Research Reports on Computer Science
https://ojs.wiserpub.com/index.php/RRCS/

UNIVERSAL WISER PUBLISHER

Research Article in Special Issue: Selected Papers from the 4th International Conference on Machine Learning, Image Processing, Network Security and Data Sciences (MIND-2022)

An Efficient Approach for Secured Data Transmission Between IoT and Cloud

Shatakshi Kokate*, Urmila Shrawankar

Department of Computer Science and Engineering, G. H. Raisoni College of Engineering, Nagpur, India
E-mail: shatakshi.kokate@gmail.com

Received: 8 March 2023; **Accepted:** 28 March 2023

Abstract: The Internet of Things (IoT) network generates a lot of data and cloud servers collect that data. The server then analyzes the collected data and based on the findings, provides appropriate intelligent services to users as a result. If there is any faulty data while the server analyzes the collected data, distorted results will be created. The data captured from IoT contains lots of heterogeneous as well as suspicious data, so cleaning, filtering, and clustering of it must be done before sending it to the server, otherwise it will unnecessarily create overhead on the server. The proposed system consists of a filtering and clustering mechanism for the data collected from IoT devices so that integrated data is transferred to the cloud server which will reduce its computational load. In the proposed system, the fog computing layer is used as an interface between IoT and cloud computing layer where data filtering and clustering take place to reduce network traffic and latency. The ultimate aim is to provide security for data transmission between IoT and the cloud.

***Keywords*:** Internet of Things (IoT), fog computing, cloud, data filtering, noisy data, data classification, k-nearest neighbor, complement naive Bayes, accuracy

1. Introduction

A vast number of small objects known as sensors make up the Internet of Things (IoT) network. To deliver intelligent services, the sensors are connected to an access network. This system is made using IoT and the cloud network. In the IoT access network, data is generated by sensors on devices. This data is further transferred to the cloud server which takes decisions based on data analysis and distributes the results to actuators in the access network. The sensor nodes in the networks have limited computational capability and also have limited energy resources and memory. IoT devices can collect unexpected sensory data due to limited resources. The server may make an inaccurate choice and produce incorrect results after analyzing this faulty data which will reduce service efficiency. Due to resource constraints, many IoT devices are also prone to failure. When a network device malfunction, it produces inaccurate or unreliable data frequently. After analyzing the faulty input, the server produces misleading results. Furthermore, extra data which is called suspicious data with inaccurate information might be injected into the network. This data can also

DOI: https://doi.org/10.37256/rrcs.2320232628

impact server decisions, and a poor decision will give incorrect results to the end users. Valid data produced from the correct objects should be used for analysis. As a result, the server's data integrity must be ensured. Figure 1 illustrates the steps involved in achieving data integration. To maintain data integrity on the server, there is a need to eliminate faulty/unreliable data from the analysis. It should be cleaned before reaching the cloud server to decrease the load on it. The data is pre-processed to remove or decrease noise through the use of smoothing techniques. Faulty data can be detected using an intrusion detection system. There is also a need for a data filtering system to collect only normal data called non-suspicious data. As a result, faulty data is avoided from analysis by the server. So that, the server makes the correct decision and utilizes less energy for computation.

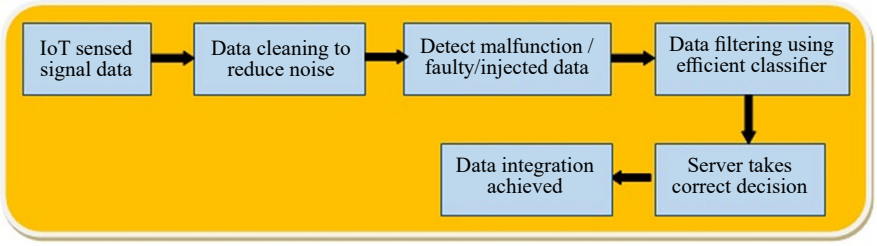

Figure 1. Steps involved in achieving data integration

If data filtering is performed at the cloud level, a large number of data is transferred to the cloud, increasing network traffic. Data loss and illegal access by outside intruders are both caused by increased network traffic. The solution is to filter data at the IoT device level. Resource limits may be a potential hurdle for the volume, velocity, and variety of IoT data if data is categorized at the device level. If data classification is performed at the fog computing layer which is present between IoT and cloud systems can be a better choice. Fog computing carries out communication, storage, and computation on devices that are close to users [1].

