An Intelligent Machine Learning Inspired Optimization Algorithm to Enhance Secured Data Transmission in IoT Cloud Ecosystem

Sreejyothsna Ankam[1], Research Scholar, Dept Of CSE,JNTUA, Ananthapuramu, India
mounika.skuphd@gmail.com

Dr.N.Sudhakar Reddy[2], Principal & Professor, Dept Of CSE,SVCE, Tirupati, India

Abstract

Traditional Cloud Computing would be unable to safely host IoT data due to its high latency as the number of IoT sensors and physical devices accommodated on the Internet grows by the day. Because of the difficulty of processing all IoT large data on Cloud facilities, there hasn't been enough research done on automating the security of all components in the IoT–Cloud ecosystem that deal with big data and real-time jobs. It's difficult, for example, to build an automatic, secure data transfer from the IoT layer to the cloud layer, which incorporates a large number of scattered devices. Addressing this issue this article presents an intelligent algorithm that deals with enhancing security aspects in IoT cloud ecosystem using butterfly optimization algorithm.

Keywords: *Security, Cloud Computing, Cloud-IoT ecosystem, Optimization*

1. Introduction

The Internet of Things (IoT) and cloud computing have grown rapidly in recent years. IoT devices for consumers are available on the market, and major cloud service providers are expanding their software stacks to include IoT services as well. In recent years, study on the safety of these smart IoT cloud systems has increased as this rising trend grows. Recent years have seen a rise in the popularity of the IoT, or "Internet of Things," which is essentially a network of everything. It makes it possible for people to get data from a variety of sources and then manipulate that data using networking technology, making it easier for them to engage with the world around them. As a result of the rapid advancements in both hardware and networking over the last decade, IoT devices have been extensively implemented. In addition, the GSMA predicts [8] that IoT devices will continue to be deployed in the future, reaching 25.2 billion devices worldwide by 2025. Aside from the Internet of Things (IoT), cloud computing has also become a new infrastructure in contemporary civilization in recent years. You may use it from any device, on any network, at any time, and from almost anywhere.

IoT devices can use the cloud as a storage, messaging, and computing backend, which allows for remote data and compute access for IoT terminal applications to be implemented. Despite cloud computing and IoT having evolved independently over the past decade or so researchers have recently integrated the two to build more powerful IoT applications. These Internet of Things (IoT) cloud services are supported by all mainstream cloud service providers at present. Developments in the IoT cloud environment are also becoming more commonplace. A good example of an IoT cloud ecosystem is the Alexa services provided by Amazon [18]. Such a solution uses Alexa's microphones to gather speech data from users and transfer it to the cloud. Once the data has been processed, the cloud will get back to Alexa. As a form of smart home hub, Alexa can also operate other Internet of Things (IoT) gadgets in a user's house. This includes things like as turning on the TV, showing a picture, and ordering meals. Using the terminal application (i.e., a mobile phone app) to talk to Alexa when away from home is an option. Cloud services are used for all of this.

IoT cloud ecosystem architecture is being used more and more in a variety of fields, including wearables, smart homes, self-driving cars, health care, and industrial equipment, but security is a major concern. Cloud-based IoT applications have been studied by researchers recently. Because all IoT

Manuscript received June 5, 2022
Manuscript revised June 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.6.14

cloud systems are connected to people and some are connected to vital infrastructures, knowing the security of these systems is crucial. A thorough knowledge of these systems and their users is essential for their protection, as well as the development of more effective methods. That's why we've put together this article, which summarizes current knowledge and proposes future research challenges in the consumer-oriented IoT cloud system area, in the hope that it serves as a reference for both practical developers and researchers who are interested in this area and calls for better solutions for IoT cloud systems by addressing existing research challenges.

