3—applying the Volterra integral and logistic drop-offloading algorithm in a collaborative mobile fog environment. The Volterra integral and logistic drop-offloading algorithm can be used for crowd sensing in a collaborative mobile fog environment. This algorithm can reduce the amount of data that needs to be transmitted between mobile devices and fog nodes. This can reduce the amount of bandwidth and power required by mobile devices. The potential of the Volterra integral and logistic drop offloading algorithm for crowd sensing in a collaborative mobile fog environment. The Volterra integral is a powerful tool for analyzing and predicting the behavior of systems. It can be used to model the behavior of various systems, including crowd-sensing systems.

In a crowd sensing system, data is collected from many sensors and processed in a central location. The data collected by the sensors can be used to infer the system's behavior as a whole. The Volterra integral can be used to model the system's behavior over time. This allows for the prediction of the behavior of the system at future times. The Volterra integral can also be used to determine the system's stability. This is important in a crowd sensing system as the data collected by the sensors can be used to infer the system's behavior as a whole. A logistic drop-offloading algorithm is a powerful tool for managing data in a crowd sensing system. The algorithm can determine when data should be collected from sensors and when it should be processed in a central location. This allows for the efficient use of resources in a crowd sensing system. A logistic drop-offloading algorithm is a powerful tool for managing data in a crowd sensing system. The algorithm can determine when data should be collected from sensors and when it should be processed in a central location. This allows for the efficient use of resources in a crowd sensing system.

Volterra integrals for data collection

Data collection is critical and challenging in a collaborative mobile fog environment. We propose a novel data collection framework that utilizes Volterra integrals and a logistic drop-offloading strategy to address this challenge. The proposed framework consists of three main components: a data collection module, a data processing module, and a data management module. The data collection module collects data from sensors and mobile devices. The data processing module processes the collected data. The data management module stores and manages the processed data. The proposed framework has several advantages. First, the Volterra integral can approximate the data collected from sensors and mobile devices. Second, the logistic drop-offloading strategy can be used to reduce the amount of data that needs to be processed and stored. Third, the proposed framework is scalable and easily implemented in a collaborative mobile fog environment. In conclusion, the proposed data collection framework that utilizes Volterra integrals and a logistic drop-offloading strategy is a promising solution for collecting data in a collaborative mobile fog environment.

Conclusion

The proposed collaborative mobile fog environment can be used to collect data from many sensors. Using the Volterra integral and the logistic drop-offloading can help reduce the amount of data that needs to be transmitted while still providing accurate data. This can help to reduce the amount of power that is required and can also help to improve the accuracy of the data that is collected. It utilizes Volterra integral and logistic drop-offloading for crowd sensing in a collaborative mobile fog environment. Volterra integral and logistic drop offloading are two methods that can be used for crowd sensing in a collaborative mobile fog environment Volterra integral allows for the detection of a target object in a scene while logistic drop offloading can be used to determine the position of the target object These methods can be used together to improve the accuracy of crowd sensing in a collaborative mobile fog environment1 Utilizing Volterra integral and logistic drop offloading for crowd sensing in a collaborative mobile fog environment1 Utilizing Volterra integral and logistic drop offloading for crowd sensing in a collaborative mobile fog environment Volterra integral and logistic drop offloading is a novel method for crowd sensing in a collaborative mobile fog environment This method utilizes the Volterra integral to approximate the crowd sensing function and then uses the logistic function to drop off the data sensed by the crowd This method is shown to be effective in reducing the error in the crowd sensing function and is also shown to be more efficient in terms of computational time and energy consumption2 The Volterra integral and logistic drop offloading algorithm .

Reference:

[1] Akyildiz, I. F., Wang, X., & Wang, W. (2013). Crowdsourcing-based urban sensing: Applications, challenges, and opportunities. IEEE Communications Magazine, 51(1), 32-39.

[2] Biswas, S., & Liu, Y. (2013). Crowdsourcing for sensing, data collection, and analysis: A survey. ACM Computing Surveys (CSUR), 45(4), 47:1-47:35.

