Machines and Algorithms
http://www.knovell.org/mna

Research Article

Service Discovery Framework for Fog Computing

Rizwan Amin[1,*], Asif Raza[1] and Tanzeela Kausar[2]

[1]Department of Computer Science, Bahauddin Zakariya University, Multan,60000, Pakistan
[2]Institute of Computer Science & Information Technology, The Women University, Multan,60000, Pakistan
[*]Corresponding Author: Rizwan Amin. Email: muhammadrizwanbrw@gmail.com
Received: 09 November 2022; Revised: 07 December 2022; Accepted: 03 February 2023; Published: 6 March 2023
AID: 002-01-000017

Abstract: Fog computing has experienced significant growth, enabling access to cloud-based services from nearby fog nodes instead of relying on centralized cloud systems. This decentralization allows for the distribution of services across various systems, providing customers with increased proximity and efficiency in accessing the resources they need. In addition to the shared skills of cloud computing, fog computing gives additional features that facilitate mobility, low latency, and real-time interactions. Nevertheless, it remains a tough issue to identify distributed fog computing resources. The purpose of this research was to develop a fog computing service discovery framework that makes use of machine learning. Random Forest, Logistic Regression, GridSearchCV, and Support Vector Machine are only a few of the current machine learning algorithms that the suggested framework uses in the service discovery process. Results are evaluated and contrasted based on the suggested learning model's performance. Because of their specific goals, Random Forest and GridSearchCV are our primary ML algorithms, even if there are others that show superior accuracy.

Keywords: IOT; Cloud Computing; Fog Computing; Service Discovery; Target access best node;

1. Introduction

The Internet of Things (IoT) refers to a community of physical devices which are interconnected with other internet-linked devices and structures that use sensors, software, and technology. The IoT revolution has the capability to create new market possibilities and enterprise fashions in areas such as safety, privacy, and technological interoperability. The extensive adoption of IoT devices is expected to impact various facets of our daily lives. These devices consist of secondary customers and innovative objects like power supplies, home automation, and internet access devices. It is expected that the global economic system of IoT will grow from $3.9 trillion to $11.1 trillion by 2025 [1].

Because the IoT continues to grow, the idea of edge devices or fog computing devices has emerged. Fog computing is one of the cloud models which have received popularity due to the growing interest in cloud computing. Fog computing allows the advent of low-latency network connections in computing devices, analytics, and nodes [2]. Further, network devices, cloud servers, and fog computing nodes can offer diverse services such as region detection, aspect transmission, facilitating mobility, low latency, and real-time interactions.

However, fog computing resources are allotted throughout a multi-layer network and are managed by multiple owners, making the resource discovery problem even more complex. Decentralized resource discovery is critical to deal with this problem [4]. Several systems and toolkits for fog computing have been proposed, however researchers still face problems identifying the precise ambiguous nodes in a service. Many researchers discuss the services discovery with quick access to nodes, reduced communication latency, improved performance and security during information transformation, and authentication [5]. However, most of the models do not focus of the accuracy and most beneficial serving nodes.

To address these challenges, we endorse a service discovery infrastructure for fog computing, using logistic regression algorithms, support vector machines, GridSearchCV, and random forest techniques.

The rest of the sections of the paper are presented as follows Section II gives you background study, Section III a literature overview, Section IV the proposed framework the results and experiments of the version. Ultimately, Section V contain conclusion.

2. Background Study

The Internet of Things (IoT) is a network of physical things that are integrated in hardware and software that can connect to and exchange data with other devices and computers on the internet. Because of low-cost computing, cloud computing, big data analytics, and mobile technologies, all appliances, from simple household items to cutting-edge industrial machines, can share and retain data with minimum human participation. Sensors that are both affordable and reliable have made IoT technology more accessible to a broader variety of manufacturers.

The Internet of Things (IoT) is a growing technology [7]. Millions of petabytes of data are collected every day by billions of networked devices throughout the world. This data might contain vital information about home safety, entertainment needs, water saving, and decreased fuel emissions. The Internet of Things has been introduced to us through smartphones, live TV services, smart TVs, and other gadgets.

2.1. Computing in the Fog

Fog computing is a sort of decentralized computing in which processing resources are shared between data sources and a cloud or other data center. Fog computing is a way of managing user requests on edge networks. Routers, gateways, bridges, and chassis are examples of fog-filled network-related equipment. These solutions, when implemented by specialists, will be capable of performing both computer and networking operations. Though their capacity is limited in comparison to cloud waitrons, their geological size and regionalized geography enable them to deliver trustworthy services across a wide area. The term "fog" refers to a location that is physically closer to a user's computer system than the cloud [10].