2. Literature Review

Several publications have examined the security of Internet of Things (IoT) and cloud computing (cloud computing). Sicari et al. [11] examined IoT security studies and issues in the domain of IoT systems security. In addition, this report assessed common IoT system implementations. Security concerns for IoT systems, encompassing hardware, software, and networking, have been outlined by Alaba et al. [4]. A review of possible blockchain solutions for IoT security was conducted by Khan and Salah. For IoT systems, Harbi et al. looked at the security threats and the security needs. Stoyanova examined IoT data forensics, including problems, theoretical frameworks, and actual solutions. According to Khalil et al., cloud computing services include security weaknesses and possible remedies that need to be addressed. They also looked at some of the most current cloud computing security issues and solutions [14, 10]. On sensitive data in the cloud, the Domingo-Ferrer et al. assessment looked at how to protect privacy. For cross-cloud federation trust assessment, Ahmed et al. conducted a survey [3]. Cloud computing security and privacy problems have also been examined by Tabrizchi [11]. IoT cloud architecture and security elements were examined by Ammar et al. for IoT cloud integration [9]. The communication protocols for IoT, fog, and cloud integration were studied by Dizdarevic et al. [3]. With the use of program analysis tools, Celik et al. [10] investigated IoT programming platform security and privacy vulnerabilities. For cloud-based IoT applications,

Kumar et al. conducted an assessment of security risks and security procedures For cloud-based IoT applications, Almolhis also looked at some basic security challenges and current solutions. According to earlier studies, IoT and cloud system integration (i.e., IoT cloud ecosystems) for constructing smart consumer-oriented apps has just recently emerged in the market. Consumer apps have just lately begun to use IoT cloud integration, which has long been well-known in the scientific world. It's common for these consumer apps to be utilized by enormous numbers of people (e.g., millions of people). Researchers are now looking at how to better protect their interests. A new security issue has arisen, and earlier assessments haven't addressed all of it. This is because an average IoT cloud ecosystem has a far higher scale than a single IoT system or cloud application. Millions of people have smart home equipment (such as smart voice assistants) installed in their residences. It's difficult to keep track of so many devices and keep the data safe at this size. Second, the increased openness provided by an IoT cloud ecosystem opens up additional potential entry points for hackers.

On the IoT side, anybody who wants may acquire/purchase the system; on the cloud side, public HTTP GET/PUT services are used to interchange data between IoT devices and the cloud. In addition, an IoT hub may enable devices from many manufacturers, owned by various users, and integrated into the IoT cloud system as well. Protecting system security is made more difficult by the variety of methods used to access it by various devices and different users. For the third time an IoT cloud ecosystem has additional options. IoT cloud applications that are commercially available are often utilized by many distinct customers, each of whom purchases a different device but the same model.

As a result, the cloud end must distinguish between these several users, which is a difficult task for the cloud end. If a person uses the same sort of

IoT device from the same manufacturer as another user, that user does not want their data read by the other user. Fourth, more people are involved in an IoT cloud ecosystem. Besides gathering data about human activities (such as in smart home applications), it also provides a mobile app that allows users to manage Internet of Things (IoT) gadgets. It's possible that a single device may be shared and used by several people, each with their own set of habits and preferences. The degree of human engagement also creates a new point of vulnerability for assault. The absence of concrete examples in the prior evaluation made it difficult to get an intuitive and application-level grasp of genuine, deployed IoT systems (e.g., as in References [4, 9, 14, 16, 4, 7]).

3. Proposed Model:

Let's start with a simple smart home app to demonstrate the concept. Assume that the smart air conditioner and the temperature sensor are both part of the IoT system for this application. Whenever the temperature rises to 30 degrees, a smart home hub transmits the temperature data to the cloud server. The cloud server's data analysis system discovers that the current temperature is too high after getting the data. A command is sent to the smart air conditioner from the cloud server, telling it to start functioning and establish the right temperature goal.

Turning on and maintaining the desired temperature is all that the smart air conditioner does. Even if the temperature rises to 30 degrees, a person suffering from a cold may be reluctant to use the air conditioner. A smartphone app or other control terminal may be used to prevent the smart air conditioner from turning on without the user's permission. Finally, an IoT cloud ecosystem may be a self-adapting system, or it can be tampered with by human intervention. This evaluation focuses on a user-centered application. Users, manufacturers, cloud service providers, and society at large are concerned about the security of these new apps, which are being utilized by millions of people. For future consumer-oriented IoT cloud systems, it helps to evaluate their security objectively.

