An Effective FOG Computing Based Distributed Forecasting of Cyber-Attacks in Internet of Things

Vandana Roy

DoEC, Gyan Ganga Institute of Technology and Sciences, Jabalpur, M. P., India

Emails: vandanaroy@ggits.org

Abstract

Existing cloud based security procedures are insufficient to manage the ever-increasing assaults in IoT due to the volume of data generated and the processing latency. IoT applications are vulnerable to cyberattacks, and some of these assaults might have catastrophic results if not stopped or mitigated quickly enough. As a result, IoT calls for self-protect security systems that can automatically interpret attacks in IoT traffic and efficiently handle the attack situation by activating the proper response quickly. Fog computing satisfies this need because it can embed the intelligent self-protection mechanism in the distributed fog nodes, allowing them to swiftly deal with the assault scenario and safeguard the IoT application with little in the way of human interaction. At the fog nodes, the forecasting method employs distributed Gaussian process regression. The cyber-attack may be predicted more quickly and with less mistake for both low- and high-rate attacks thanks to the local forecasting about the IoT traffic characteristics at fog node. One of the fundamental necessities of an IoT security mechanism is the ability to forecast attacks in a timely manner with a high degree of accuracy, and the simulation results highlight this fact.

Keywords: IoT; GPR; FPGR; MSE.

1. Introduction:

The provision of transparent and wide-ranging seamless services with security is the major issue in deploying IoT. With the proliferation of IoT devices, increasing sophistication, and massive amounts of data, it has become a prime target for cybercriminals. The hackers not only steal information but also utilise the infected IoT device as part of a larger botnet to launch massive attacks. Security and privacy in IoT applications are of the utmost importance, as physical objects are constantly detecting and sharing personal data of our daily lives. If these gadgets are attacked, it might have catastrophic results [1]. The complexity of existing computing and communication systems is increased by the deployment of IoT devices in both managed and unmanaged environments, hence increasing the vulnerabilities of IoT. Due to its reliance on wireless data transfer, IoT applications must adhere to the same stringent security standards as the Internet and device networks [2]. The anticipated widespread entry of devices and sensors into personal places like the home, the car, and wearable gadgets poses significant privacy and security risks in IoT applications [3].

To harness the full potential of the IoT, numerous cutting-edge technologies have been developed, including cloud computing, Software Defined Networking (SDN), big data analysis, intelligent sensors, etc. Most of these IoT technologies, however, are still in their infancy, and their implementation carries with them significant technological concerns. As a result, maintaining security and privacy in the IoT's cutting-edge technologies presents fresh issues [4]. The security risks of the present Internet are exacerbated by the diverse nature of IoT. Most IoT data consists of personally identifiable information that must be safeguarded. In an Internet of Things

Doi : https://doi.org/10.54216/JCIM.120201
Received: December 12, 2022 Revised: February 05, 2023 Accepted: May 09, 2023

setting, perimeter defences are insufficient. To develop security instincts and to deliver an effective defence response to evolving threats, it must analyse and interpret the huge structured and unstructured data from IoT devices [5].

Several forms of cyberattacks are frequently seen in the Internet of Things. While these attacks may be modelled after those already in use, their effects will be unique for each IoT use case, device type, communication protocol, layer of security, and other factors.

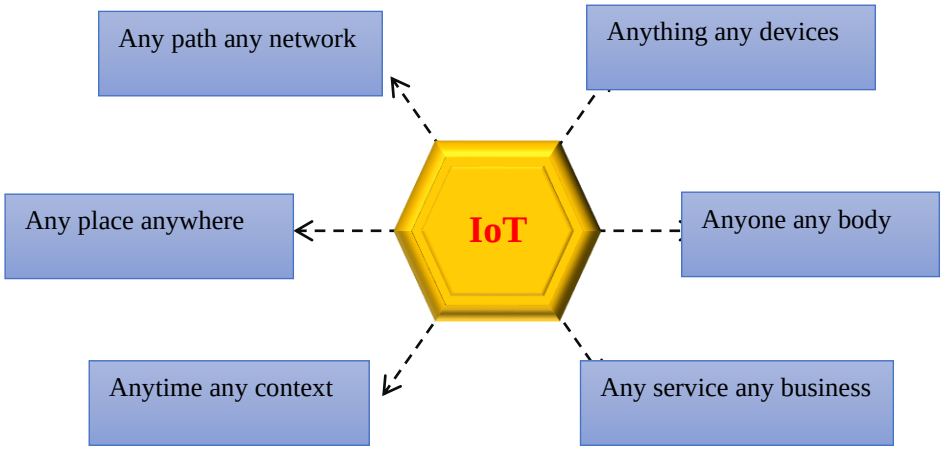

Figure 1: Connectivity in internet of things (IOT).