3. Review of Literature

A novel computing architecture known as fog computing increases the processing capabilities of cloud services. There are resources in the form of fog nodes along the beach [1]. For location, fog provides efficient data processing and storage, decreased latency, and network bandwidth savings. One of the criteria for attaining these goals is the identification of optimal fog using the edge tool. Sign one before deciding on the best contract. There are numerous approaches to achieve the needs of these objectives. The goal of this study is to provide a comprehensive assessment of the country. Explore the fog node area to locate and select art. To do this, we create another first. Certain requirements must be followed in order to establish the greatest fog retention. So, we define and compare this to the literature from 2014 to 2021, as well as the specific identifying criteria. There is still a significant quantity of fog. Based on the investigation, we propose gaps and unsolved issues in the available literature. The largest node was chosen and utilized to detect fog.

When distributed systems are deployed on fuzzy computing nodes, microservice architecture becomes an important and difficult challenge. In cloud contexts, service discovery algorithms are widely deployed in computer networks and clusters [3]. Using a service discovery technique, the service container writes its own data in a customer-centric manner. When a request comes in, the services proxies utilize a lookup table to route it to an available server. Clients in fog computing, on the other hand, are no longer restricted to cloud resources, but can instead request resources from local fog nodes.

Smart buildings are comprised of real-time information systems comprised of a variety of sensors and equipment linked by communication networks [4]. On the other side, its complexity develops. The implementation of Internet of Things (IoT) models has the potential to play a greater role in building energy management by significantly increasing its capacity to acquire, analyze, and display data as knowledge. This complex and growing data requires sophisticated processing resources, which cloud computing models may provide. Although technology is widely utilized, it has problems such as insufficient latency, traffic congestion, and a lack of support for localization. As a result, fog has emerged as a viable solution for offering fluctuating resources at network edges. Intelligent smart building framework for cloud and location, a unified framework for IoT devices placed in buildings to do computing, routing, and simultaneous interaction with apps operating in facilities.

Smart agriculture is a strategy of achieving long-term agricultural and food production that makes use of interactive tools and cutting-edge technology. Smart farming is purely based on the Internet of Things, which relieves farmers and farmers of physical labor and thereby enhances production in every way imaginable. With the present emergence of agricultural trends, the Internet of Things provides enormous benefits such as efficient water consumption, improved resources, and more. The main distinctions are the large benefits and the current occurrence of revolutionary agriculture.

The smart cloud service discovery framework (2020) was developed by Al-Sayed et al. He regards his method as rational, clear, adaptable, and all-encompassing. As we all know, intelligent gadgets make considerable use of artificial intelligence technologies. Standard: Provides all cloud providers with a uniform default identity model for offering their services in a competitive cloud service segment. It requires a flexible limit that allows even non-technical employees to construct text queries and grasp ambiguity and complicated investigations. Because it covers all common functional elements of cloud service providers, this technique gives a graphical interface with little understanding of cloud concepts while addressing the restrictions of all cloud providers. Users may now ask complex questions in their own language. The results show a low error rate, excellent agreement, and the utilization of conventional detection methods.

To address the limitations of previous solutions, Sharma et al. [23] created a blockchain-based cross-cloud discovery platform for non-federated intermediate models. One of the goals is to enhance the design of the P2P structure combination so that CSPs may be formed almost densely. It also implies that using a blockchain for inter-cloud service discovery will eliminate the need for a right-hand third entity or intermediary among participating cloud service providers, with scalable service detection being the primary benefit of the proposed method. This strategy, however, is limited by delayed reflexes.

Finally, Ka et al. [24] propose the SIM SIM service discovery to improve service discovery performance for P2P mobile cloud applications. SimSim uses the same 2D grayscale environment and segmentation to show services. Single-hop routing and thorough environmental research on the effectiveness of internal control in difficult situations. In terms of similar hash maps, the proposed method outperforms previous approaches in terms of clustering. Sorting hashes improves the comparability of layer and stored resources. Because it uses a hierarchical hash aggregation approach to seek installation and discovery methods for services with keywords criterion in the gray space, this technique provides services with comparable geographic proximity. As a result, the SIM outperforms in job scheduling and reaction time. Its weakness, however, is a lack of security.

In addition, in order to explore intricate system behavior, Zahoor et al. [27] proposed UML characteristics for service discovery in an organizational cloud architecture, as well as the synthesis approach necessary for system modeling. Their strategy is to make the system easier to understand and use for consumers, developers, and developers. The goal of this methodology is to give a UML configuration file to describe a discovery method for establishing proxy-based online facilities, as well as the best UML file type that can be used with any discovery strategy. Furthermore, it is not overly complicated, and the mechanism's weakness is its poor response time.

