



Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A Gaussian process regression approach to predict the k -barrier coverage probability for intrusion detection in wireless sensor networks

Abhilash Singh^a, Jaiprakash Nagar^b, Sandeep Sharma^{c,*}, Vaibhav Kotiyal^c

^a Fluvial Geomorphology and Remote Sensing Laboratory, Indian Institute of Science Education and Research, Bhopal, 462066 Madhya Pradesh, India

^b Subir Chowdhury School of Quality and Reliability, Indian Institute of Technology, Kharagpur, 721302 West Bengal, India

^c University School of ICT, Gautam Buddha University, Greater Noida, 201312 Uttar Pradesh, India

ARTICLE INFO

Keywords:

WSNs
GPR model
 k -barrier coverage probability
Intrusion detection

ABSTRACT

Sensors in a Wireless Sensor Network (WSN) sense, process, and transmit information simultaneously. They mainly find applications in agriculture monitoring, environment monitoring, smart city development and defence. These applications demand high-end performance from the WSN. However, the performance of a WSN is highly vulnerable to various types of security threats. Any intrusion may reduce the performance of the WSN and result in fatal problems. Hence, fast intrusion detection and prevention is of great use. This paper aims towards fast detection and prevention of any intrusion using a machine learning approach based on Gaussian Process Regression (GPR) model. We have proposed three methods (S-GPR, C-GPR and GPR) based on feature scaling for accurate prediction of k -barrier coverage probability. We have selected the number of nodes, sensing range, Sensor to Intruder Velocity Ratio (SIVR), Mobile to Static Node Ratio (MSNR), angle of the intrusion path, and required k as the potential features. These features are extracted using an analytical approach. Simulation results demonstrate that the proposed method III accurately predicts the k -barrier coverage probability and outperforms the other two methods (I and II) with a correlation coefficient ($R = 0.85$) and Root Mean Square Error (RMSE = 0.095). Further, the proposed methods achieve a higher accuracy as compared to other benchmark schemes.

1. Introduction

Technological advancements in Micro-Electro-Mechanical Systems (MEMS) have lead to the miniaturisation of electronic devices along with advanced wireless communication technologies, signal processing abilities and efficient power consumption (Ali et al., 2015). These advancements have allowed the manufacturers to design and develop tiny wireless sensor nodes which in turn are used to form a WSN. A WSN may consist of hundreds to thousands of sensor nodes spread independently and uniformly within the Region of Interest (RoI) (Singh, Sharma, & Singh, 2021; Amutha, Sharma, & Nagar, 2020; Singh, Kotiyal, Sharma, Nagar, & Lee, 2020). A WSN does not need any pre-installed base for support and operates in a self-structured and decentralised manner (Nagar, Chaturvedi, & Soh, 2020). Also, ease in deployment in remote/inaccessible regions, hazardous environments and emergency conditions, have paved the path for their numerous military and civilian applications such as border surveillance, industrial monitoring and

control, security, structural health monitoring, precision agriculture, healthcare, remote landslides monitoring and forest fire detection (Noel et al., 2017; Jawad, Nordin, Gharghan, Jawad, & Ismail, 2017; Dey, Ashour, Shi, Fong, & Sherratt, 2017; Kumar, Duttagupta, Rangan, & Ramesh, 2020; Aponte-Luis et al., 2018; Singh, Sharma, Singh, & Kumar, 2019).

There are several countries throughout the world with no sentinels, patrolling soldiers or residents in the vicinity of their international borders. Besides, international borders have a few checkpoints along the stretch of borders and there exists a large no man's area between the checkpoints and the border on the opposite side. Also, patrolling of soldiers along the borders is conventional, limited, and periodic; as a result, international borders remain unguarded for a long duration of time. It is highly likely that the enemies can enter the territory without being detected from these unattended areas. They may take highly sensitive and classified information or destroy some key infrastructure which may cause a significant loss to the country. Therefore, intrusion

* Corresponding author at: Gautam Buddha University, Greater Noida 201312, India.

E-mail addresses: sabhilash@iiserb.ac.in (A. Singh), jpnagar91@gmail.com (J. Nagar), sandeepsvce@gmail.com (S. Sharma), kotiyalvaibhav98@gmail.com (V. Kotiyal).

<https://doi.org/10.1016/j.eswa.2021.114603>

Received 10 October 2020; Received in revised form 12 December 2020; Accepted 11 January 2021

Available online 20 January 2021

0957-4174/© 2021 Elsevier Ltd. All rights reserved.

detection is a paramount security issue for any nation.

One of the critical applications of WSNs is intrusion detection *i.e.*, for monitoring the border areas, no man's territories, infrastructures and their perimeters to detect and prevent any unauthorised access to these regions. A plethora of literature is available proposing different analytical models and algorithms for intrusion detection. [Assad, Elbhiri, Faqih, Ouadou, and Aboutajdine \(2016\)](#) have proposed a mathematical model to investigate the quality of intrusion detection in terms of k -coverage and k -connectivity metrics. They have conjectured that an intruder pursues a parametric curve route to move from one boundary of the region to another. This assumption is not always true because an intruder may pursue a linear/zigzag route or travel at a given angle to avoid its detection or minimise the chances of its discovery. False alarms are another concern as it is an undesirable trait of WSNs. To tackle this issue, [Sharma and Chauhan \(2020\)](#) have presented a three-level hierarchy-based sensor fusion scheme to ensure the presence of an intruder. They have used an acoustic signal and sensing probability model along with the k -mean clustering and Likelihood Ratio Test (LRT) to minimise false alarm rate and maximise intruder detection probability. Due to the implementation of various security measures for WSNs, an intruder may be aware of the position of sensing nodes in the region and move accordingly to avoid its detection. To deal with such troubles, [Wang, Huang, Li, He, and Sha \(2020\)](#) have proposed a vehicle collaboration sensing network model. In this model, they have deployed static sensor nodes and mobile sensing vehicles that work together to detect the presence of an intruder trying to cross the border areas. Similarly, [Mohapatra, Sahoo, and Wu \(2016\)](#) have proposed an architecture based on big data analytical techniques. The proposed architecture is used to process and study the immense volume of data produced by static microwave sensors and mobile cameras deployed to render a barrier coverage against an intruder trying to cross the border areas. Further, [Ghosh, Neogy, Das, and Mehta \(2018\)](#) have proposed two energy-efficient and loop-free routing schemes for intrusion detection at unattended borders, large military barracks and other sensitive places. The proposed schemes render an improved lifetime for the concerned network as compared to LEACH and TEEN routing schemes.