CISCO has invented a new computing paradigm called "fog computing," which moves previously cloud-based data and services to the network's periphery [6]. It uses the principle of distributed computing, in which tasks such as data processing, storage, and service delivery are delegated to nodes closer to the network's periphery. Data from the burgeoning IoT's millions of connected devices is massive and must be analysed quickly. Fog computing can fulfil this need [7]. The heterogeneity of the underlying equipment is hidden by an abstraction layer in fog nodes, and a standard programmable interface is provided through virtualization. Orchestration is needed in fog computing to coordinate the sharing of services and resources amongst fog nodes [8].

Fog computing offers increased assistance to IoT applications, along with reduced latency and lower bandwidth use. Tracing fog nodes gives IoT end devices awareness of their physical location. For large-scale IoT applications, it can be deployed in multiple locations for maximum availability and scalability [9]. Protocols exist in fog computing that allow Internet of Things devices to move freely. Supporting interoperability and adaptability in IoT applications, it solves the problems caused by IoT's heterogeneity [10].

There are typically three components to an IoT application design that makes use of fog computing: the end device, the fog, and the cloud. Sensors of various kinds, from the simplest to the most complex, are found in the Internet of Things' end device tier. This tier's primary function is to collect data from its surroundings and transmit that information to the fog tier [11]. Edge devices like access points, gateways, and routers manage data processing, storage, and services in the fog tier of a distributed computing architecture. The data is transferred from the fog nodes to the cloud tier, which is responsible for global data management. The data is also presented in a final form, tailored to the needs of the IoT application.

2. Existing Work Done:

Many studies have been conducted on the topic of developing self-protecting autonomic computing systems. The authors advocated a decentralised method for self-defence. This solution uses a software component design to detect and remove potentially harmful nodes automatically [12]. As a defensive measure, it uses a variety of sensors to identify malicious nodes and then to isolate them. The fact that this device can only be used in lab conditions is a big limitation.

Unsupervised Behaviour Learning (UBL) is a self-protection mechanism developed by researchers for use in cloud environments. Automatically capturing dynamic system behaviours via Self-Organizing Map (SOM) learning. It analyses patterns of behaviour to discover previously unseen outliers and anticipates new kinds of

Doi : https://doi.org/10.54216/JCIM.120201
Received: December 12, 2022 Revised: February 05, 2023 Accepted: May 09, 2023

outliers [13]. Although UBL is capable of spotting and anticipating anomalies, it lacks a response mechanism to deal with them.

The authors presented a self-configuring, self-protecting autonomic computing system. To aid in the deployment of an autonomous system, it is built around two primary modules: the Component Management Interface (CMI) and the Component Runtime Manager (CRM). Anomaly detection and defence in this system are handled by an automated online monitoring tools and a feature selection method [14-16]. The use of two modules requires more resources, making it inappropriate for devices with limited power.

The researchers introduced a self-protecting cloud infrastructure design they named VESPA. This design safeguards the virtual space with a suite of regulatory mechanisms. Security within and between layers of infrastructure is governed by a set of policies. It is made up of attack-detection and -reaction components that may be coordinated in a variety of ways, allowing for customizable security policies in the cloud [17]. Only self-defence in cloud environments is addressed by this approach.

A method of self-defence against denial-of-service attacks was proposed by the authors. The network's foundation is the Cognitive Packet Network architecture, which employs intelligent packets to make QoS-aware path selections. Downstream resources are made available thanks to the system's ability to autonomously recognise and modify harmful packets using trace-back of attack flows [18]. This system's fundamental flaw is that it is unfit for usage in a low-resource setting because it relies on a trace-back approach for recovery.

Using the Quality of Service (QoS) metric of network flows, researchers presented a self-defence architecture to identify the abnormality. It is capable of online monitoring and was developed using Hotelling's T2 methodology to identify malicious data by comparing deviations in QoS characteristics from typical network traffic [19]. It divides the flow of traffic into four categories—normal, probable normal, probable abnormal, and abnormal—using a novel metric called Abnormality Distance (AD). Priority scheduling is given according to an AD metric to lessen the effects of an attack if anomalous traffic is detected [20].

The system's primary flaw is that countermeasures seldom halt an assault entirely. For stationary wireless sensor networks, the researchers presented a self-protect strategy. In this method, sensors in the same network actively keep an eye on one another. For the least p-self protection problem, this method yields a centralised and distributed algorithm with constant approximation ratio [21]. However, this method has the main limitation of being applicable exclusively to stationary wireless sensor networks.