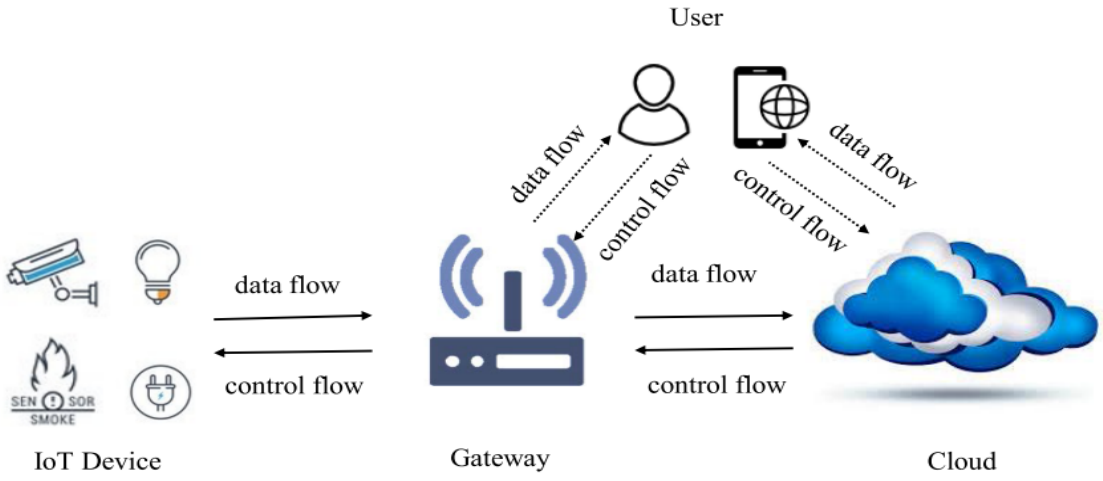

Figure 1: IoT Cloud ecosystem

Securing the gateway plays a vital role in IoT cloud ecosystem as shown in figure 1. Addressing this context this article presents an intelligent mechanism where in the attack feature extraction and classification is done at the gateway.

3.1 Feature selection using bat induced butterfly optimization (BBO)

Feature selection is a preliminary process used to improve product quality. FS is considered to be an integrated set of optimizations aimed at finding the

optimal subset of properties in the original database that accurately reflects the original data. There are two main stages in a typical FS process: (i) finding the minimum reduction and (ii) evaluating the selected characteristics. The main challenge is to find out if the best FS still exists about the properties of the original data. Providentially, FS is considered a search unit that represents a subset of the attribute at each point of the search point. For this, we applied a bat induced butterfly optimization (BBO) for selecting the optimal feature and for removing unwanted data.

The first change is that we use a certain frequency and sound instead of a different frequency g_j. In BBO, each bat is determined by its position y_j^T, velocity U_j^T. The new solutions y_j^T and velocities U_j^T at time step T are given by

$$U_j^T = U_j^{T-1} + (y_j^T - y_*)g \tag{1}$$

$$y_j^T = y_j^{T-1} + U_j^T \tag{2}$$

The global best solution is referred as y∗. In this g is equal to 0.5. To increase demographic diversity the search performance is improved by Eq. (3)

$$Y_{NEW} = y_{s1}^T + G(Y_{s2}^T - Y_{s3}^T) \tag{3}$$

where G is the mutation weight factor, while s_1, s_2, s_3 are evenly divided into random numbers. The migration process can be expressed as follows:

$$y_{j,z}^{T+1} = y_{s1,z}^T \tag{4}$$

where $y_{j,z}^{T+1}$ zth denotes an element of y_j at generation T+1 it gives the position of King Butterfly *i*. Similarly, $y_{s1,z}^T$ indicates the *z*th newly formed stage of the monarch butterfly s_1. *T* is the number of the current generation. Monarch butterfly s_1 is approximately selected from the sub-population. Here, *s* can be calculated as

$$s = Rand * Peri \tag{5}$$

Peri indicates immigration period. Rand is a random number obtained as a result of consolidated distribution. Or rather, if s>q, the kth element in the butterfly is the newly formed king

$$y_{j,z}^{T+1} = y_{s2,z}^T \tag{6}$$

where $y_{j,z}^{T+1}$ the newly formed phase of the monarch butterfly is the return element s_2. Monarch butterfly r_2 is approximately selected from the sub-population. If the generated probable number q is less than or equal to q for all components of the monarch butterfly, it can be updated as follows:

$$y_{j,z}^{T+1} = y_{Best,z}^T \tag{7}$$

where $y_{j,z}^{T+1}$ zth denotes an element of y_i at generation T+1 gives the position of King Butterfly j. Similarly, $y_{Best,z}^T$ zth denotes an element of y_{Best} that is Best King Butterfly in Land 1 and Land 2. T is the number of the current generation. Or rather, if larger than the Rand P, it can be upgraded

$$y_{j,z}^{T+1} = y_{s3,z}^T \tag{8}$$

where $y_{s3,z}^T$ and zth denotes an element of y_{s3}. In this case, if it is Rand >BAR, it can be updated as follows

$$y_{i,z}^{T+1} = y_{i,z}^{T+1} + \alpha \times (dy_z - 0.5) \tag{9}$$

where it indicates butterfly adjustment speed. dy is the according to the monarch butterfly i Levy calculate this by flight.

$$dy = Levy(y_i^T) \tag{10}$$

In Eq. (9), α is the expectation factor is given as Eq. (11)

$$\alpha = R_{Max} / T^2 \tag{11}$$

The working function of algorithm 1 represents the function of the BBO.

Algorithm 1 Optimal feature selection using bat induced butterfly optimization

Input : Velocity
Output : Weight factor

1 Initialize the parameters
2 Compute the new solutions
$U_j^T = U_j^{T-1} + (y_j^T - y_*)g$
3 Improve the performance using
$Y_{NEW} = y_{s1}^T + G(Y_{s2}^T - Y_{s3}^T)$
4 Compute the migration process using
$y_{j,z}^{T+1} = y_{s1,z}^T$
5 Determine the new population using
$y_{j,z}^{T+1} = y_{s2,z}^T$
6 Upgrade the position of the butterfly
7 Calculate the levy flight using
$dy = Levy(y_i^T)$
8 End

3.2 Classification using Random Forest algorithm

As early as 2001, Bremen devised the first random forest algorithm. In order to monitor the random forest, a classification method employs decision trees. Data mining techniques like the decision tree algorithm are quite common. Data attributes and current data are used to form a judgment on the class or category in a decision tree, which is how the classification is made. CART is a binary tree algorithm that is part of the decision tree method. For each stage of the random forest, there are four CART trees [33]. In the training phase (D), the Bootstrap sampling approach is used to choose a subset of the training samples (D1, D2,..., Dk). Finally, the K decision tree will be built. According to minimal purity requirements, we shall appoint just the best special from among all candidate M branches at node N of the classification tree. As a result, trees will mature. The third phase is a rerun of the previous one. K Created decision tree. Four. The well-established crucial trees create an asymmetrical forest. There's still a long way to go until the final sample is selected in the random forest.

All of the characteristics are assessed using a multi-class SVM classifier for each individual group of features. An initial set of data are used to train this category, but only those attributes that are relevant to the task at hand are included. Filtered experimental data sets are then used to test this hypothesis.' It is necessary to teach a different categorization for every category (one-vs-all approach). Lastly, the feature subset is evaluated based on how well it can classify the experimental data using a variety of different support vector machines. Attributes are encoded by using binary strings along the number of attributes, where a zero indicates that an attribute is not chosen, and one indicates that it has been selected in an attribute subset.

It's a kind of meta-heuristic algorithm that combines new approaches like local search with more traditional search engines like evolutionary algorithms. Memetic algorithm Improve the fundamental search algorithm's performance, such as lowering the time it takes for an ideal answer [22]. In most cases, evolutionary algorithms are developed to cover the whole of the search domain. On the other hand, a local neighborhood search employs an evolutionary algorithm to locate better solutions. An algorithm's execution outcomes will be greatly influenced by its choice of generation operators, as well as the algorithm's type and local search strategy. A local search method is thus utilized in this study to determine the solution's closeness after it has been received by the algorithm that estimates distributions. Algorithm picks the closest feasible neighboring subset to find the most acceptable one. Finally, it replaces the present solution with the best one discovered.