Finally, Abbas et al. [28] provide a cloud discovery service that includes an acceptable job design model and cloud ontology. To some extent, the proposed architecture addresses vendor security issues while also improving cloud portability and interoperability. The multi-agent schema is made up of several smart agents such as Directories Mediator, Detection Mediator, Rating Agent, Consumption Agent, and others. This document describes the methodical approach of creating a reference model. It is used in multi-agent systems to build intelligent agents and provide interoperability. Because it overcomes portability and interoperability [28] difficulties in cloud technology, as well as general supplier security concerns, the findings show that the reference design performs much better in terms of lookup effectiveness, reply time, and implementation time.

A truly decentralized mechanism. Navimipour [26] provided a trust-based strategy to personalizing cloud service discovery and verification techniques. The NuSMV model validator, as well as relevant models, administrative diagrams, historical logic, and the NuSMV model validity for verifying model constructions and elements, are discussed. Because the predicted design and structure of trustworthy HR discovery are established using temporal logic, the approach may efficiently search for effective, available, efficient, and discontinuous services. Clean, fair, and long-lasting. However, this technique disregards quality of service (QoS) and permissions.

To achieve safe service discovery for IoMT, we suggest using SSD blockchain-based fog computing [22]. We suggest a cross-blockchain design comprised of numerous self-governing comparable blockchains, motivated by the premise that a large number of smoke producers may successfully offer clients in various regions. In this situation, each secure concurrent blockchain may demonstrate the reliability of a certain multimedia application at a fair cost. In a recent publication, we developed a secrecy image-based domain query approach for fog-enabled vehicle networks. This research also looks into the improved performance and security of proposed SSDs [25].

4. Proposed Methodology

The main principles and procedures of the suggested approach were covered in this study. All of the methods and strategies utilized in the fog computing service discovery framework must then transit through various phases of the framework layer. The flowchart and algorithm below demonstrate all of the basic steps of the proposed technique.