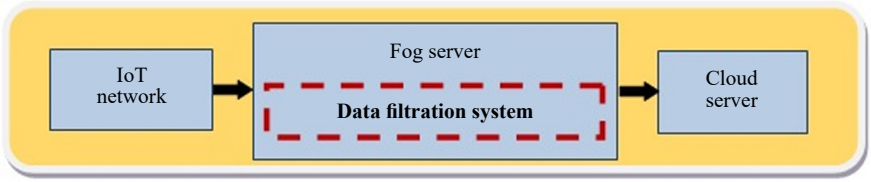

Figure 2. Data filtering system at fog level

Figure 2 shows that data filtering takes place at the fog layer which is present between IoT and the cloud layer. Fog computing increases efficiency for massive IoT applications by processing data locally instead of remotely on a cloud server. In terms of early intervention, this guarantees that responses are sent to the user with low latency [2]. To support extensive IoT applications, fog computing offers decentralized cloud architecture by extending network, repository, and computing capabilities to the network's edge. It is typically situated on the local network and is close to the system's nodes. On the other hand, a cloud is a server or data center that can be accessed from anywhere online. Data analysis, processing, and filtering are all made easier by fog computing, which also improves security for critical data. Fog computing provides better privacy and security, uses less network bandwidth, which lowers operating costs, and can resist severe environments in places like on roads, in cars, underwater, on factory floors, etc. Cloud computing technology handles the massive amounts of data generated by IoT devices and makes resources available to them as needed. Additionally, this technology offers low-cost services, powerful processing, versatility, improved performance, and openness for device accessibility.

When data classification is done at the fog level, only relevant data is sent to the cloud, resulting in lower network traffic and lower latency. Reduction in network traffic leads to achieving data security. When the server analyzes the

acquired data, the data integrity leads to a reduction in the server computing burden. When the server processes the data, reducing the computational load might result in lower energy consumption.

2. Related work

IoT represents a new notion for the Internet and smart data. Preparing and processing data are two major obstacles for researchers working with IoT. Many significant and enlightening discoveries regarding data features have been found after reviewing the real-world perspective of how IoT data is examined by various authors. A literature review aims to find out the best system for processing IoT data and choosing the best data filtering technique which will filter out suspicious data from IoT. Table 1 shows the techniques and mechanisms used by various researchers in their research.

Table 1. Techniques and mechanisms used by various researchers

Technique	Reference	Mechanism / Algorithm used	Observations
Privacy and security	[1, 3-5]	Fog computing	Fog computing is a better option to overcome the limits of the cloud computing paradigm
Data classification	[6]	Asynchronous Altering Direction Method of Multipliers (ADMM) algorithms	Designed a cross-layer optimization problem for optimal energy consumption and communication
	[7]	Boosted Trees Classifiers (BTC)	Investigate which factors consistently contributed to prediction accuracy
	[8]	Complement Naive Bayesian	Increase computational time and accuracy performance
	[9]	Fog computing-based Content-aware filtering approach for Security Services (FCSS)	Context-aware filtering of the data at the fog computing level
	[10]	Machine learning algorithms	Context-aware data filtering systems for smartphone users
Clustering	[5]	K-Nearest-Neighbors (KNN) algorithm and KNN with Run Length Encoding (RLE)	Filter out IoT data at the fog level
	[11]	KNN	Filter the IoT data and automatically generate the value of k
Data lost recovery	[12]	Reduced Variable Neighborhood Search (RVNS)	Automatically recover lost or erroneous data

After reviewing various research papers following limitations are observed:

- In an IoT cloud system, sensors connected to an IoT network generate a large number of data. When this data is transferred to the cloud, it will increase network traffic and latency.
- Malicious agents can easily attack such a network.
- Data filtration at the cloud level will increase the load on the cloud system.
- If a server receives faulty data, it may make an inaccurate choice and produce incorrect results which will reduce service efficiency.
- To ensure data integrity there should be a separate process for data cleaning; suspicious data detection and it should be further verified by event detection.

The solution to the aforementioned problem is whatever data is generated by IoT devices must be filtered at the network's edge, which will boost bandwidth by transferring only relevant data from IoT to the cloud and reduce latency. Fog is located in the local area network and offers a decentralized environment. Fog computing lowers latency as only summarized information is delivered to the cloud. Fog is a much better solution than cloud because it has a faster reaction time and can work in a weak network. As a result, fog computing is the ideal option.