Previous studies indicate that mobility in WSNs significantly improves their performance ([Liu, Brass, Dousse, Nain, & Towsley, 2005; Keung, Li, & Zhang, 2012; He, Chen, Li, Shen, & Sun, 2013](#)). Thus, [Nagar and Sharma \(2018\)](#) have studied k -barrier coverage probability in a Mobile Sensor Network (MSN). They have also analysed the influence of various network and system parameters such as sensor and intruder speed, nodes count, and node's sensing range, on the k -barrier coverage probability. Recently, [Sharma and Nagar \(2020\)](#) have derived a closed-form analytical expression to compute the k -barrier coverage probability for an intruder by deploying mobile sensor nodes in a rectangular RoI. They have assumed that an intruder moves at a given angle with the minimum length (w) route to transverse the RoI. At first, they have calculated the area covered by the intruder when travelling from one boundary region to another. Then, this area is used to obtain an analytical expression for the k -barrier coverage probability.

Although the above-discussed methods are useful and effective for intrusion detection in border regions, they have some significant drawbacks like very high computational cost and time complexity. Since WSNs generate an immense volume of data, and the processing and analysis of this data are very time consuming and cumbersome task. To overcome this problem, we have proposed an efficient machine learning approach for accurate prediction of the k -barrier coverage probability with less time and computational requirements. To the best of our knowledge, no other study has been done and published to address this problem.

In this paper, we have proposed three machine learning methods based on the GPR model. We have used the analytical approach to select and extract six features, namely the number of nodes, sensing range, SIVR, MSNR, angle of the intrusion path, and required k . Afterwards, we trained all the methods and evaluated its performance to accurately

predict the k -barrier coverage probability using R, RMSE, and time complexity parameters as metrics.

Further, we have divided the rest of the paper into six sections. In Section 2, we have discussed the related work. In the first half of Section 3, we have discussed the coverage area of a mobile sensor with the details of the analytical formulation for k -barrier coverage probability. Furthermore, we have discussed the importance of each feature and the GPR model in detail. In Section 4, we have presented the simulation scenarios of k -barrier coverage probability and GPR simulations. In Section 5, we have discussed the results of all the three methods for k -barrier coverage probability prediction. Finally, in Sections 6 and 7, we have presented the discussion and conclusion respectively.

2. Related works

In this section, we have discussed various approaches that had been already proposed to improve the intrusion detection capabilities in WSNs. Several studies have been conducted for intrusion detection and prevention using machine learning approaches. [Otoum, Kantarci, and Mouftah \(2019\)](#) have proposed a Restricted Boltzmann Machine-based Clustered Intrusion Detection System (RBC-IDS) for efficient intrusion detection in WSNs. They compared the performance of RBC-IDS with adaptive machine learning-based IDS and reported a higher accuracy. Further, [Han, Zhou, Jia, Dalil, and Xu \(2019\)](#) proposed an energy-efficient low-consumption IDS for WSNs based on game theory and an auto-regressive model. The proposed model is capable of predicting the attack time and reduces energy consumption. [Tan et al. \(2019\)](#) has proposed intrusion detection technique based on Synthetic Minority Oversampling Technique (SMOTE) and Random Forest (RF) algorithm. They have used SMOTE to address the class imbalance issue of the intrusion dataset and RF algorithm to train the classifier. They reported an overall accuracy of 92.57%. Recently, [Nancy et al. \(2020\)](#) proposed an efficient intrusion detection approach for WSNs based on a fuzzy decision tree with a dynamic recursive feature selection algorithm. They have reported high precision level and low network delay. More recently, [Riyaz and Ganapathy \(2020\)](#) has proposed a deep learning approach based on Convolutional Neural Network (CNN) for effective intrusion detection in wireless networks. They have reported accuracy of 98.88%. [Nguyen and Kim \(2020\)](#) have proposed a genetic CNN approach for intrusion detection in WSNs. They have proposed a unique feature subset approach for intrusion detection. The features are selected using a Genetic Algorithm (GA), Fuzzy C-mean clustering, and CNN extractor method. They have reported an accuracy of 98.2 %. To deal with the big data scenario, [Hassan, Gumaei, Alsanad, Alrubaiyan, and Fortino \(2020\)](#) proposed a hybrid deep learning model for efficient intrusion detection. This model is based on CNN and a Weight-Dropped, Long Short-Term Memory (WDLSTM) network. They reported an accuracy of 96.97%.

Overall, this study aims to improve the computational efficiency and accuracy in the previous studies using regression-based machine learning approach.

3. System model

We uniformly and independently distribute a finite count of mobile sensors inside a 2D rectangular RoI with width w meters and length l meters, respectively, as shown in [Fig. 1](#). This random spread of nodes renders a homogeneous Poisson point distribution of nodes with density $\rho_A = \frac{N(A)}{A}$ in the RoI, where, $N(A)$ and $A = l \times w$, represent the nodes count and the area of RoI respectively. All sensors are assumed homogeneous, *i.e.*, they possess an equal amount of initial energy, hardware and software capabilities and identical sensing range. Furthermore, we assume that sensor nodes travel conforming to the random direction mobility model ([Camp, Boleng, & Davies, 2002](#)) in which the speed and direction of motion of a given node are independent of the speed and

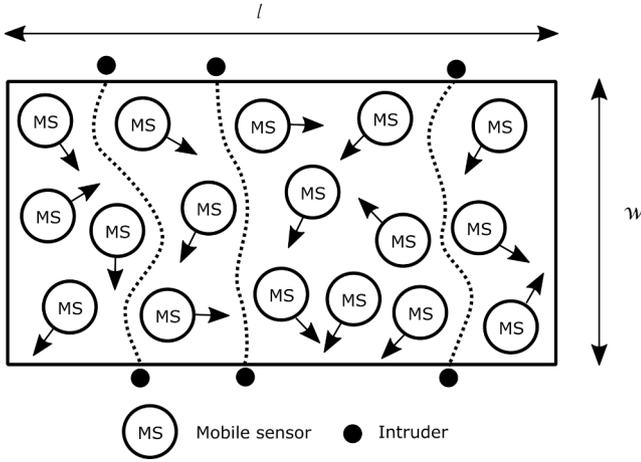


Fig. 1. Mobile sensor network scenario with intruder.