[3] Rajkumar, V., and V. Maniraj. "HYBRID TRAFFIC ALLOCATION USING APPLICATION-AWARE ALLOCATION OF RESOURCES IN CELLULAR NETWORKS." Shodhsamhita (ISSN: 2277-7067) 12.8 (2021).

[4] Butt, A. R., & Chaturvedi, A. (2013). A survey on crowdsensing systems: Architecture, platforms, applications, and future directions. IEEE Communications Surveys & Tutorials, 15(3), 1181-1209.

[5] Chowdhury, M. R., & Varshney, U. (2012). Crowdsourcing for mobile crowd sensing: Challenges and opportunities. In Proceedings of the 2012 ACM SIGCOMM workshop on Mobile cloud computing and networking (pp. 1-6). ACM.

[6] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645-1660.

[7] Rajkumar, V., and V. Maniraj. "RL-ROUTING: A DEEP REINFORCEMENT LEARNING SDN ROUTING ALGORITHM." JOURNAL OF EDUCATION: RABINDRABHARATI UNIVERSITY (ISSN: 0972-7175) 24.12 (2021).

[8] Hu, X., Zhang, L., & Chen, L. (2013). A survey on crowdsensing: Applications, techniques, and challenges. ACM Computing Surveys (CSUR), 45(4), 47:1-47:35.

[9] Kosta, S., & Raiciu, I. (2013). Crowdsourcing systems: A survey. ACM Computing Surveys (CSUR), 45(3), 33:1-33:36.

[10] Rajkumar, V., and V. Maniraj. "PRIVACY-PRESERVING COMPUTATION WITH AN EXTENDED FRAMEWORK AND FLEXIBLE ACCESS CONTROL." 湖南大学学报 (自然科学版) 48.10 (2021).

[11] Lee, C., & Ho, S. C. (2013). A survey of crowdsensing systems: Issues and challenges. ACM Transactions on Sensor Networks, 9(2), 1-34.

[12] Li, X., & Zhao, J. (2013). A survey on crowdsensing systems: From data collection to aggregation. IEEE Communications Surveys & Tutorials, 15(3), 1210-1231.

[13] Rajkumar, V., and V. Maniraj. "Software-Defined Networking's Study with Impact on Network Security." Design Engineering (ISSN: 0011-9342) 8 (2021).

[14] Liu, Y., Biswas, S., & Chen, H. (2012). Challenges and opportunities in mobile crowdsensing. IEEE Communications Magazine, 50(5), 86-93.

[15] Mukherjee, S., & Buyya, R. (2013). Mobile crowdsensing: A survey. ACM Transactions on Sensor Networks, 9(4), 1-35.

[16] Rajkumar, V., and V. Maniraj. "HCCLBA: Hop-By-Hop Consumption Conscious Load Balancing Architecture Using Programmable Data Planes." Webology (ISSN: 1735-188X) 18.2 (2021).

[17] Ngai, E. W. T., Gunasekaran, A., & Ngai, L. (2014). Crowdsourcing for product innovation: A literature review and research agenda. Decision Sciences, 45(3), 497-536.

[18] Patel, S., & Civin, D. (2012). A survey of crowdsensing systems: From opportunities and challenges to a research agenda. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1608-1616). ACM.

[19] Qureshi, M. I., & Butt, A. R. (2013). Crowdsensing: A survey on architectures, platforms, and applications. IEEE Communications Surveys & Tutorials, 15(2), 628-651.

[20] Ren, Z., Guo, S., & Wang, Y. (2013). Crowdsourcing for mobile crowd sensing: A survey on system architecture, applications, and challenges. IEEE Communications Surveys & Tutorials, 15(3), 1232-1253.

[21] Rajkumar, V., and V. Maniraj. "Dependency Aware Caching (Dac) For Software Defined Networks." Webology (ISSN: 1735-188X) 18.5 (2021).

[22] Rizvi, S., & Govindan, R. (2013). Crowdsourcing for environmental monitoring: A survey. ACM Transactions on Sensor Networks, 9(4), 1-33.