There should be a mechanism that can deliver only relevant data to the cloud. Data filtering is the process of identifying potentially valuable and relevant patterns in large data sets. The classification and clustering techniques can be used to filter data. The purpose of data filtering is to extract meaningful information from large data. Data filtering systems classify the data collected from IoT devices. The input to these systems is training data sets. Naive Bayes (NB), KNN, K-Means, Random Forest, Support Virtual Machine, and other common machine learning algorithms are currently being used by many researchers for analyzing and making decisions on IoT-generated smart data.

The purposes and capacities of these algorithms for obtaining and processing data vary depending upon the input. In the case of conditional independence, complement naive Bayes (CNB) performs well and takes less time as compared to other machine learning algorithms. To analyze the correlation among comparable data, the KNN algorithm uses a relatively straightforward methodology.

3. Implementation

3.1 *System model*

Figure 3 shows the proposed three-tier architecture which consists of IoT, fog computing, and cloud computing layers. The services provided by IoT are based on data which is collected from IoT devices. The fog computing layer receives the data from IoT devices. At this layer first filtration of data takes place. The fog layer helps to manage the data transmitted to the cloud layer and pulls useful information for intelligent services.

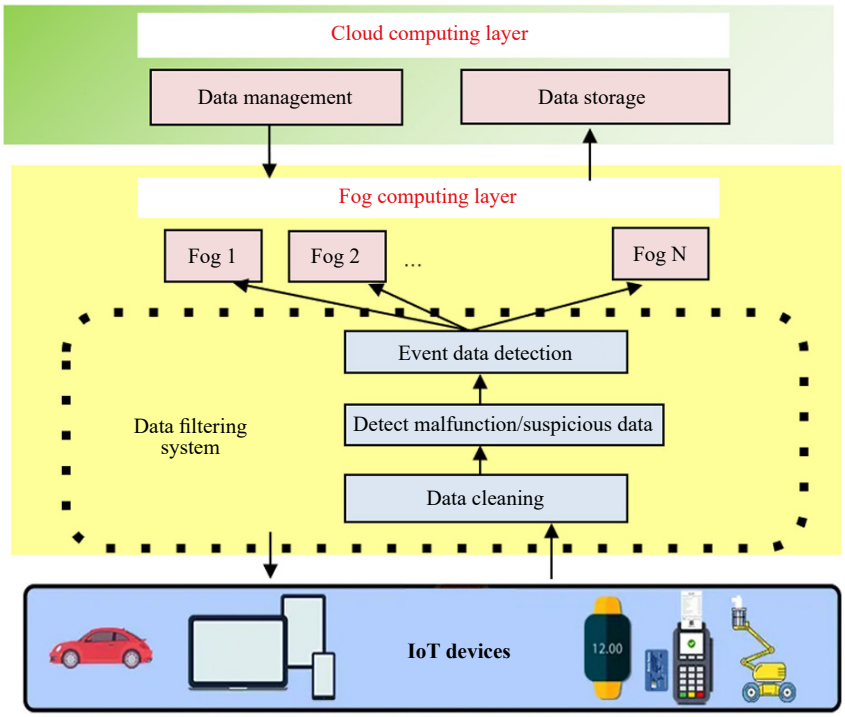

Figure 3. Proposed system model

The fog layer receives enormous amounts of real-time data from IoT devices, which is then dispersed to numerous devices connected in this layer. Fog computing offers constrained network, storage, and compute services in addition to logical intelligence and data tampering for data centers. The data gathered by IoT devices is analyzed and aggregated by the fog computing layer. Data cleaning, removal of suspicious data, and finding event data are done at this layer and then data is sent to the cloud server. The server present at the cloud layer receives the data, saves the data, and analyzes it.

3.2 *Data filtering system*

The data filtering system is shown in Figure 4. Data collected from IoT devices is routed through the system's data queue. The data handler reads the data present in front of the queue. Different types of data are produced by a variety of IoT devices. The data handler is therefore required before the detecting function. It transforms the data into the appropriate data format for the detection function's learning. The detection function then employs a CNB classifier to predict data attributes using previously saved training samples. If the data is predicted as regular data then it will be forwarded to the filtering function. The data with error values are considered malfunctioning data whereas the data with valid values are considered event data. The filtering function uses the KNN algorithm to determine if the information is event data or not. The detection and filtering functions are the two main features of the system. The detection function's analytical results are used in the filtering function to make decisions.