direction of other sensor nodes of the network. The random direction mobility model keeps the distribution of mobile nodes uniform at any particular instant of time by avoiding the accumulation of sensor nodes in the middle of the RoI. A point within the RoI is assumed to be monitored by a sensor node if and only if its position is at a distance less than or equal to the node's sensing range. We assume that an intruder is a dot-like object striving to transverse the RoI from one parallel boundary to the opposite parallel boundary of the RoI. Further, it is assumed that the intruder does not have any prior information about the position and mobility pattern of the sensor nodes. To avert its discovery or maximise its discovery time, an intruder may follow different routes, e.g., it can pursue a linear route, it can travel at a zigzag/curved route or travel at a particular angle to avoid its discovery or maximise its discovery time. In this work, we consider that an intruder moves with a given speed v_I at a given angle $\theta \in \left[0, \arctan\left(\frac{l}{w}\right)\right]$ with the width of the region w from the spot of intrusion (x, w) . To calculate the k -barrier coverage probability (i.e., the probability of an intruder being discovered by at least k distinct sensor nodes cumulatively not simultaneously) for a given intrusion route, we need to compute the area of the intrusion route. Computing the intrusion route area is an efficacious method. It depends on the abscissa of the spot of intrusion and angel (θ) at which an intruder moves from one boundary to the other. We discovered three elementary routes an intruder may follow to transverse the RoI and all other potential routes will be the subsets of these three elementary routes and can be incorporated accordingly.

3.1. Coverage area of a mobile sensor node

At first, we compute the area of fundamental intrusion routes that an intruder may follow to transverse the RoI. Then, this intrusion route area is used to attain a closed-form analytical expression for the k -barrier coverage probability of an intruder.

Case 1: When the spot of intrusion is $(0, w)$ and $\left(0 \leq \theta \leq \arctan\left(\frac{l}{w}\right)\right)$.

In this case, we assume that an intruder infiltrates the RoI from the point $(0, w)$ and moves at θ to transverse the RoI as shown in route 1 of Fig. 3. Then, the intruder itinerary is given by

$$Y = X \cdot \tan(90^\circ + \theta) + w \quad (1)$$

In this case, we need to calculate the area of two semicircular regions and one rectangular region, as shown in Fig. 2. Then, the value of ordinate when the value of abscissa is r_s is given by

$$Y_i = r_s \cdot \tan(90^\circ + \theta) + w \quad (2)$$

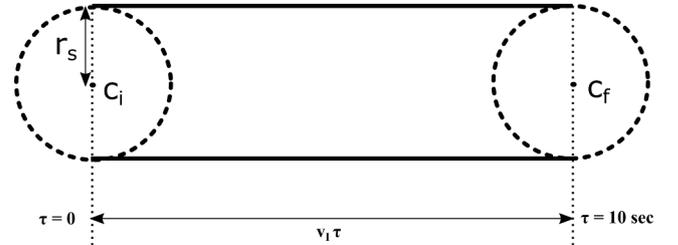


Fig. 2. Coverage area of a mobile sensor in time τ seconds.

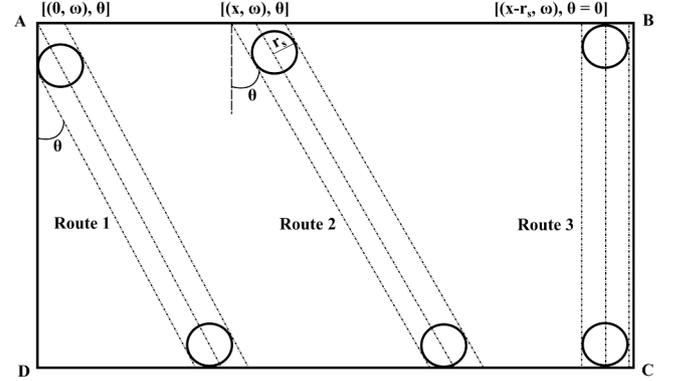


Fig. 3. Motion of an intruder at various intrusion path angles (θ).

In this way, we obtain the center of the circle at the starting point and is given by $C_i = (r_s, Y_i)$. Now, the center of the circle at the end of the route is denoted by $C_f = (X_f, r_s)$ and the value of X_f when $Y_f = r_s$ is given by

$$X_f = \frac{r_s - w}{\tan(90^\circ + \theta)} \quad (3)$$

The length of the rectangular region of the intrusion route denoted by λ is equal to the distance between initial $C_i = (r_s, Y_i)$ and final center $C_f = (X_f, r_s)$ of the circular part and is given by

$$\lambda = \sqrt{(X_f - r_s)^2 + (Y_i - r_s)^2} \quad (4)$$

The width of the rectangular region will be twice of the sensing range r_s . The total area of the intrusion route will be equal to the sum of the areas of the rectangular and two semicircular parts and is given by

$$A_I = \lambda \cdot 2r_s + \pi r_s^2 \quad (5)$$

Case 2: When intrusion spot is $(r_s < x < l - r_s, w)$ and $\left(0 \leq \theta \leq \arctan\left(\frac{l}{w}\right)\right)$.

In this case, an intruder is conjectured to enter from the position (x, w) at an angle θ to transverse the RoI as depicted in route 2 of Fig. 3. The itinerary of the intruder is given by

$$Y = (X - x) \cdot \tan(90^\circ + \theta) + w \quad (6)$$

Then, the abscissa of the center of the circle at the starting of the route two can be obtained by putting $Y = w - 2r_s$ and is given by

$$X_i = x - \frac{r_s}{\tan(90^\circ + \theta)} \quad (7)$$

and the final center of the circle at the end of the path is obtained by putting $Y = r_s$ and is given by $C_f = (X_f, r_s)$, where X_f is

$$X_f = x + \frac{r_s - w}{\tan(90^\circ + \theta)} \quad (8)$$

Therefore, the center of the circle at the beginning and the end of the route are denoted by $C_i = (X_i, w - r_s)$ and $C_f = (X_f, r_s)$ respectively. As discussed above, the width of the rectangular part of the region will be $2r_s$ and the distance between the initial and final center of the circle in this path will serve as the length (λ) of the rectangular area of the path and can be calculated as

$$\lambda = \sqrt{(X_f - X_i)^2 + (2r_s - w)^2} \quad (9)$$

In this way, the total area of the intrusion route will be equal to the sum of the areas of the rectangular and the two semicircular parts and is computed as

$$A_I = \lambda \cdot 2r_s + \pi r_s^2 \quad (10)$$

Case 3: When intrusion spot is $(r_s \leq x \leq l - r_s, w)$ and $\theta = 0^\circ$.