4. Performance Analysis

4.1 Data Set

There are 43 fields in each record in the NSL-KDD dataset. Attribute 41 is a closed behavior field that indicates the usual behavior or kind of intrusion, and the last field reflects the difficulty of detecting intrusion. The label column includes five classifications: one for conventional attacks and three for intrusions: DoS, U2R, R2L, and Prob. By overloading the target computer with connection requests, a denial of service attack causes the server to incur significant cost, preventing it from responding to normal network traffic. User attacks against root are carried out using a regular user account in attempt to achieve root access by exploiting a system vulnerability. In external penetration, the attacker has the capacity to transmit packets to a computer, but it does not have an ID on the machine and cannot access the system like a user. As part of the intrusive scanning infiltration process, the system is scanned to identify any potential vulnerabilities or attacks that might be exploited later. These vulnerabilities may be exploited to carry out an attack on a system. Numerical and textual information is categorized into three groups in this dataset: basic; content and traffic.

An IP connection's TCP/IP functions are among the most basic. These qualities slow down the detection of intrusions. These elements include the length of the connection, the protocol and service utilized, and the number of bytes exchanged on a connection. Informational features: These attacks do not follow a pattern of sequential abnormal recurrence, in contrast to many other types of service-preventing and scanning assaults. External intrusion and penetration of the root system occur because network data packets are contained in the data section and only have a single connection, as opposed to the service-blocking and scanning assaults that have several connections to hosts over a short period of time Features that can look for infiltration behavior in the packet data portion, including the amount of unsuccessful attempts, are necessary for detecting this form of assault. They're known as content attributes. Examples of these characteristics include the amount of activities conducted on a connection, the number of unsuccessful logins on a connection, and the user's ability to access the system as an administrator. When it comes to traffic characteristics, you'll find that they're broken down into two categories: Time-based connections are those that have had the same service and host in the last two seconds as the present connection; a second kind of time-based connection is one that is used to analyze assaults that take place over a longer time period. They're termed machine-based features since they calculate the proportion of prior connections to present connections with the same service and host. CIDDS-001, KDD99, and VIRUS TOTAL datasets are also used for cross-validation of the methods.

4.2 Simulation Results

NSL-KDD database findings are shown in this portion of the article. There are five feature selection strategies that are evaluated using the support vector machine in populations of 50, 100, and 150. The leading selection and backward selection algorithms are population-free, meaning that population expansion has no influence on their performance. The proposed algorithm beat the distribution estimation method while populations were smaller, but the accuracy gap has narrowed as populations have grown. Distribution estimation technique and local search have also substantially enhanced its performance in small populations. The accuracy of the proposed mechanism is consistent in all the levels of population as shown in figure 2.

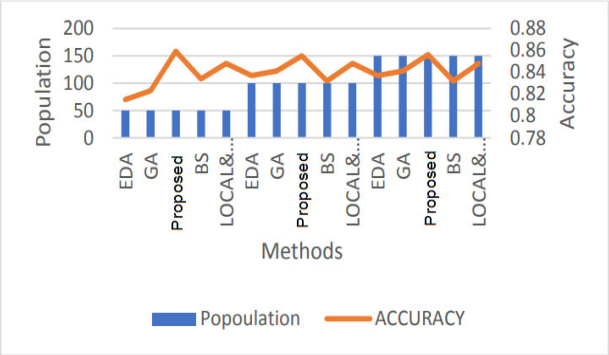

Figure 2: Accuracy evaluation with other algorithms

When a database contains packets that are categorized into five distinct categories, the resulting classification accuracy is shown in above figure utilizing various feature selection techniques and populations of various sizes. The detection accuracy of influences with a limited number of examples in the training database is greatly lowered, resulting in a reduction in overall detection accuracy.

5. Conclusion:

Classification is fundamental to infiltration detection, and feature selection is one of the concerns addressed. To minimize detection time and cost and improve classifier performance, feature selection may be used to huge datasets. As a function of algorithmic fit, this research examined the performance of genetic attribute selection techniques, distribution estimation, hybrid distribution estimation with local search, leading selection, and backward selection with SVM classification. It was thus necessary to conduct a general experiment using two benchmark datasets (NSL-KDD and VIRUS TOTAL) and four state-of the art machine learning classifiers (KNN-RF; PSO; SVM GA; and GA) to examine the impact of the four feature assessment metrics on an IDS's classification accuracy. All classifiers showed comparable results, but Proposed mechanism had the best detection accuracy with all feature assessment metrics at optimum parameter values.

6. References

[1] Sniderman, B.; Mahto, M.; Cotteleer, M.J. Industry 4.0 and Manufacturing Ecosystems; Deloitte University Press: London, UK, 2016; pp. 1–23.