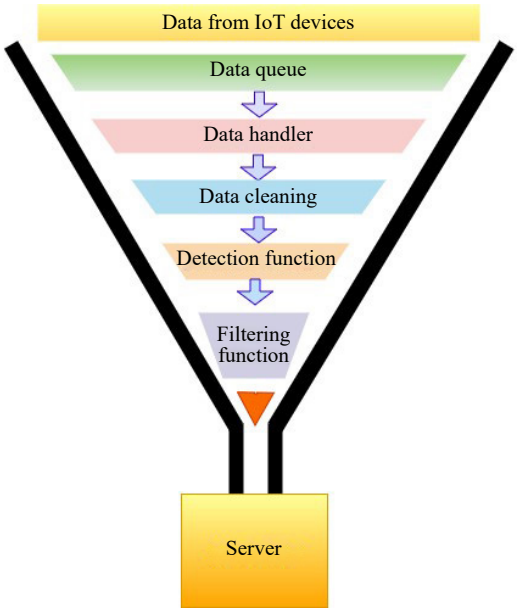

Figure 4. Data filtering system

Following are the steps involved in removing data distortion and achieving data integrity.

Data cleaning using noise reduction. Noise data means irrelevant, useless, meaningless, or corrupt data. Machines cannot correctly understand and analyze data containing unstructured text. The outcomes of any data analysis can be negatively impacted by noisy data, which also unnecessarily increases the amount of storage space needed.

The focus of data cleaning approach is on detecting and removing noise caused by a poor data collection procedure. The most powerful signal denoising filtering technique is Empirical Mode Decomposition (EMD). Using EMD, a complex and multiscale signal can be decomposed adaptively into a set of Intrinsic Mode Functions (IMFs). IMFs are a collection of finite-number zero-mean oscillating components. The instantaneous frequency of the IMFs is calculated using the Hilbert Huang Transform, an analytical signal processing technique. If a signal has an equal number (or differs by one) of extrema and zero crossovers, as well as a zero mean in both the upper and lower envelopes then it is considered IMF. The IMF is deconstructed from the raw sensor signal. The filtered signal z is constructed using the following formula to remove the IMF components

$$z = \sum_{j=2}^{T} IMF(j) \tag{1}$$

where T is the total number of IMFs. The high-frequency noise, which is represented by the first IMF of sensor signals, is removed here. The signal's significant qualities are preserved while the high frequencies are filtered out.

Suspicious data detection. In a network during data transmission, additional data with wrong information which is called suspicious data may be injected into normal data. After analyzing such data, the server gives the wrong results. If suspicious data is separated from regular data, the computation load of the server will decrease. Data filtering system uses learning techniques to identify suspicious data and intrusion data. To detect suspicious data among the incoming data, the suggested system uses CNB [8] classifier. Working with unbalanced data sets is a specialty of CNB. Instead of calculating the likelihood that an item belongs to a certain class, CNB calculates the likelihood that the item belongs to all classes.

The 'naive' part of the name comes from the fact that the predictor variables are assumed to be independent of one another. In other ways, the presence of one feature in a data set has nothing to do with the presence of any other feature. They do so by computing the 'posterior' probability of a certain event. The detection function predicts the suspicious data using posterior probabilities which are calculated from a priori probabilities. The detection function is denoted as

$$P(c|x) = \frac{P(x|c)\,P(c)}{P(x)} \tag{2}$$

$P(c|x) = P(x_1|c) \times P(x_2|c) \times ..... \times P(x_n|c) \times P(c)$

where

$P(c|x)$ is the posterior probability of class (target) given-predictor (attribute).

$P(c)$ is the prior probability of class.

$P(x|c)$ is the likelihood which is the probability of the predictor given class.

$P(x)$ is the prior probability of the predictor.

Event data detection. The event data is the data having valid values. To find out event data among incoming data, the KNN is used. The idea of the KNN algorithm is to find a k-long list of samples that are close to a sample that's to classify. KNN is best for scaling data and handling missing values. KNN is learning by analogy method that contrasts similar training and test data sets. There are n properties that describe these tuples. In n-dimensional space, all training tuples are stored. In KNN, the test tuple for classification is provided. This approach looks for k training tuples that are most similar to the test tuple; these k tuples are known as nearest neighbors.