In this case, an intruder will take the least distance route to transverse the RoI as shown in route 3 of Fig. 3 and the total covered area can be obtained by summing the area of rectangular and circular regions and is computed as

$$A_I = (w - 2r_s) \cdot 2r_s + \frac{1}{2} \pi r_s^2 + \frac{1}{2} \pi r_s^2 \quad (11)$$

As an illustration, let us assume that an intruder wants to cross a rectangular belt RoI from one parallel boundary AB to the opposite boundary CD. As per its convenience, the intruder may choose to start from the corner A with coordinates $(0, w)$, a random point between r_s and $(l - r_s)$, and a random point between $(l - r_s)$ and (r_s) . Similarly, the intrusion may travel at an angle of 0° representing the shortest length route between AB and CD. It is also possible that the intruder may follows an angle $(0 < \theta \leq \arctan(\frac{l}{w}))$. All the possible combinations of starting point between A and B, and the intrusion route angle θ result in three fundamental routes discussed above. Any other possible route will be a subset of these three elementary routes and can be dealt with accordingly.

3.2. Analytical formulation for k -barrier coverage probability

To obtain an analytical closed-form expression for k -barrier coverage probability ($P(\Phi \geq k)$), we need to compute the average uncovered distance (γ_{avr}), the relative velocity of mobile sensor with respect to (w.r.t.) the intruder (\bar{v}_{rel}), coverage rate (δ_s) and unattended time duration between consecutive sensor coverage as prerequisites. The definitions of these prerequisites are taken from Sharma and Nagar (2020). The mean unguarded distance by nodes in a stationary network is given by

$$\gamma_{avr} = \frac{1}{(\rho_A) \cdot (C_s)} \quad (12)$$

where, $C_s = (2r_s + \frac{\pi r_s^2}{v_I \cdot \tau})$ is the coverage cross section by static sensors.

The expected velocity of mobile sensor nodes w.r.t. the intruder can be calculated by

$$\bar{v}_{rel} = \frac{2(v_I + v_s)}{\pi} \chi(\varphi) \quad (13)$$

where, $\chi(\varphi)$ is an incomplete elliptical integral and is computed as

$$\chi(\varphi) = \int_0^\varphi \sqrt{1 - \phi \sin^2(\phi)} d\phi \quad (14)$$

where, ϕ is the angle between the direction of sensor and intruder, and

$$\phi = \frac{4v_I v_s}{v_I^2 + v_s^2 + 2v_I v_s}$$

Here, we assume that mobile sensor nodes move with constant ve-

locity v_s and an intruder tries to transverse the RoI with velocity v_I .

Further, the total sensor coverage in time τ is given by

$$\delta_s \tau = \rho_A \cdot \bar{v}_{rel} \cdot \left(2r_s \cdot \tau + \frac{\pi r_s^2}{v_I} \right) \quad (15)$$

The time duration between successive overages for which a given region is unguarded is the reciprocal of coverage rate.

Now, the k -barrier coverage probability ($P(\Phi \geq k)$) can be computed using the closed-form expression below (Sharma & Nagar, 2020)

$$P(\Phi \geq k) = 1 - \sum_{j=0}^{k-1} \left(\frac{e^{-\delta_s \tau} (\delta_s \tau)^j}{j!} \right) \quad (16)$$

where, δ_s and τ represents the coverage rate and time respectively.

3.3. k -barrier coverage probability for a hybrid network

In this section, we attain a closed-form formula for the k -barrier coverage probability of a hybrid network consisting of both mobile as well as static sensor nodes. We assume that the network is made of $\frac{N(A)}{\mu+1}$ and $\frac{N(A) \cdot \mu}{\mu+1}$ number of static and mobile sensor nodes respectively, where μ denotes MSNR. Thus, the coverage rates for mobile and static sensor nodes are denoted by δ_{ms} and δ_{ss} respectively and can be calculated as

$$\delta_{ms} = \rho_A \cdot \frac{\mu}{\mu+1} \cdot \left(2r_s + \frac{\pi r_s^2}{v_I \cdot \tau} \right) \bar{v}_{rel} \quad (17)$$

$$\delta_{ss} = \frac{\rho_A}{\mu+1} \cdot \left(2r_s + \frac{\pi r_s^2}{v_I \cdot \tau} \right) \bar{v}_I \quad (18)$$

The intensities of Poisson processes for static and mobile sensor nodes are achieved with the help of coverage rates of static and mobile sensor nodes and are denoted by $\delta_{ss} \tau$ and $\delta_{ms} \tau$ respectively. The inhomogeneous Poisson processes obey superposition principle, therefore, the resultant intensity for the hybrid network is $(\delta_{ss} + \delta_{ms}) \tau$ and the k -barrier coverage probability is given by

$$P(\Phi \geq k) = 1 - \sum_{j=0}^{k-1} \left(\frac{e^{-(\delta_{ms} + \delta_{ss}) \tau} ((\delta_{ms} + \delta_{ss}) \tau)^j}{j!} \right) \quad (19)$$

3.4. Machine learning model

3.4.1. Feature importance

Feature importance scores play a critical role in any predictive model. To estimate the importance of each feature, we have used the regression ensemble method. In doing so, firstly, we trained a regression ensemble by boosting hundred regression trees (i.e., ensemble learning cycle) through LSBoost ensemble aggregation approach. We considered the regression tree as a weak learner with unity learning rate. Once we trained the ensemble, we have estimated the importance of each feature by summing these estimates over all the weak learner in it. Finally, we plotted the feature importance graph (Fig. 4). We found that out of six features, the number of nodes is the most important feature with highest feature score followed by sensing range, MSNR, SIVR, required k and lastly the angle of the intrusion path with least feature importance or score.

Further, to show the average marginal effect of each feature on the predictand, we have plotted the partial dependency plot (Friedman, 2001) of each feature (Fig. 5). In the same plot, we have also plotted the individual conditional expectation. It dis-aggregates the average effect and helps us to observe the functional relationship between the predictand and features at each instance.