The authors propose a self-protect mechanism framework built specifically for the Internet of Things ecosystem. To prevent breaches in the IoT-based system, this strategy employs a model-based cyber security management approach [22]. Using the Master Controller Virtual Machine (MC-VM), it can identify and prevent numerous types of assaults. This system's primary flaw is that it is too heavy to easily deploy on low-powered Internet of Things gadgets [23].

The suggested IoT self-protection system differs in significant ways from the aforementioned efforts. To begin, it can independently anticipate, identify, and defend against cyber-attacks in an Internet of Things (IoT) setting [24]. Then, it uses the huge amounts of structured and unstructured data generated by IoT devices to develop security instincts and respond effectively to evolving threats. In addition, it is deployed at fog nodes to effectively safeguard low-powered devices [25].

3. The objective of the Research Work:

The study suggested here aims to improve the speed and accuracy with which cyber attacks on IoT applications may be predicted. The most important results of this study are:

1). Cyber-attack prediction utilising fog node-based distributed Gaussian process regression.

2). To foresee cyberattacks, we modified distributed Gaussian process regression to model attack traffic in IoT applications.

4. The Proposed Work:

The proposed distributed attack forecasting technique for IoT aims to predict zero-day assaults with high prediction accuracy and low error rate at a higher speed. To predict attacks more quickly than cloud-based methods, the forecasting mechanism is implemented at the distributed fog nodes that are closer to the end devices.

Doi : https://doi.org/10.54216/JCIM.120201
Received: December 12, 2022 Revised: February 05, 2023 Accepted: May 09, 2023

As an added bonus, the suggested forecasting method employs distributed Gaussian process regression, which is able to predict efficiently with reduced training time from massive data with high uncertainty emanating from an Internet of Things (IoT) setting.

In this study, we examine the challenge of predicting cyberattacks in IoT applications running on fog nodes. It is expected that the IoT application makes use of numerous clusters of IoT devices, each of which is linked to a fog node. Each fog node is linked to the IoT application's cloud backend. The fog nodes in the network connect the Internet of Things devices to the cloud application servers. IoT-connected devices' network traffic is treated as a stochastic variable. Under the assumption that discrete subsets of these random variables form a combined multivariate Gaussian distribution, these variables are used to simulate a Gaussian process. The cyber attack is predicted based on the characteristics of the network traffic. In order to anticipate a cyberattack, the fog node monitors the underlying clusters' traffic parameters and predicts their future values.