The same event is detected by numerous IoT devices. Here, the KNN technique is used to analyze the correlation among comparable data. To do correlation testing, the filtering function uses the algorithm to determine the Euclidean distance of the data characteristics.

3.3 *Process flow*

Figure 5 represents the data flow of the filtering function. The data cleaning step is used to remove noisy data i.e., data with null values and corrupted or malfunctioning data by using the intrusion detection technique. CNB algorithm is used to detect suspicious data among incoming data. Suspicious data means the data that may be injected by the outside intruder. Then KNN algorithm is used to detect event data among collected data. The data is detected as event data if its value is within the valid range.

4. Evaluation matrix

The evaluation metrics are Time and Accuracy which are described as follows:

Time: Time spent developing the model and making predictions.

Accuracy: It is a ratio of accurately anticipated observations to the total observations.

$$\text{Accuracy} = \frac{TP + TN}{\text{Total tuples in test data set}}$$

where TP (True Positive): is the number of correct predictions that the occurrence is positive,

TN (True Negative): is the number of correct predictions that the occurrence is negative.

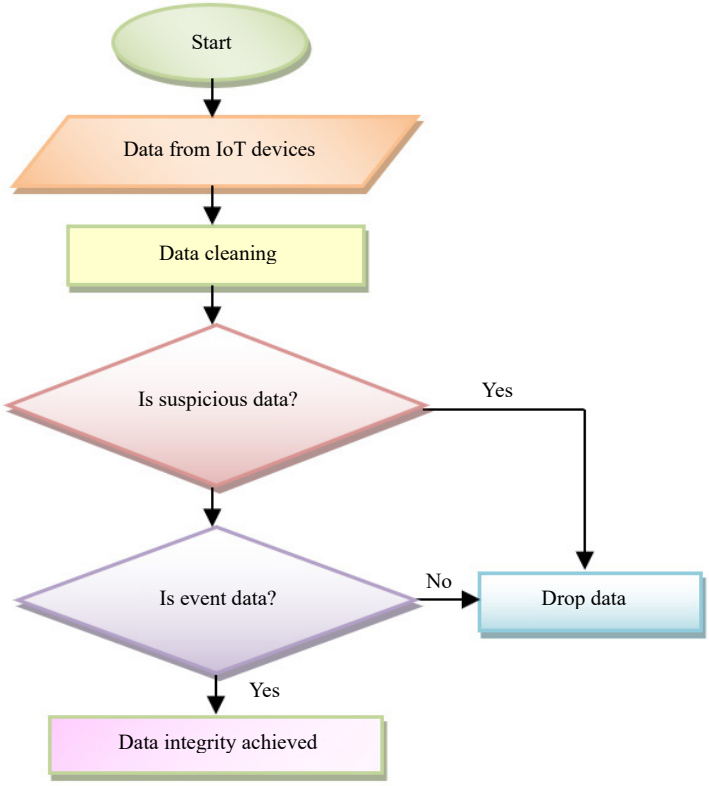

Figure 5. Flow chart of data filtering

5. Results

The suggested system is simulated using the iFogsim2 simulator. Figure 6 shows the topology used for simulating the proposed system which consists of sensors, actuators, fog nodes, and the cloud data center.

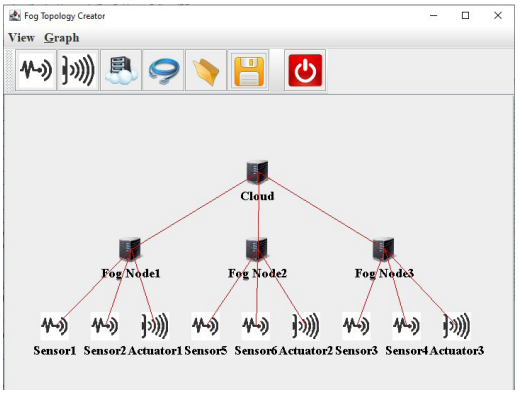

Figure 6. iFogSim2 graphical user interface (GUI) for building a network topology

The proposed system uses three steps of data filtering such as data cleaning, detecting suspicious data, and detecting event data. The data cleaning technique is used to remove noise caused by a poor data collection procedure. The detection function of detect suspicious data technique determines which inbound data traffic contains the suspicious material. The event data detection technique is used to investigate the correlation among comparable data.