3.4.2. GPR model

Regression is a type of supervised learning problem. It attempts to

model a relationship between a definite number of features and a continuous response variable called predictand. Amongst the various regression algorithms, GPR (Williams & Rasmussen, 2006) is the most robust, accurate and easy to implement (Østergård, Jensen, & Maa-gaard, 2018). It is a probabilistic model that defines the uncertainty about the predictand. It assumes that the predictand follows a joint multivariate normal distribution. It found applications in vast areas like

$$\mathbf{f}_* | \mathbf{X}_*, \mathbf{X}, \mathbf{f} \sim \mathcal{N}(\mathbf{K}(\mathbf{X}_*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{f}, \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}_*)) \quad (22)$$

image processing, healthcare, wireless sensor networks, traffic analysis, ageing of metals and alloys and various other types of datasets to predict the desired parameter (He & Siu, 2011; Richter & Toledano-Ayala, 2015; Sun & Xu, 2010; Yabansu, Iskakov, Kapustina, Rajagopalan, & Kalidindi, 2019; Saha, Saha, Saxena, & Goebel, 2010). Further, we have two cases in GPR: GPR with zero mean and GPR with non zero mean. In GPR with zero mean, the mean vector of the joint multivariate normal distribution is supposed to be a zero and the co-variance is calculated using the co-variance function. We consider our input-output pairs of training data to be a set, S , given by $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$, where $\mathbf{x}_1, \dots, \mathbf{x}_N$ are the vectors of the multivariate input and $\mathbf{y}_1, \dots, \mathbf{y}_N$ are the corresponding scalar output. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T$ represents the input matrix whose rows are the training sets and T represents the transpose, $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N]^T$ represents the vector of the output for our training data sets. It is assumed and considered that the training output observations are effected by additive noise that follow an independent identically distributed (iid) zero-mean Gaussian distribution i.e., $\mathcal{N}(0, \sigma_n^2)$, where σ_n^2 is the noise variance.

Let \mathbf{f} be the joint distribution of the noiseless training output values corresponding to the \mathbf{X} training input matrix and \mathbf{f}_* be the joint distribution of the test output values corresponding to the \mathbf{X}_* testing input matrix. It can be written in matrix form as

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right) \quad (20)$$

$$\mathbf{y} \sim \mathcal{N}(\mathbf{f}, \sigma_n^2 \mathbf{I}) \quad (21)$$

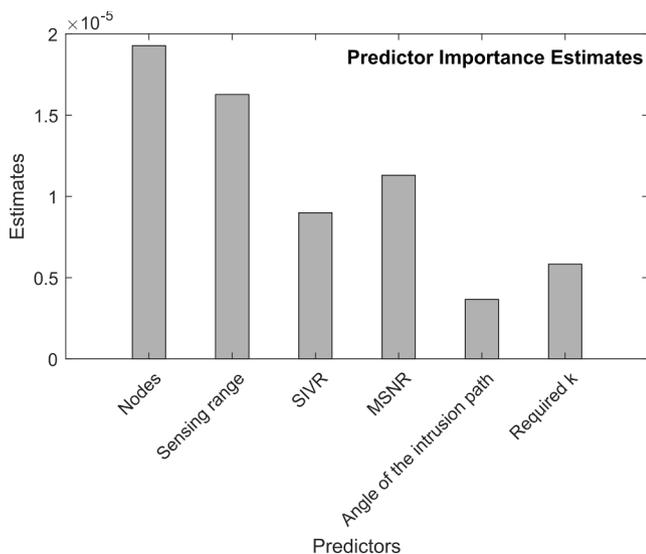


Fig. 4. Predictor importance graph.

For n training and $n_* = (N - n)$ testing samples, the term $K(\mathbf{X}, \mathbf{X}_*)$ represents the co-variances matrix of size $n \times n_*$ calculated over all the training and testing points. Similar explanation can be drawn for the remaining co-variance matrix $K(\mathbf{X}, \mathbf{X})$, $K(\mathbf{X}_*, \mathbf{X}_*)$ and $K(\mathbf{X}_*, \mathbf{X})$. Finally, we estimate the \mathbf{f}_* corresponding to the test inputs set, \mathbf{X}_* . In probabilistic term, it can be easily obtained by sampling the joint posterior distribution by obtaining the mean and co-variance matrix using

The noisy version of the predicted output can be mathematically written in the form $\mathbf{y} = f(\mathbf{x}) + \varepsilon$ where ε is the iid Gaussian noise with zero mean and noise variance σ_n^2 , and the corresponding noisy observation is given by

$$\text{cov}(y_p, y_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq} \text{cov}(\mathbf{y}) = \mathbf{K}(\mathbf{X}, \mathbf{X})\sigma_n^2 \mathbf{I} \quad (23)$$

where δ_{pq} represents the binary Kronecker delta whose value is unity for $p = q$ and zero if $(p \neq q)$. The noisy version of Eq. (20) is

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right) \quad (24)$$

With reference to Eq. (22), the conditional distribution leads to the main predictive GPR equations as

$$\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*)) \quad (25)$$

where

$$\begin{aligned} \bar{\mathbf{f}}_* &\triangleq E[\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_*] = K(\mathbf{X}_*, \mathbf{X}) [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{y} \\ \text{cov}(\mathbf{f}_*) &= \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X}) [K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \end{aligned} \quad (26)$$

The expression consists of $K(\mathbf{X}, \mathbf{X})$, $K(\mathbf{X}, \mathbf{X}_*)$ and $K(\mathbf{X}_*, \mathbf{X}_*)$ can be written in concise form by assigning $K = K(\mathbf{X}, \mathbf{X})$ and $K_* = K(\mathbf{X}, \mathbf{X}_*)$. For single test point \mathbf{x}_* , we put $\mathbf{k}(\mathbf{x}_*) = \mathbf{k}_*$ to mark covariance vectors among the n training points and the test points. Taking into account this concise notation, Eq. (26) reduced to

$$\begin{aligned} \bar{f}_* &= \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} \\ \mathbb{V}[\mathbf{f}_*] &= \mathbf{k}(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_* \end{aligned} \quad (27)$$

An alternate way to see Eq. (27) is to visualise it as a linear combination of the n number of kernel functions with each being centred at every available training sample.

$$\bar{f}(\mathbf{x}_*) = \sum_{i=1}^n \alpha_i \mathbf{k}(\mathbf{x}_i, \mathbf{x}_*) \quad (28)$$

where, $\alpha = (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}$.