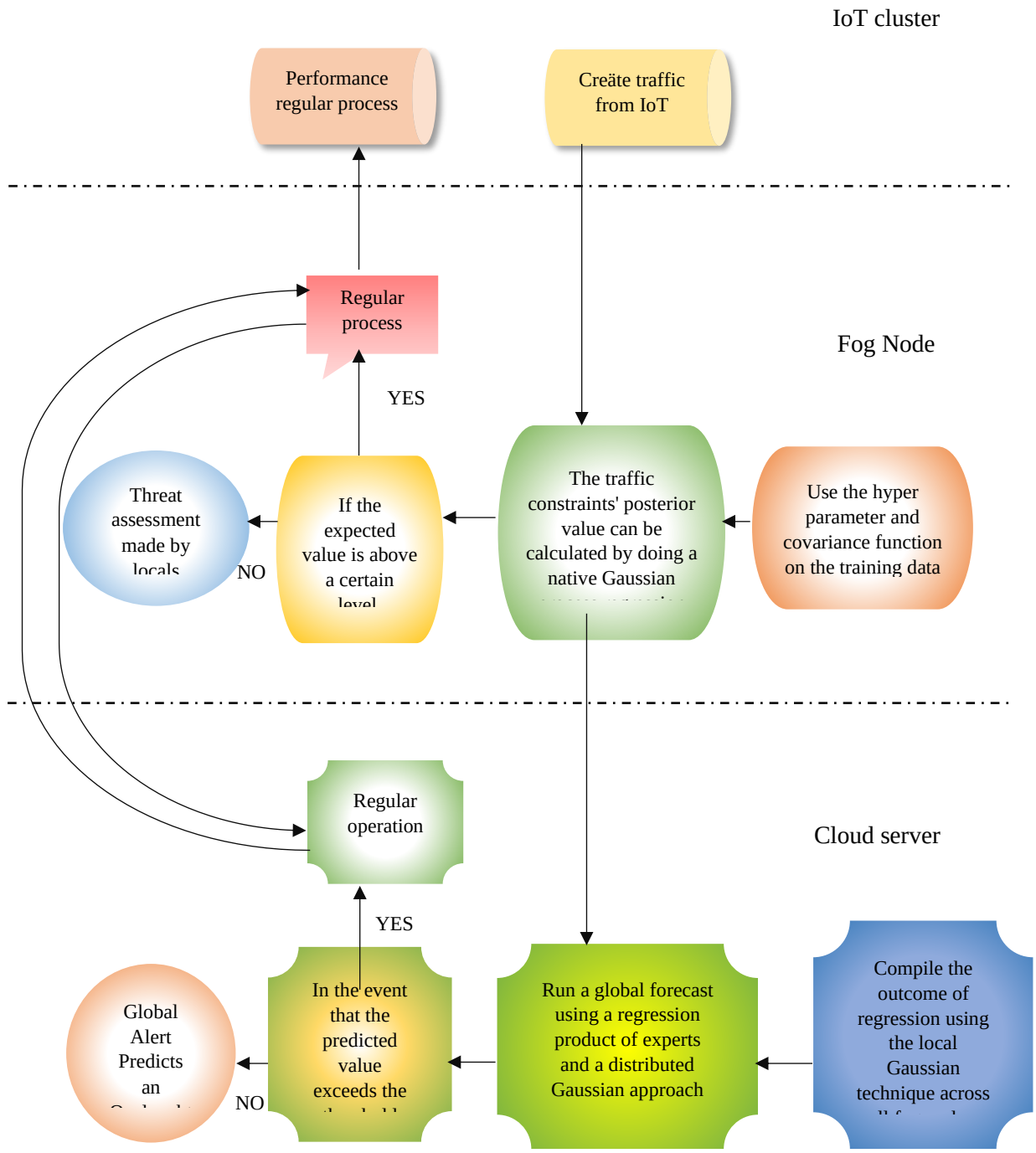

Figure 2: Current example of a model for forecasting cyber attacks in an Internet of Things setting.

Doi : https://doi.org/10.54216/JCIM.120201
Received: December 12, 2022 Revised: February 05, 2023 Accepted: May 09, 2023

The data generated by the IoT cluster's end devices is transmitted to the cloud's application server via the fog nodes. Gaussian process regression is used by the fog nodes to make predictions about the parameters of the network's traffic. The cyberattack is predicted by the fog nodes by comparing the predicted values to threshold levels. If the projected values are within the thresholds, regular operations are maintained; otherwise, an attack is considered. In order to conduct worldwide forecasting, the fog node transmits its predictions to a cloud-based application server. Using the Product-of-Experts approach of distributed Gaussian process regression, all of the local projected values are integrated at cloud to achieve the global forecasting result. Information about the expected attack's efficacy over the whole application is provided by the global forecasting result. Figure 2 depicts the process of the suggested technique for predicting cyber attacks.

Due to the vast number of unsynchronized or poorly synchronised devices, IoT traffic is highly variable. K_R, or the rational quadratic covariance function, is used to describe the inherent volatility of Internet of Things (IoT) communication. This covariance function, which is one of many that already exist and which reflects fluctuations, is defined by the equation

$$Fluctuation = K_R(R_1, l_1, \sigma) = {\alpha_1}^2(1 + \frac{{R_1}^2}{\sigma {l_1}^2})^{-\sigma} \quad (1)$$

where the form parameter is, the length scale parameter is l1, and the variance is 1. Long- and short-term dependencies in IoT application network traffic are accounted for by using the product of two isotropic squared exponential covariance functions K_S, as shown in Equation

$$Dependancy = K_S(R_2, l_2) + K_S(R_3, l_3) = {\alpha_2}^2\left(-\frac{{R_2}^2}{2{l_2}^2}\right) + {\alpha_3}^2(-\frac{{R_3}^2}{2{l_3}^2}) \quad (2)$$

The variances are denoted by 2 and 3, while l2 and l3 are the length scale parameters. As demonstrated in Equation (1), the periodic covariance function K Periodic and the isotropic squared exponential covariance KSE are utilized to simulate the periodicity and cyclic behavior of IoT traffic.

$$\text{Periodicity} = K_P(R_4, l_4, P) * K_S(R_5, l_5) = {\alpha_4}^2\left(-\frac{2\,sin^2\left(\frac{\pi r_4}{p}\right)}{{l_4}^2}\right) * {\alpha_5}^2(-\frac{{R_5}^2}{2{l_5}^2}) \quad (3)$$

The length scale parameters l4 and l5 along with the periodic parameter p and their associated variations 4 and 5 are shown below. As a result, the suggested Gaussian process regression modelling of IoT application traffic at the fog node makes use of a composite covariance function.