Figure 7 shows the incoming data to the cloud server is 32,444 without data filtering system and using a data filtering system, it is 29,691 when simulation time is 4 sec. The detection function shows 91% accuracy. In other words, by filtering the data gathered from IoT devices, the computing load on the cloud server is decreased.

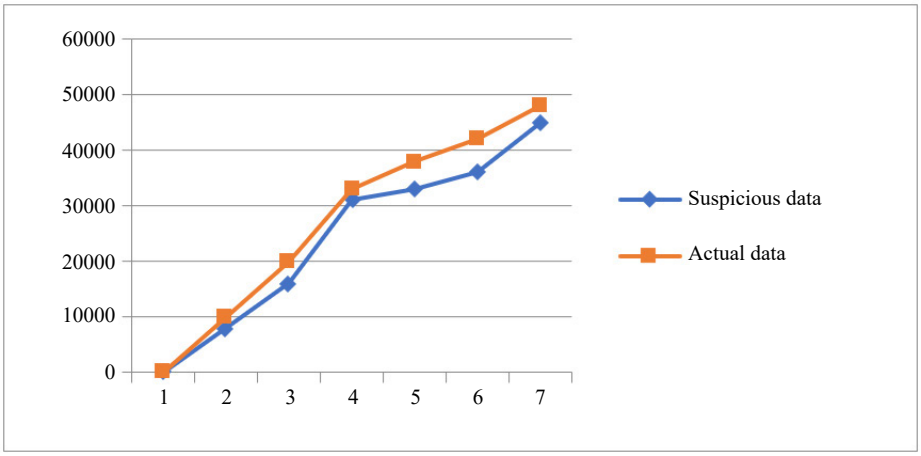

Figure 7. Amount of data traffic to cloud server over time

The second most improved thing is implanting the whole filtering system at the fog layer instead of implanting it at the cloud level. It reduces the data traffic and thus average execution delay of the proposed system significantly decreases as compared to the cloud-centric approach.

The performance of the NB, CNB, and suggested systems is shown in Table 2. There are 89 cases utilized for training. The data for ten tests were chosen using a random sample with a replacement method. The system's output is determined by characteristics such as correct classification, wrong classification, and accuracy.

Table 2. Accuracy performance of proposed system compared with other algorithms

Ex. No.	NB			CNB			Proposed system		
	CC	ICC	Acc.	CC	ICC	Acc.	CC	ICC	Acc.
1	64	25	71.9	74	15	83.2	84	5	94.38
2	67	22	75.2	84	5	94.4	86	3	96.63
3	69	20	77.5	89	0	100	89	0	100
4	69	20	77.5	89	0	100	89	0	100
5	69	20	77.5	89	0	100	89	0	100
6	66	23	74.1	89	0	100	89	0	100
7	67	22	75.2	89	0	100	89	0	100
8	70	19	78.7	89	0	100	89	0	100
9	77	12	86.5	89	0	100	89	0	100
10	78	11	87.6	89	0	100	89	0	100
			78.18			97.736			99.10

Note: CC – Correctly Classify, ICC – Incorrectly Classify, Acc. – Accuracy

The above simulation results clearly state that the proposed system gives an improved and more accurate

classification of IoT data as compared to the existing system developed using either NB or CNB. It is observed that the data loss and tampering data are reduced considerably. The ultimate aim of secured transmission of data between IoT and cloud is achieved.

6. Related work

The filtering technique is used to enhance IoT data integrity. Most of the research done so far regarding data integrity and the filtering of IoT data transmission uses data cleaning, malfunctioning data detection, and event data detection techniques separately. The combination of three techniques in one gives prominent and tremendous results regarding data filtering. In the proposed system, the data filtering takes place using three steps which give better results than the existing systems. As filtering takes place at the fog computing layer, now the cloud will receive the relevant data. It reduces the data traffic between IoT and the cloud. As data traffic is reduced, the outside intruder will not attack the network, and security is provided to the data transmitted from IoT to the cloud. The proposed system will prove very useful in many applications such as augmented reality, healthcare, agriculture, smart utility services, caching and processing, gaming, decentralized smart building controls, and agriculture.