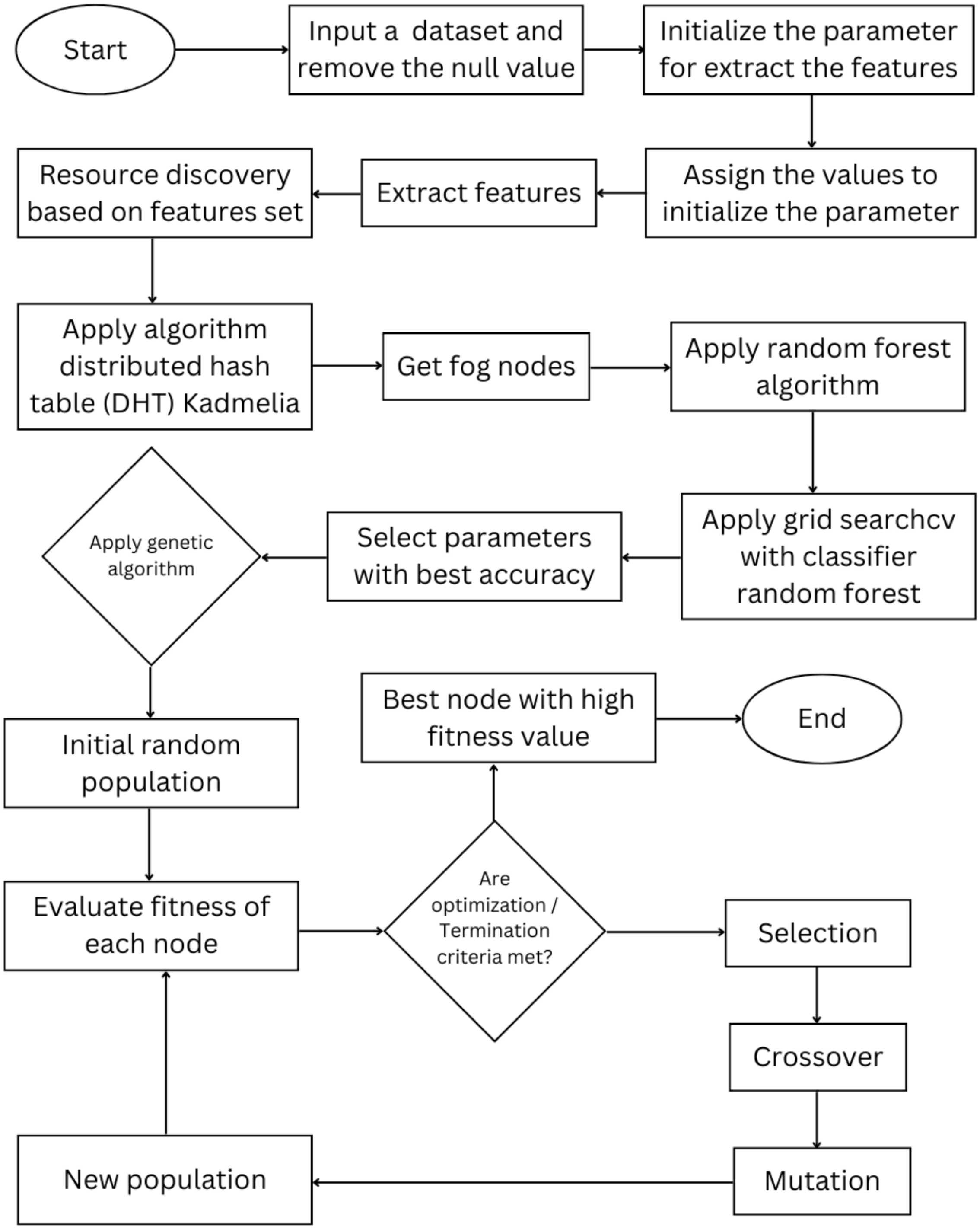

Figure1: Methodology Data Flow Diagram for GA

- Algorithm 1: Service Discovery Framework for Fog Computing

Input: **D** ▷ **D = Dataset**
Output: n is a best matching node
D ← Remove $(Null/NaN, D)$
1. $(X, Y) \leftarrow Extract\ (D)$ ▷ X , Y = Extract the values x,y on dataset
2. $(x\_train, x\_test, y\_train, y\_test) \leftarrow split\ (X, Y)$
3. S ← Assign (n_estimators, max_features, max_depth, max_samples) ▷ S = Set of Hyper Parameters (n_estimators, max_features,max_depth, max_samples)
4. $S^{'}$ ←Pram_Grid (n_estimators, max_features, max_depth, max_samples)
5. (R, A) ←GridsearchCV (Random Forest, $S^{'}$, S) ▷ R = Best parameter, A = Accuracy
6. n$\epsilon S'$ ←GA(R) ▷ GA = Genetic Algorithm
7. **T**← T Random population ▷ T= Random Population set generated by GA
8. N; ▷ N =No. of Iterations set to predefined values
9. FFV = 5; ▷ FFV = Fitness function value
10. i = 0; ▷ i = set of condition
11. While (FFV! = 0 OR i ≠ N)
12. O←Count (Desired Resources – T[i] ▷ ∀ i ϵ size(T) ▷ T[i] = Nodes available resources
13. $O^{'}$ ← Mutation(O)
14. End
15. Output n ← best of O

- Algorithm 2: Genetic Algorithm

Input: D ▷ **P = Generate Initial Random Population**
Output: n ▷ n = Best Node, high fitness value
1. S ← Param ▷ Param = Set of GA parameters
2. P ← Assign (**S**) ▷ P = Generate Random Papulation
3. F ← P ▷ F = Evaluate Fitness Value
4. While (F == T) ▷ T = Termination Criteria
5. $P^{'} \leftarrow S^{'}$ ▷ $S^{'}$= Select a new population
6. $C^{'} \leftarrow P^{'}$ ▷ $C^{'}$ = Cross Over
7. $M^{'} \leftarrow C^{'}$ ▷ $M^{'}$ = Mutation
8. n ← $M^{'}$ ▷ n = Generate a new population
9. END
10. **Output** n ← best node with high fitness value

4.1. Remove Empty Values in CSV dataset

The data set was not without flaws. They can end up with values that are incorrect, corrupted, or missing. I have no NULL or NaN values for the project I'm working on. This discussion group will teach you how to construct a simple Python script to check if your dataset contains NULL or NaN values and, if so, how to update the data. He'll offer you alternatives.

4.2. Initialize and Assign the Value of Parameters

At this point, every model we've built has required us to begin with parameters based on a certain distribution. So far, we've examined the initial pattern and how this decision was reached. These things may appear to be unimportant to you. Instead, the initial mode is key for maintaining numerical stability and playing an important role in the learning service discovery architecture.

4.3. Parameter Management

Once we've chosen a structure and defined our hyperparameters, we start a learning loop with the objective of finding parameter values that reduce our loss function. These parameters will be required for future projections following training [15]. We may also need to get parameters in order to reuse them in another framework, save our prototype to disk in order to use it with other programmers, or collect scientific knowledge for testing purposes.

4.3.1. Parameter Access

When a model is defined by a join class, we can access any class by indexing the first model as if it were a slope. In its table, we can easily find the parameters of each coat. As tails, we may test the parameters of the next totally related coating.

4.4. Feature Extraction

Feature extraction is a dimension reduction approach that condenses a huge dataset into a smaller processing set. One advantage of such large datasets is the presence of a high number of variables that require considerable processing resources to execute [15]. Feature extraction is a method of discovering and/or merging parameters into features that efficiently reduces the amount of data that must be addressed while maintaining the correctness and completeness of the real data set.