[2] Corotinschi, G.; G ˘aitan, V.G. Enabling IoT connectivity for Modbus networks by using IoT edge gateways. In Proceedings of the 2018 International Conference on Development and Application Systems (DAS), Suceava, Romania, 24–26 May 2018; pp. 175–179.

[3] Geissbauer, R.; Schrauf, S.K.V. Industry 4.0-Opportunities and Challanges of the Industrial Internet. Available online: https: //www.strategyand.pwc.com/gx/en/insights/2015 /industrial-internet.html (accessed on 2 February 2021).

[4] Frankó, A.; Vida, G.; Varga, P. Reliable Identification Schemes for Asset and Production Tracking in Industry 4.0. Sensors 2020, 20, 3709. [CrossRef]

[5] Massaro, A.; Galiano, A. Re-engineering process in a food factory: An overview of technologies and approaches for the design of pasta production processes. Prod. Manuf. Res. 2020, 8, 80–100. [CrossRef]

[6] Weerasiri, D.; Barukh, M.C.; Benatallah, B.; Sheng, Q.Z.; Ranjan, R. A Taxonomy and Survey of Cloud Resource Orchestration Techniques. ACM Comput. Surv. 2017, 50, 1–41. [CrossRef]

[7] Maiti, P.; Shukla, J.; Sahoo, B.; Turuk, A.K. QoS-aware fog nodes placement. In Proceedings of the 2018 4th International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, India, 15–17 March 2018; pp. 1–6. [CrossRef]

[8] Groover, M. Fundamentals of Modern Manufacturing: Materials, Processes, and Systems; John Wiley & Sons, Inc: Hoboken, NJ, USA, 2020.

[9] Deshmukh, U.; More, S.A. Fog Computing: New Approach in the World of Cloud Computing. FInt. J. Innov. Res. Comput.Commun. Eng. 2016, 4, 16310–16316. [CrossRef]

[10] Luan, T.H.; Gao, L.; Li, Z.; Xiang, Y.; Wei, G.; Sun, L. Fog computing: Focusing on mobile users at the edge. arXiv 2015, arXiv:1502.01815

[11] Puliafito, C.; Vallati, C.; Mingozzi, E.; Merlino, G.; Longo, F.; Puliafito, A. Container Migration in the Fog: A Performance Evaluation. Sensors 2019, 19, 1488. [CrossRef]

[12] Gil, D.; Ferrández, A.; Mora-Mora, H.; Peral, J. Internet of things: A review of surveys based on context aware intelligent services. Sensors 2016, 16, 1069. [CrossRef]

[13] Perera, C.; Qin, Y.; Estrella, J.C.; Reiff-Marganiec, S.; Vasilakos, A.V. Fog Computing for Sustainable Smart Cities. ACM Comput. Surv. 2017, 50, 1–44. [CrossRef]

[14] Naha, R.K.; Garg, S.; Georgakopoulos, D.; Jayaraman, P.P.; Gao, L.; Xiang, Y.; Ranjan, R. Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions. IEEE Access 2018, 4, 1–31. [CrossRef]

[15] Maag, B.; Zhou, Z.; Thiele, L. A survey on sensor calibration in air pollution monitoring deployments. IEEE Internet Things J. 2018, 5, 1–15. [CrossRef]
[16] Mukherjee, M.; Shu, L.; Wang, D. Survey of fog computing: Fundamental, network applications, and research challenges. IEEE Commun. Surv. Tutor. 2018, 20, 1–30. [CrossRef]
[17] Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. 2015, 17, 2347–2376. [CrossRef]
[18] Yassein, M.B.; Shatnawi, M.Q.; Aljwarneh, S.; Al-Hatmi, R. Internet of Things: Survey and open issues of MQTT protocol. In Proceedings of the 2017 International Conference on Engineering & MIS (ICEMIS), Monastir, Tunisia, 8–10 May 2017. [CrossRef]
[19] Maheswari, K.; Bhanu, S.S.; Nickolas, S. A Survey on Data Integrity Checking and Enhancing Security for Cloud to FogComputing. In Proceedings of the IEEE Xplore, Bangalore, India, 5–7 March 2020; pp. 121–127.