7. Conclusion

IoT devices generate a vast amount of data. Processing such a vast amount of data, it becomes risky to communicate IoT devices with the cloud and vice versa. Traditional cloud servers filter data in a centralized fashion, resulting in a single point of failure. Furthermore, outside intruders can target an IoT network, resulting in data tampering. Unreliable and unauthenticated data results from a high number of heterogeneous IoT. While various algorithms have been applied to data classification research, it is observed that some algorithms gave better results than the other algorithms. This paper suggested a better filtering technique using three steps of data filtering such as data cleaning; detecting suspicious data, and event data detection at the fog computing layer to increase the results of existing data filtering systems. The developed system increases bandwidth and reduces the latency as data filtering takes place at the fog computing layer instead of the cloud computing layer and also provides the ultimate solution for the secure transmission of data between IoT and the cloud. In the future, the work will be expanded to include the implementation of the system for a variety of applications

Conflict of interest

The authors declare that there is no conflict of interest.

References

[1] Neware R, Shrawankar U. Fog computing architecture, applications and security issues. *International Journal of Fog Computing (IJFC)*. 2020; 3(1): 75-105. http://doi.org/10.4018/IJFC.2020010105

[2] Nemade B, Shah D. An efficient IoT based prediction system for classification of water using novel adaptive incremental learning framework. *Journal of King Saud University-Computer and Information Sciences*. 2022; 34(8): 5121-5131. https://doi.org/10.1016/j.jksuci.2022.01.009

[3] Rani R, Kashyap V, Khurana M. Role of IoT-cloud ecosystem in smart cities: Review and challenges. *Materials Today: Proceedings*. 2022; 49: 2994-2998. https://doi.org/10.1016/j.matpr.2020.10.054

[4] Bittencourt L, Immich R, Sakellariou R, Fonseca N, Madeira E, Curado M, et al. The Internet of Things, fog and cloud continuum: Integration and challenges. *Internet of Things*. 2018; 3-4:134-155. https://doi.org/10.1016/j.iot.2018.09.005

[5] Ribeiro FM, Prati R, Bianchi R, Kamienski C. A nearest neighbors based data filter for fog computing in IoT smart agriculture. In: *2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*.

Trento, Italy: IEEE; 2020. p.63-67. https://doi.org/10.1109/MetroAgriFor50201.2020.9277661
[6] Xenakis A, Karageorgos A, Lallas E, Chis AE, González-Vélez H. Towards distributed IoT/cloud based fault detection and maintenance in industrial automation. *Procedia Computer Science*. 2019; 151: 683-690. https://doi.org/10.1016/j.procs.2019.04.091
[7] Goldstein A, Fink L, Meitin A, Bohadana S, Lutenberg O, Ravid G. Applying machine learning on sensor data for irrigation recommendations: revealing the agronomist's tacit knowledge. *Precision Agriculture*. 2018; 19: 421-444. https://doi.org/10.1007/s11119-017-9527-4
[8] Anagaw A, Chang YL. A new complement naïve Bayesian approach for biomedical data classification. *Journal of Ambient Intelligence and Humanized Computing*. 2019; 10: 3889-3897. https://doi.org/10.1007/s12652-018-1160-1
[9] Wu J, Dong M, Ota K, Li J, Guan Z. FCSS: Fog-computing-based content-aware filtering for security services in information-centric social networks. *IEEE Transactions on Emerging Topics in Computing*. 2019; 7(4): 553-564. https://doi.org/10.1109/TETC.2017.2747158
[10] Sarker IH, Kayes ASM, Watters P. Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. *Journal of Big Data.* 2019; 6(1): 57. https://doi.org/10.1186/s40537-019-0219-y
[11] Guo G, Wang H, Bell D, Bi Y, Greer K. KNN model-based approach in classification. In: Meersman R, Tari Z, Schmidt DC. (eds.) *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003*. Lecture Notes in Computer Science, vol 2888. Berlin: Springer; 2003. p.986-996. https://doi.org/10.1007/978-3-540-39964-3_62
[12] Wang K, Shao Y, Xie L, Wu J, Guo S. Adaptive and fault-tolerant data processing in healthcare IoT based on fog computing. *IEEE Transactions on Network Science and Engineering*. 2020; 7(1): 263-273. https://doi.org/10.1109/TNSE.2018.2859307