In order to get to the meat of the data, analysts painstakingly isolated a number of attributes. 'Time,' 'Period Time,' and 'Time since reference,' among others, are temporal components that give a deeper understanding of the time dimension. The frames are subjected to additional analysis utilizing 'Previous Capture Frame' and 'Previous Display Frame,' which allows for a comprehensive evaluation of the data. 'Payload Length' and 'Total Length,' which stand for payload attributes, reveal details on the structure and amount of the material. The 'Date' function, which adds a chronological anchor to the data, completes the picture and makes it easier to analyze.

4.5. Random Forest

Random Forest is a supervised machine learning method. The "forest" they make is made up of deciduous trees that are typically gathered together in "flocks." The main notion of the activation approach is to integrate the learning model in order to optimize the entire effect.

4.6. GridSearchCV

To begin, what is a web search? It is the process of experimenting with totally acceptable hyper - parameters in order to get the optimal values for a certain model. As previously stated, hyperparameter settings have a major influence on model performance. It is important to note that knowing the optimal solution of hyper - parameters in advance is impossible, thus we must test all conceivable values to find the best one.

- Estimator: Pass the model instances for which the hyperparameters should be verified.
- Params_grid: the dictionary object holding the hyperparameters to be tested.
- Scoring: You may just enter a valid string as the evaluation measure you want to employ.
- CV: A specific number of cross-validations must be attempted for each set of hyperparameters.
- Verbose: While fitting the data to GridSearchCV, set it to 1 to receive a detailed report.
- Jobs: If you provide -1 for the number of processes to run in parallel for this task, all available processors will be used.

4.7. Genetic Algorithms

They are commonly employed to provide high-quality solutions to optimization and search problems. Using the genetic method, we obtain an initial random population, which is then utilized to evaluate the fitness value of each node. As a result, two outcomes are possible. If the termination requirements are fulfilled, it shows the best node with a high fitness value; otherwise, the selection and crossover processes are repeated.

4.7.1. Fitness Function

The fitness function assists in determining everyone's fitness. It determines each individual's own fitness score, which influences their chance of reproducing. The more fit you are, the more likely you are to be chosen for fertility.

Fitness function and Node fitness formula, for example, are explained further below.

$$Fitness\ Function\ Value\ =\ Desired\ Resources - Nodes\ Available\ Resources$$

$$FFV\ =\ 11111 - 111$$

$$FFV\ =\ 11$$

$$FFV\ =\ 2$$

$$Node\ Fitness\ =\ Node\ Available\ Resources\ /\ desired\ Resources$$

$$Node\ Fitness\ =\ 2\ /\ 5$$

$$Node\ Fitness\ =\ 0.40$$

4.7.2. Initial Population

The first stage in the Genetic Algorithm is to populate the population. Our suggested algorithm incorporates this step. Genetic algorithms produce a random population. We analyze the population of the 51 nodes and compute the fitness value for each node.

$$Population\ population\ =\ new\ Population(\,)$$

$$Demo.population.initialize(51)$$

$$\sum_{i=0}^{L} new\ individual(\,)$$

4.7.3. Selection

In this procedure, parents are picked at random from the current population.

$$P(N_{51}) = \left|\frac{P(N_{51})}{\sum_{i=1}^{n\,pop} f(N_o)}\right|$$

4.7.4. Crossover

At this point, the tails of the two parents are exchanged to make new offspring at a randomly determined crossover site and pick the next population among people.

$$int\ crossOverPoint\ =\ rn.nextInt(population.individuals[0].nodeLength)$$

4.7.5. Mutation

In the process of mutation, the resources are replaced with one another. Mutation derives the number from the total resources of the nodes. When the value is 0, it gets replaced with 1, and vice versa.

$$int\ mutationPoint\ =\ rn.nextInt(population.individuals[0].nodeLength)$$

5. Experiments and Results

We receive the results after applying the machine learning model to the given dataset. The Random Forest Algorithms that we used on our Model. However, in order to compare the results, we also tried various different Algorithms on the data set.

Other algorithms provide superior accuracy, but our focus is on Random Forest and GridSearchCV, which are used for certain purposes and methods.

5.1. Accuracy of the Training Model with Train and Test Datasets

First, we split the train and test data sets in the original model. And we allocate 80% of the knowledge set to training and 20% to testing. When we tested logistic regression, we achieved 59% accuracy. Then I tested Support Vector Machine, which had a 60% accuracy rate. Then I looked into Random Forest, which has a 62% accuracy rate. GridSearchCV was tested and found to be 72% accurate, which is higher than any other algorithms.