$$K_F = Fluctuation + Dependancy + \text{Periodicity} \quad (4)$$

$$K_F = K_R(R_1, l_1, \sigma) = {\alpha_1}^2(1 + \frac{{R_1}^2}{\sigma {l_1}^2})^{-\sigma} + K_S(R_2, l_2) + K_S(R_3, l_3) = {\alpha_2}^2\left(-\frac{{R_2}^2}{2{l_2}^2}\right) + {\alpha_3}^2\left(-\frac{{R_3}^2}{2{l_3}^2}\right) + K_P(R_4, l_4, P) * K_S(R_5, l_5) = {\alpha_4}^2\left(-\frac{2\,sin^2\left(\frac{\pi r_4}{p}\right)}{{l_4}^2}\right) * {\alpha_5}^2(-\frac{{R_5}^2}{2{l_5}^2}) \quad (5)$$

The above covariance function includes parameters whose values determine how well the proposed Gaussian process regression model fits the data. Hyperparameters refer to the range of the covariance function that affects the precision of the model.

4.1. Cyber-attack predicting Algorithm:

Input: Hyperparameter, Covariance Function

Output: Cyberattack Predictions

Fog nodes at which:

1. To train the IoT traffic data, first use the determined covariance function KFOG and Hyperparameter.
2. Using the prior values from the training data, do local Gaussian process regression to predict the posterior mean and variance.
3. To determine if an attack is possible on an IoT system, step three is to compare the anticipated value to previously established thresholds for the relevant traffic metrics.
4. When the posterior values deviate from the predetermined traffic thresholds, the fog node will claim an assault has occurred and alert the response module.

Doi : https://doi.org/10.54216/JCIM.120201
Received: December 12, 2022 Revised: February 05, 2023 Accepted: May 09, 2023

5. To execute a global prediction of the assault, the fog node communicates the values from its local prediction to the cloud.

Gather all the fog nodes'

1. Local forecast values at Cloud server.
2. Make a worldwide forecast using the Product-of-Experts model.

By comparing the expected traffic quantities to the threshold, local predictions about the cyberattack may be made. In a typical IoT setting, the threshold values are determined. By using local forecasting, the IoT application is warned of the impending attack in time to activate a defence mechanism. Concurrently, the cloud server receives the projected result from the fog nodes in order to make a global prediction regarding the cyberattack throughout the entire IoT application.

5. Result and Discussion:

In this subsection, we simulate a flooding assault in a wireless sensor network to show how the suggested fog computing based cyber-attack forecasting works. Flooding assaults are a common sort of cyberattack that can significantly slow down a network.

The proposed solution is shown effective by means of a simulated flooding attack. In Wireless sensor networks, flooding attacks are widespread and can take several forms depending on the routing protocol in use. The experiment takes into account the routing-based flooding attack in the DSR protocol. If the route to the destination node is not already in the route cache, the sending node will send a Route Request packet (RREQ) to its neighbour in order to request it. The neighbour node will reply with a Route Reply message (RREP) if it possesses the route. If a neighbour does not already have a route to the destination node in its route cache, it will rebroadcast the RREQ and include its own address in the route list. After the RREQ has been processed at the final hop, a Route Reply (RREP) message will be sent back to the originating node. The sending node adds the new route to its route cache and then uses it when sending a message. When a node on the route detects a broken connection, it reports the problem by sending a Route Error (RRER) message back to the route's origin. When a link fails, the route cache at its origin node is cleared and a replacement path is chosen. If a relay node along the path detects a failed connection to the final destination, it will send a route discovery query (RREQ) and response (RREP) during the packet's transit. Attackers use this to launch a flooding attack, and it's called the route recovery mechanism.

An attacker node in the network will insert an unavailable node into the route and then broadcast an RRER message. An RREQ is sent by an intermediate node when it is unable to locate the destination node along the route. In a similar vein, the attacker uses other nodes as puppets to flood the network with RREQ requests for downed or otherwise unreachable hosts. Puppet attack is another name for this type of flooding attack. Because it employs genuine nodes to create attack, the impact of a puppet attack is greater than that of a standard flooding attack, and it is also more difficult to detect.

Gaussian process regression with full data and Gaussian process regression with an approximation based on a subset of data (SoD) are both applied to the same assault on 10 puppets on a cloud application server in order to evaluate the suggested strategy. For the Gaussian distribution with SoD, we transmit data from our network of 150 wireless nodes to a central server, where we randomly select a chunk of data equal in size to the training data we use in the fog nodes.

The proposed method uses the same composite covariance function to fit the Gaussian process. In Table 1, we can see how the mean squared error (MSE) compares to the full Gaussian process and the Gaussian process based on the technique of moments (SoD). In terms of accuracy, the full Gaussian process regression excels, but it takes more time to train. While the proposed method outperforms the Gaussian process based on SoD in terms of training time, the latter's greater error rate can be attributed to its use of randomly selected training data points. Therefore, the error rate is higher than the Gaussian process based on SoD since mistakes from several fog nodes are aggregated at the cloud. When compared to the other two techniques, the suggested method's performance is markedly superior in predicting the attack with less training time and higher accuracy.