The co-variance functions have few free hyperparameters, which are usually learned from the data. The most common method to fit these free parameters is by maximising the log marginal likelihood of the training features, expressed as

$$\log p(\mathbf{y} | \mathbf{X}) = -\frac{1}{2} \mathbf{y}^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} - \frac{1}{2} \log |\mathbf{K} + \sigma_n^2 \mathbf{I}| - \frac{n}{2} \log 2\pi \quad (29)$$

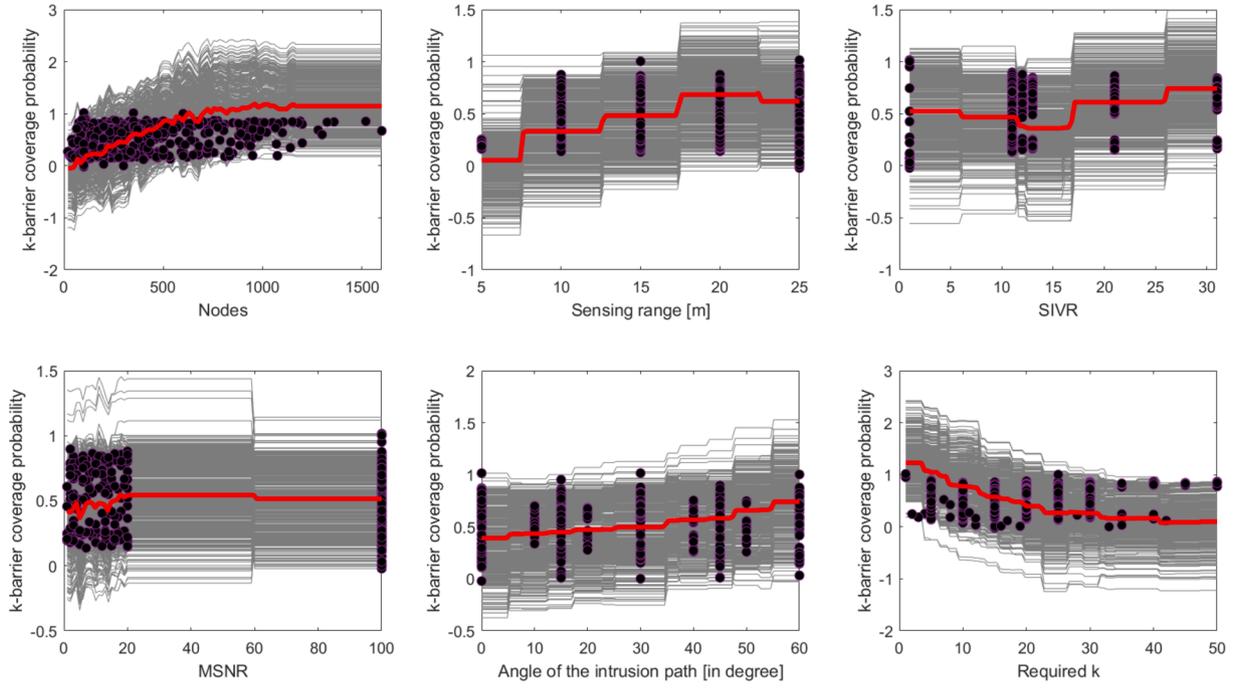


Fig. 5. Partial dependence plot (PDP) and individual conditional expectation (ICE) plots.

It is customary to assume a Gaussian process to have a zero mean, but it is not necessarily true in all cases. For reasons such as model understanding, prior information conveying benefits and a number of methodical and mathematical limits, the mean could be non-zero and requires to be a model for the GP for exact results. We can easily model a non-zero mean into the functions by using the concept of explicit basis functions. To incorporate the deterministic mean function $m(\mathbf{x})$, we need to define the conventional zero-mean GP as a difference of observations and deterministic mean function along with

$$f(\mathbf{x}) \sim \mathcal{L}(\mathbf{m}(\mathbf{x}), \mathbf{k}(\mathbf{x}, \mathbf{x}')) \quad (30)$$

where the predictive mean can be estimated by

$$\bar{\mathbf{f}}_* = \mathbf{m}(\mathbf{X}_*) + \mathbf{k}(\mathbf{X}_*, \mathbf{X}) \mathbf{K}_y^{-1} (\mathbf{y} - \mathbf{m}(\mathbf{X})) \quad (31)$$

where, \mathbf{K}_y , can be expressed as $\mathbf{K} + \sigma_n^2 \mathbf{I}$, and the Eq. (26) will serve as the expression for a predictive variance without any change. However, due to the difficulty in specifying a fixed mean function, it is convenient to have a few fixed basis functions expressed as

$$g(\mathbf{x}) = f(\mathbf{x}) + \mathbf{h}(\mathbf{x})^T \boldsymbol{\beta}, \quad (32)$$

where $f(\mathbf{x})$ is the zero-mean GP having a distribution $f(\mathbf{x}) \sim \mathcal{L}(\mathbf{0}, \mathbf{k}(\mathbf{x}, \mathbf{x}'))$, $\mathbf{h}(\mathbf{x})$ represents the set of fixed basis functions and $\boldsymbol{\beta}$ represents the basis coefficients whose values is estimated from the training data. When fitting the model, we need to optimise $\boldsymbol{\beta}$ simultaneously with the hyperparameters of the co-variance function. An alternate approach is to integrate these parameters by taking prior on $\boldsymbol{\beta}$ to be Gaussian, while $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{b}, \mathbf{B})$ (O'Hagan, 1978). This give rise to another form of GP with an added feature in the co-variance function resulting from the mean uncertainty. It give rise to the expression

$$\begin{aligned} \bar{\mathbf{g}}(X_*) &= \mathbf{H}^T \bar{\boldsymbol{\beta}} + \mathbf{K}^T \mathbf{K}_y^{-1} (\mathbf{y} - \mathbf{H}^T \bar{\boldsymbol{\beta}}) = \bar{\mathbf{f}}(X_*) + \mathbf{R}^T \bar{\boldsymbol{\beta}} \\ \text{cov}(\mathbf{g}_*) &= \text{cov}(\mathbf{f}_*) + \mathbf{R}^T (\mathbf{B}^{-1} + \mathbf{H} \mathbf{K}_y^{-1} \mathbf{H}^T)^{-1} \mathbf{R} \end{aligned} \quad (33)$$