After Applying the Machine Learning Model on Given Dataset as we get the Results. The Algorithms that we applied it Random Forrest on our Model. But for compare the Results we try to some other Algorithms as well on data set.

The other Algorithms is also giving better accuracy but our focus is on Random Forest and GridSearchCV that applied with specific purpose and Methods.

5.2. Training Model's Accuracy

As evidenced by the accuracy table. Logistic Regression has a 59% accuracy, Support Vector Machine has a 60% accuracy, Random Forest has a 62% accuracy, and GridSearchCV has a greater accuracy than all of them together.

Table 1: Training Model's Accuracy

Models	**Accuracy**
GridSearchCV	72%
Random Forest	62%
Support Vector Machine	60%
Logistic Regression	59%

5.3. Confusion Matrix

A confusion matrix is a table which is used to describe a classification procedure's presentation. The effectiveness of a cataloging model is shown and brief using a confusion matrix. Values show a confusion matrix with benign nodes accessible and not accessible is given below

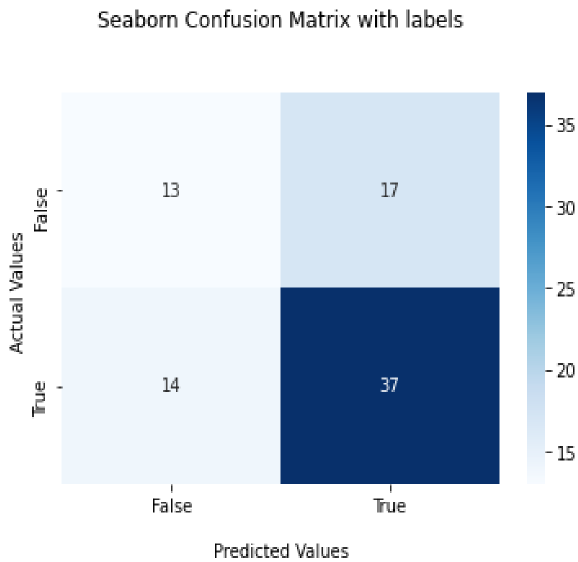

Figure 2: Confusion Matrix

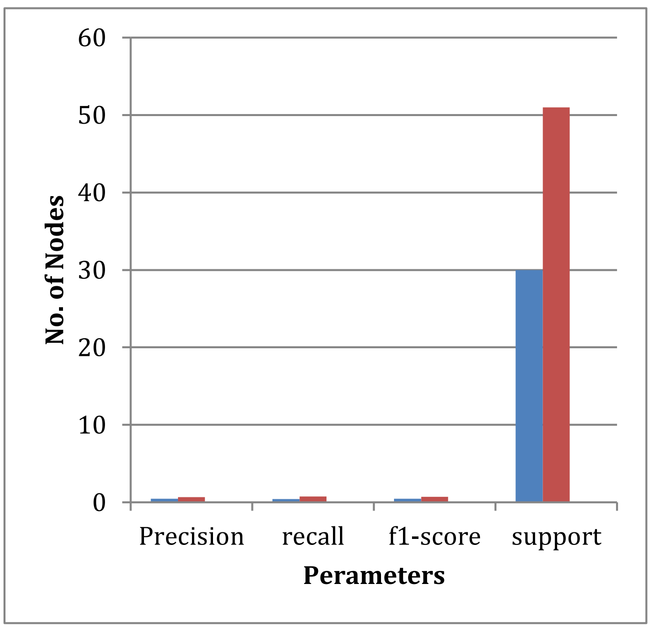

Figure 3: No. of Nodes with Parameters Graph

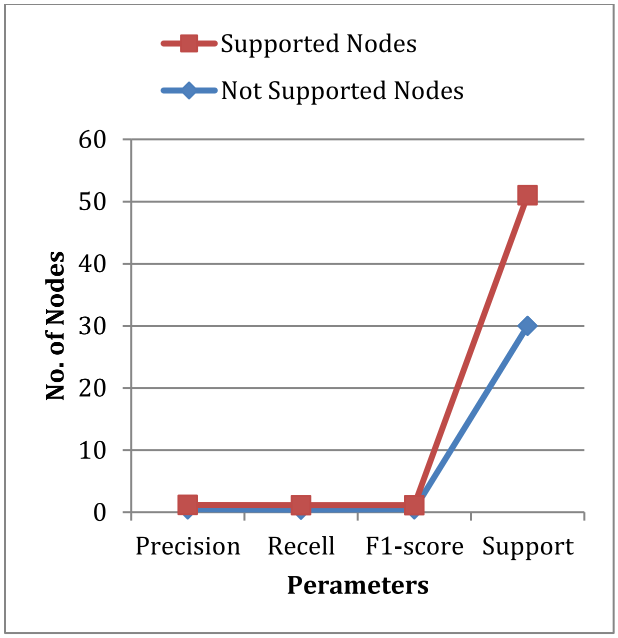

Figure 4: No. of Nodes with supported and not supported graph

Table 2: Confusion Matrix Values with Derivation

Measure	Value	Derivations
Sensitivity	0.7308	TPR = TP / (TP + FN)
Specificity	0.4333	SPC = TN / (FP + TN)
Precision	0.6909	PPV = TP / (TP + FP)
Negative Predictive Value	0.4815	NPV = TN / (TN + FN)
False Positive Rate	0.5667	FPR = FP / (FP + TN)
False Discovery Rate	0.3091	FDR = FP / (FP + TP)
False Negative Rate	0.2692	FNR = FN / (FN + TP)
Accuracy	0.6292	ACC = (TP + TN) / (P + N)
F1 Score	0.7103	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.1692	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

6. Conclusion

The various limitations of cloud computing have been solved through the use of platform fog computing [1]. By connecting computing, storage, and networking to the network edge, fog helps minimize compute latency, security, network bandwidth, reaction time, connection fees, and power consumption when network resources are limited. To do this, effective fog node services must be discovered and selected as needed. This is a precondition for reaching these goals. The best fog node service should be placed first, followed by the best fog node. To achieve these objectives, certain selection criteria must be satisfied. The goal of this research is to locate the optimal fog node for the service discovery framework that can be used in fog computing. In this study, we proposed a model for a fog computing service discovery framework based on random forest and GridSearchCV. In this study, we provided a model for a fog computing service discovery framework based on GridSearchCV and Random Forest. We apply our suggested machine