Doi : https://doi.org/10.54216/JCIM.120201
Received: December 12, 2022 Revised: February 05, 2023 Accepted: May 09, 2023

Table 1: MSE and training time Comparison of the proposed method.

S. No.	Methods	Mean Square Error	Training Time
1	Proposed method	0.20 +/- 0.04	38
2	GPR at cloud	0.53 +/- 0.14	41
3	FGPR at cloud [20]	0.19+/-0.08	56

Predicting low-rate attacks is not a good fit for full Gaussian process regression due to its lengthy training period. The suggested method's main benefit is that the fog nodes can predict low-rate attacks locally with minimal training.

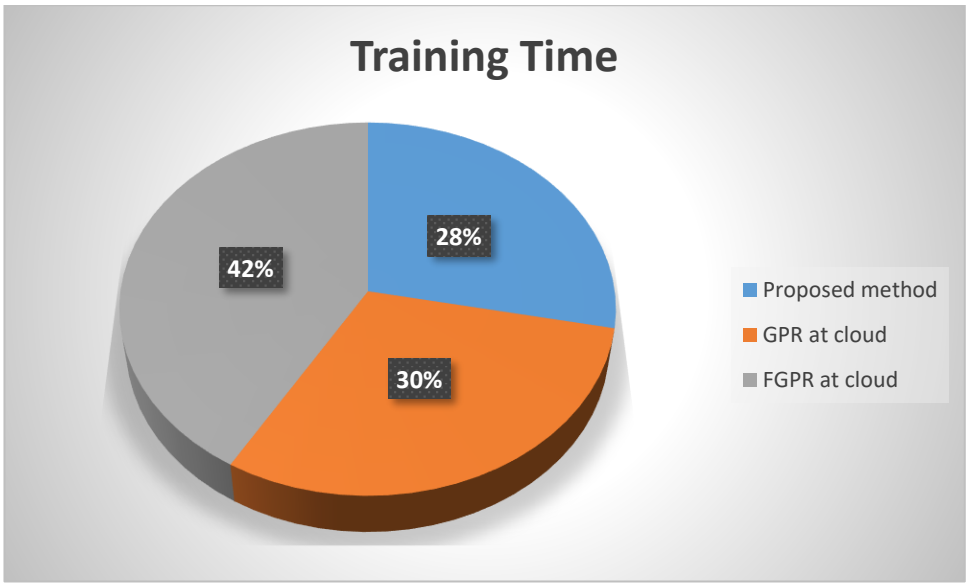

Figure 3: Training Time Comparison of the Proposed Method with Existing approach.

When compared to full Gaussian process regression in the cloud, the suggested method accurately predicts the low-rate attack at fog nodes. With this in mind, the suggested fog based cyber-attack forecasting system accurately forecasts both low-rate and high-rate attacks. To demonstrate this, we constructed low-rate flooding attacks using a small number of puppets (between 1 and 6), trained for 40 seconds, and then predicted their success with the proposed approach and complete Gaussian Process regression in the cloud.

Table 2: Prediction time for low-rate attacks.

Puppet Count	Approach	Attack Time (Sec.)
1	Actual Attack	98
	Projected Approach	102
	FGP Approach [20]	138
6	Actual Attack	85
	Projected Approach	83

Doi : https://doi.org/10.54216/JCIM.120201
Received: December 12, 2022 Revised: February 05, 2023 Accepted: May 09, 2023

	FGP Approach [20]	102
12	Actual Attack	68
	Projected Approach	67
	FGP Approach [20]	69