where the \mathbf{H} and \mathbf{H}_* matrix collects the fixed basis function vector, $\mathbf{h}(\mathbf{x})$, for all training cases and test cases respectively. $\bar{\boldsymbol{\beta}}$ being the mean of the

global linear model parameter and expressed as $\bar{\boldsymbol{\beta}} = (\mathbf{B}^{-1} + \mathbf{H} \mathbf{K}_y^{-1} \mathbf{H}^T)^{-1} (\mathbf{H} \mathbf{K}_y^{-1} \mathbf{y} + \mathbf{B}^{-1} \mathbf{b})$, and $\mathbf{R} = \mathbf{H}_* - \mathbf{H} \mathbf{K}_y^{-1} \mathbf{K}_*$. The limit of Eq. 33 as the prior on the $\boldsymbol{\beta}$ parameter is not defined, i.e., $\mathbf{B}^{-1} \rightarrow \mathbf{0}$ becomes an array of zeros, which leads to a predictive distribution that is independent of the term \mathbf{b} . Limiting the value of $\bar{\boldsymbol{\beta}} = (\mathbf{H} \mathbf{K}_y^{-1} \mathbf{H}^T)^{-1} \mathbf{H} \mathbf{K}_y^{-1} \mathbf{y}$, we have

$$\begin{aligned} \bar{\mathbf{g}}(X_*) &= \bar{\mathbf{f}}(X_*) + \mathbf{R}^T \bar{\boldsymbol{\beta}} \\ \text{cov}(\mathbf{g}_*) &= \text{cov}(\mathbf{f}_*) + \mathbf{R}^T (\mathbf{H} \mathbf{K}_y^{-1} \mathbf{H}^T)^{-1} \mathbf{R} \end{aligned} \quad (34)$$

Similar to the case of zero-mean GPR, the marginal likelihood for the model with non zero-mean can be expressed as

$$\begin{aligned} \log p(\mathbf{y} | \mathbf{X}, \mathbf{b}, \mathbf{B}) &= -\frac{1}{2} (\mathbf{H}^T \mathbf{b} - \mathbf{y})^T (\mathbf{K}_y + \mathbf{H}^T \mathbf{B} \mathbf{H})^{-1} (\mathbf{H}^T \mathbf{b} - \mathbf{y}) \\ &= -\frac{1}{2} \log |\mathbf{K}_y + \mathbf{H}^T \mathbf{B} \mathbf{H}| - \frac{\mathbf{n}}{2} \log 2\pi \end{aligned} \quad (35)$$

For estimating the limits where $\mathbf{B}^{-1} \rightarrow \mathbf{0}$. Substituting the mean equals to zero, $\mathbf{b} = \mathbf{0}$, we get

$$\begin{aligned} \log p(\mathbf{y} | \mathbf{X}, \mathbf{b} = \mathbf{0}, \mathbf{B}) &= -\frac{1}{2} \mathbf{y}^T \mathbf{K}_y^{-1} \mathbf{y} + \frac{1}{2} \mathbf{y}^T \mathbf{C} \mathbf{y} \\ \log p(\mathbf{y} | \mathbf{X}, \mathbf{b} = \mathbf{0}, \mathbf{B}) &= -\frac{1}{2} \log |\mathbf{K}_y| - \frac{1}{2} \log |\mathbf{B}| - \frac{1}{2} \log |\mathbf{A}| - \frac{\mathbf{n}}{2} \log 2\pi \end{aligned} \quad (36)$$

with $\mathbf{A} = \mathbf{B}^{-1} + \mathbf{H} \mathbf{K}_y^{-1} \mathbf{H}^T$ and $\mathbf{C} = \mathbf{K}_y^{-1} \mathbf{H}^T \mathbf{A}^{-1} \mathbf{H} \mathbf{K}_y^{-1}$

Eq. 36, is composed of three expressions: a quadratic expression in \mathbf{y} , a log determinant expression, and expression with $\log 2\pi$. After performing an eigen decomposition of the covariance matrix, we observed that the contributions of the quadratic expression to the infinite-variance directions becomes zero. In contrast, the contribution of log determinant expression tends to minus infinity. On projecting \mathbf{y} in a direction orthogonal to the span of \mathbf{H}^T and computing the marginal likelihood, we get the standard solution (Wahba, 1985; Ansley & Kohn,

1985). Considering m be the rank of H^T then the term $-\frac{1}{2}\log|B| - \frac{m}{2}\log 2\pi$ in Eq. (36) can be ignored (Ansley & Kohn, 1985) and the expression we get is

$$\log p(y|X) = -\frac{1}{2}y^T K_y^{-1}y + \frac{1}{2}y^T C y - \frac{1}{2}\log|K_y| - \frac{1}{2}\log|A| - \frac{n-m}{2}\log 2\pi \quad (37)$$

with, $A = HK_y^{-1}H^T$ and $C = K_y^{-1}H^T A^{-1}HK_y^{-1}$.

In this study we have used the squared-exponential co-variance function, k_y as

$$k_y(x_p, x_q) = \sigma_f^2 \exp\left(-\frac{1}{2l_s^2}(x_p - x_q)^2\right) + \sigma_n^2 \delta_{pq} \quad (38)$$

where l_s represents the length scale, σ_f^2 represents the signal variance and σ_n^2 represents the noise variance. All these parameters can be varied.

The performance of GPR depends on the features and scaling method. In this study, we have selected six features, namely the number of nodes, sensing range, SVIR, MSNR, angle of the intrusion path and required k . All these features are extracted using the analytical approach, as discussed in Section 3.2. The scaling method in GPR depends on the characteristic of the problem. In this study, we proposed three methods, as shown in Fig. 6. The method I is S-GPR (Scaling GPR). In this method, the features are scaled using Eq. 39.

$$x_s = \frac{x}{\sigma} \quad (39)$$

where, x represent the features, x_s represents the standardised features, and σ is the standard deviation of the features. The second method, method II, C-GPR (Center mean GPR). Here, the features are scaled using Eq. 40.

$$x_s = x - \bar{x} \quad (40)$$

where, \bar{x} represents the mean of the features. The last method, method III, is the native GPR method in which the original features are feed to train and test the GPR. In this study, the dimension of the features and predictand data set is 542×7 . We have divided the data set in 75:25 ratio for training and testing of the GPR models. Hence, we have used 406×7 data set for training and 136×7 for testing.