learning model to the given dataset in this study, analyze the outcomes, and explain the findings. We employed gridsearchcv, logistic regression, random forest, and support vector machine techniques in our model. To compare the findings, we run a variety of other algorithms on the dataset. Although other algorithms provide more accuracy, we are focusing on random forest and GridSearchCV since they were used with specific purposes and targets. We changed our proposed model using Python environments. Finally, GridSearchCV has the best accuracy of 72%, while Logistic Regression, Support Vector Machine, and Random Forest have the lowest accuracy of 59%, 60%, and 62%, respectively. GridSearchCV has greater in-sample average performance for this sample data, according to this result.

References

[1] Bukhari, Afnan, Farookh Khadeer Hussain, and Omar K. Hussain. "Fog node discovery and selection: A Systematic literature review." *Future Generation Computer Systems* 135 (2022): 114-128.S.

[2] Tang, Tsung-Yi, Li-Yuan Hou, and Tyng-Yeu Liang. "An IOTA-Based Service Discovery Framework for Fog Computing." *Electronics* 10, no. 7 (2021): 844.

[3] Viji Rajendran, V., and S. Swamynathan. "SD-CSR: semantic-based distributed cloud service registry in unstructured P2P networks for augmenting cloud service discovery." *Journal of Network and Systems Management* 27 (2019): 625-646.

[4] Liang, Haoran, Jun Wu, Xi Zheng, Mengshi Zhang, Jianhua Li, and Alireza Jolfaei. "Fog-based secure service discovery for internet of multimedia things: a cross-blockchain approach." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16, no. 3s (2020): 1-23.

[5] Sohofi, Ali, Aliasghar Amidian, and Yaghoub Farjami. "SocioChain: A Distributed Ledger-based Social Framework for the Internet of Things." *International Journal of Intelligent Engineering & Systems* 14, no. 5 (2021).

[6] Patil, V. C., K. A. Al-Gaadi, D. P. Biradar, and M. Rangaswamy. "Internet of things (Iot) and cloud computing for agriculture: An overview." *Proceedings of agro-informatics and precision agriculture (AIPA 2012), India* 292 (2012): 296.

[7] Gupta, Brij B., and Megha Quamara. "An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols." *Concurrency and Computation: Practice and Experience* 32, no. 21 (2020): e4946.

[8] Al-Qaseemi, Sarah A., Hajer A. Almulhim, Maria F. Almulhim, and Saqib Rasool Chaudhry. "IoT architecture challenges and issues: Lack of standardization." In *2016 Future technologies conference (FTC)*, pp. 731-738. IEEE, 2016.

[9] Mohammed, Chnar Mustafa, and Subhi RM Zeebaree. "Sufficient comparison among cloud computing services: IaaS, PaaS, and SaaS: A review." *International Journal of Science and Business* 5, no. 2 (2021): 17-30.

[10] Stojmenovic, Ivan, Sheng Wen, Xinyi Huang, and Hao Luan. "An overview of fog computing and its security issues." *Concurrency and Computation: Practice and Experience* 28, no. 10 (2016): 2991-3005.