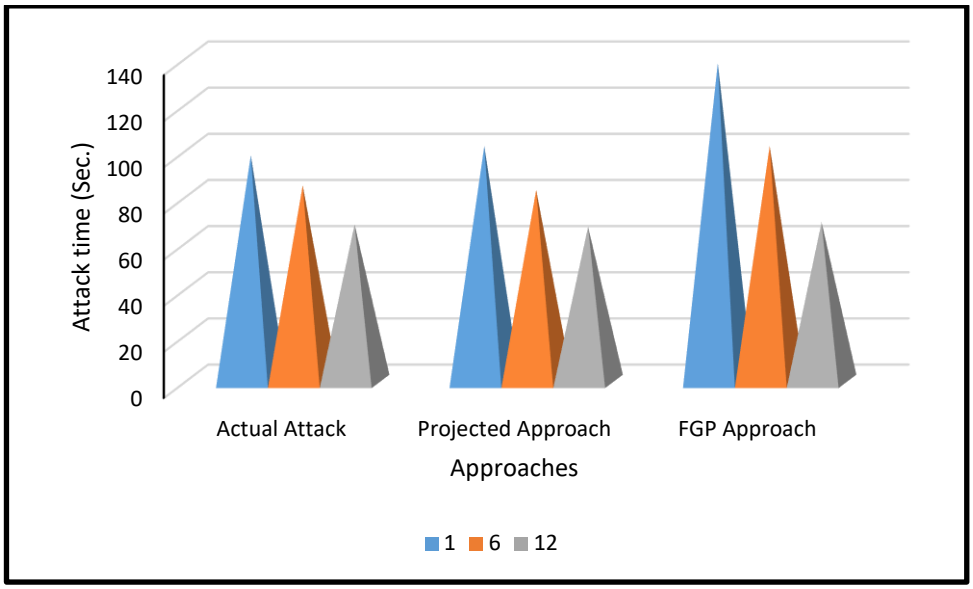

Figure 4: Prediction time for low-rate attacks.

Cyber-attacks, both low- and high-velocity, can be predicted more quickly and with less mistake thanks to local forecasting regarding the characteristics of IoT traffic at the fog node. In order to offer a worldwide perspective of the assault in the IoT application, global forecasting is carried out at the cloud server of the application itself using the Product-of-Experts technique. Simulated results highlight that the suggested system may forecast the assault at a faster rate with improved accuracy, a necessity for any security mechanism including the Internet of Things.

6. Conclusion

This study presented a self-protection solution for Internet of Things (IoT) networks that is based on fog computing. In order to implement the IoT security system, the suggested system makes use of the key advantages of fog computing, namely the reduction in latency and the reduction in bandwidth use. A mechanism for predicting cyber attacks was proposed to help identify zero-day exploits. The method relies on distributed Gaussian Process Regression at fog nodes to make predictions. Cyber-attacks, both low- and high-velocity, can be predicted more quickly and with less mistake thanks to local forecasting regarding the characteristics of IoT traffic at the fog node. The IoT application's cloud server performs the worldwide forecasting using the Product-of-Experts technique to give a comprehensive picture of any potential attacks. The simulation findings highlight the fact that the suggested method can predict the attack more quickly and accurately than the current methods.

The theoretical and simulated outcomes are presented in this study. Future study will utilise the statistical model to evaluate the effectiveness of the suggested self-protection mechanism and compare the results to those from the simulated environment. A real-world Internet of Things application can now make use of the suggested technology.

References

[1]. Ammar, M, Russello, G & Crispo, B 2018, 'Internet of things: A survey on the security of IoT frameworks', Journal of Information Security and Applications, vol. 38, pp. 8-27.

Doi : https://doi.org/10.54216/JCIM.120201
Received: December 12, 2022 Revised: February 05, 2023 Accepted: May 09, 2023

[2]. Arabo Abdullahi & Bernardi Pranggono 2013, 'Mobile malware and smart device security: Trends, challenges and solutions', 19th International Conference on In Control Systems and Computer Science (CSCS), pp. 526-531.

[3]. Fawaz Ahmed, Robin Berthier & William H Sanders 2016, 'A response cost model for advanced metering infrastructures', IEEE Transactions on Smart Grid, vol. 7, no. 2, pp. 543-553.

[4]. Lyu, L, Jin, J, Rajasegarar, S, He, X & Palaniswami, M 2017, 'Fogempowered anomaly detection in IoT using hyperellipsoidal clustering', IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1174-1184. [5]. Malialis, K, Devlin, S & Kudenko, D 2015, 'Distributed reinforcement learning for adaptive and robust network intrusion response', Connection Science, vol. 27, no. 3, pp. 234-252.

[6]. Pongle, P & Chavan, G 2015, 'Real time intrusion and wormhole attack detection in internet of things', International Journal of Computer Applications, vol. 121, no. 9, pp. 1-9.

[7]. Phan, L.A.; Nguyen, D.T.; Lee, M.; Park, D.H.; Kim, T. Dynamic Fog-to-Fog Offloading in SDN-Based Fog Computing Systems. Future Gener. Comput. Syst. 2021, 117, 486–497

[8]. Martinez, I.; Hafid, A.S.; Jarray, A. Design, Resource Management, and Evaluation of Fog Computing Systems: A Survey. IEEE Internet Things J. 2021, 8, 2494–2516.

[9]. Dash, S.; Biswas, S.; Banerjee, D. Atta-Ur-Rahman Edge and Fog Computing in Healthcare—A Review. Scalable Comput. 2019, 20, 191–206.