[11] Al-Sayed, Mustafa M., Hesham A. Hassan, and Fatma A. Omara. "An intelligent cloud service discovery framework." *Future Generation Computer Systems* 106 (2020): 438-466.

[12] Park, Kisung, Youngho Park, Ashok Kumar Das, Sungjin Yu, Joonyoung Lee, and Yohan Park. "A dynamic privacy-preserving key management protocol for V2G in social internet of things." *IEEE Access* 7 (2019): 76812-76832.

[13] Kumar, Manoj, Dr Mohammad Husain, Naveen Upreti, and Deepti Gupta. "Genetic algorithm: Review and application." *Available at SSRN 3529843* (2010).

[14] Mathew, Tom V. "Genetic algorithm." *Report submitted at IIT Bombay* (2012): 53.

[15] Sarhan, Mohanad, Siamak Layeghy, Nour Moustafa, Marcus Gallagher, and Marius Portmann. "Feature extraction for machine learning-based intrusion detection in IoT networks." *Digital Communications and Networks* (2022).

[16] Park, Kisung, Youngho Park, Ashok Kumar Das, Sungjin Yu, Joonyoung Lee, and Yohan Park. "A dynamic privacy-preserving key management protocol for V2G in social internet of things." *IEEE Access* 7 (2019): 76812-76832.

[17] Sohofi, Ali, Aliasghar Amidian, and Yaghoub Farjami. "SocioChain: A Distributed Ledger-based Social Framework for the Internet of Things." *International Journal of Intelligent Engineering & Systems* 14, no. 5 (2021).

[18] Al-Qaseemi, Sarah A., Hajer A. Almulhim, Maria F. Almulhim, and Saqib Rasool Chaudhry. "IoT architecture challenges and issues: Lack of standardization." In *2016 Future technologies conference (FTC)*, pp. 731-738. IEEE, 2016.

[19] Jabraeil Jamali, Mohammad Ali, Bahareh Bahrami, Arash Heidari, Parisa Allahverdizadeh, Farhad Norouzi, Mohammad Ali Jabraeil Jamali, Bahareh Bahrami, Arash Heidari, Parisa Allahverdizadeh, and Farhad Norouzi. "IoT architecture." *Towards the Internet of Things: Architectures, Security, and Applications* (2020): 9-31.

[20] Al-Sayed, Mustafa M., Hesham A. Hassan, and Fatma A. Omara. "An intelligent cloud service discovery framework." *Future Generation Computer Systems* 106 (2020): 438-466.

[21] Ananthi, M., R. Sabitha, S. Karthik, and J. Shanthini. "FSS-SDD: fuzzy-based semantic search for secure data discovery from outsourced cloud data." *Soft Computing* 24 (2020): 12633-12642.

[22] Liang, Haoran, Jun Wu, Xi Zheng, Mengshi Zhang, Jianhua Li, and Alireza Jolfaei. "Fog-based secure service discovery for internet of multimedia things: a cross-blockchain approach." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16, no. 3s (2020): 1-23.

[23] Sharma, Geeta, and Sheetal Kalra. "A lightweight user authentication scheme for cloud-IoT based healthcare services." *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* 43 (2019): 619-636.

[24] Cai, Zhiming, Ivan Lee, Shu-Chuan Chu, and Xuehong Huang. "SimSim: a service discovery method preserving content similarity and spatial similarity in P2P mobile cloud." *Journal of Grid Computing* 17 (2019): 79-95.

[25] Xu, Chenhan, Kun Wang, Peng Li, Song Guo, Jiangtao Luo, Baoliu Ye, and Minyi Guo. "Making big data open in edges: A resource-efficient blockchain-based approach." *IEEE Transactions on Parallel and Distributed Systems* 30, no. 4 (2018): 870-882.

[26] Navimipour, Nima Jafari. "A formal approach for the specification and verification of a trustworthy human resource discovery mechanism in the expert cloud." *Expert Systems with Applications* 42, no. 15-16 (2015): 6112-6131.

[27] Parhi, Manoranjan, Sanjaya Kumar Jena, Niranjan Panda, and Binod Kumar Pattanayak. "A semantic matchmaking technique for cloud service discovery and selection using ontology based on service-oriented architecture." In *Intelligent and Cloud Computing: Proceedings of ICICC 2019, Volume 1*, pp. 31-40. Springer Singapore, 2021.

[28] Abbas, Ghulam, Amjad Mehmood, Jaime Lloret, Muhammad Summair Raza, and Muhammad Ibrahim. "FIPA-based reference architecture for efficient discovery and selection of appropriate cloud service using cloud ontology." *International Journal of Communication Systems* 33, no. 14 (2020): e4504.