[10]. Sahil; Sood, S.K. Fog-Cloud Centric IoT-Based Cyber Physical Framework for Panic Oriented Disaster Evacuation in Smart Cities. Earth Sci. Inform. 2022, 15, 1449–1470.

[11]. Basir, R.; Qaisar, S.; Ali, M.; Aldwairi, M.; Ashraf, M.I.; Mahmood, A.; Gidlund, M. Fog Computing Enabling Industrial Internet of Things: State-of-the-Art and Research Challenges. Sensors 2019, 19, 4807.

[12]. Pop, P.; Zarrin, B.; Barzegaran, M.; Schulte, S.; Punnekkat, S.; Ruh, J.; Steiner, W. The FORA Fog Computing Platform for Industrial IoT. Inf. Syst. 2021, 98, 101727.

[13]. Sadaf, K.; Sultana, J. Intrusion Detection Based on Autoencoder and Isolation Forest in Fog Computing. IEEE Access 2020, 8, 167059–167068.

[14]. Sheikh Sofla, M.; Haghi Kashani, M.; Mahdipour, E.; Faghih Mirzaee, R. Towards Effective Offloading Mechanisms in Fog Computing. Multimed. Tools Appl. 2022, 81, 1997.

[15]. Vilela, P.H.; Rodrigues, J.J.P.C.; Righi, R.D.R.; Kozlov, S.; Rodrigues, V.F. Looking at Fog Computing for E-Health through the Lens of Deployment Challenges and Applications. Sensors 2020, 20, 2553.

[16]. Roy V et. Al., "Network Physical Address Based Encryption Technique Using Digital Logic", International Journal of Scientific & Technology Research, Vol. 9, No. 4, 2020, Pp no.- 3119-3122.

[17]. Seema Gaba , Kavita . , Sahil Verma , Monica Sood, Multicasting Data Routing for Vehicular Ad hoc Network using Fog Computing, International Journal of Wireless and Ad Hoc Communication, Vol. 3 , No. 1 , (2021) : 37-48 (Doi : https://doi.org/10.54216/IJWAC.030104)

[18]. Zhang, P.Y.; Zhou, M.C.; Fortino, G. Security and Trust Issues in Fog Computing: A Survey. Future Gener. Comput. Syst. 2018, 88, 16–27.

[19]. Hu, P.; Dhelim, S.; Ning, H.; Qiu, T. Survey on Fog Computing: Architecture, Key Technologies, Applications and Open Issues. J. Netw. Comput. Appl. 2017, 98, 27–42.

[20]. Prabavathy, S, Sundarakantham, K & Mercy Shalinie, S 2018, 'Design of cognitive fog computing for autonomic security system in critical infrastructure', Journal of Universal Computer Science, vol. 24, no. 5, pp. 577-602.

[21]. Lera, I.; Guerrero, C.; Juiz, C. YAFS: A Simulator for IoT Scenarios in Fog Computing. IEEE Access 2019, 7, 91745–91758.

[22]. Mouradian, C.; Naboulsi, D.; Yangui, S.; Glitho, R.H.; Morrow, M.J.; Polakos, P.A. A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges. IEEE Commun. Surv. Tutor. 2018, 20, 416–464.

Doi : https://doi.org/10.54216/JCIM.120201
Received: December 12, 2022 Revised: February 05, 2023 Accepted: May 09, 2023

[23]. Moshayedi, A.J.; Roy, A.S.; Taravet, A.; Liao, L.; Wu, J.; Gheisari, M. A Secure Traffic Police Remote Sensing Approach via a Deep Learning-Based Low-Altitude Vehicle Speed Detector through UAVs in Smart Cites: Algorithm, Implementation and Evaluation. Future Transp. 2023, 3, 12.

[24]. Tselios, C.; Politis, I.; Amaxilatis, D.; Akrivopoulos, O.; Chatzigiannakis, I.; Panagiotakis, S.; Markakis, E.K. Melding Fog Computing and IoT for Deploying Secure, Response-Capable Healthcare Services in 5G and Beyond. Sensors 2022, 22, 3375.

[25]. Tuli, Shreshth & Mirhakimi, Fatemeh & Pallewatta, Samodha & Zawad, Syed & Casale, Giuliano & Javadi, Bahman & Yan, Feng & Buyya, Rajkumar & Jennings, Nicholas. (2023). AI augmented Edge and Fog computing: Trends and challenges. Journal of Network and Computer Applications. 216. 103648. 10.1016/j.jnca.2023.103648.

Doi : https://doi.org/10.54216/JCIM.120201
Received: December 12, 2022 Revised: February 05, 2023 Accepted: May 09, 2023