4. Simulation experiment

4.1. Simulation for k -barrier coverage probability

We validate the given analytical model by Monte-Carlo simulations using the value of different simulation parameters given in Table 1. At first, we consider a rectangular region of 1000×500 meter square area

Table 1
Simulation parameters for k -barrier coverage probability.

Parameter	Value (s)
Rectangular region	$l = 1000$ m and $w = 500$ m
Sensing range of nodes (r_s)	(5–30) m
Number of sensor nodes (N)	10–1500
Maximum sensor (v_s)/intruder speed (v_i)	(10–30) m/sec
Sensor to Intruder Velocity Ratio (SIVR)	1:1, 1:2, 1:3, 2:1, 3:1
Sensing time (τ)	10 sec
Mobility Model	Random direction mobility model
Sensing range model	Binary sensing range model
Value of required k	5–50
Node distribution model	Uniform Random Distribution
Intrusion path angle (θ)	$[0, \arctan(\frac{l}{w})]$

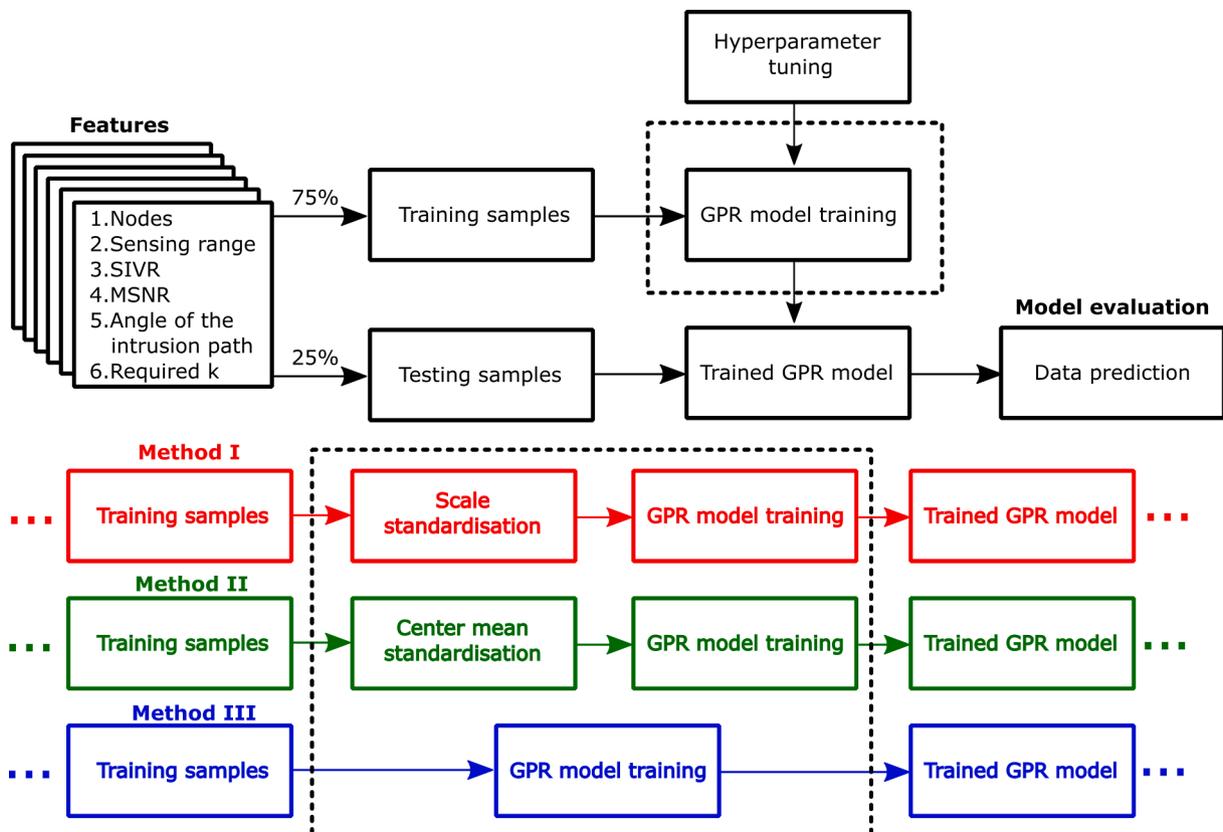


Fig. 6. Flowchart of the methodology.

and deploy N number of homogeneous mobile sensor nodes uniformly and independently inside the RoI. All the mobile sensor nodes travel with a given speed and direction within the boundaries of the RoI. Similarly, an intruder is taken to be a dot-like object and is characterised by its speed and the angle at which it may move to cross the border area. These mobile sensors try to build a barrier in such a manner that every possible route that an intruder might take have at least k distinct sensor monitoring it. Therefore, an intruder is taken to be discovered by a sensor node if and only if it falls inside the coverage area of that sensor node. We evaluate the performance of the model in terms of k -barrier coverage probability denoted by $P(\Phi \geq k)$.

4.2. GPR simulation for k -barrier coverage probability

In the GPR simulation, we have three free hyperparameters, namely characteristic length scale, signal standard deviation and noise standard deviation. The covariance function is parameterised in terms of these hyperparameters. In this study, we have used the square exponential kernel as the covariance (or kernel) function. This covariance function has separate characteristic length scales for each predictor. In this study, these length scales are defined by the standard deviation of the predictors or features. Similarly, the other two hyperparameters are tuned through statistical parameters and their values/methods are listed in Table 2.

In this study, we have used a constant explicit basis function. This basis function will add the term $H * \beta$ to the model, where H represents the basis matrix, and β represents the basis coefficients.

5. Results

In this section, we have presented the performance of all the three methods (*i.e.*, Method I, Method II, and Method III) in predicting the k -barrier coverage probability. We have plotted a linear regression curve between the simulated and GPR predicted k -barrier coverage probability.

5.1. Performance of the Method I

We have compared the simulated k -barrier coverage probability with the predicted response of the Method I. In doing so, we observed that the predicted response is fairly correlated with the simulated results with $R = 0.64$ and $RMSE = 0.137$ (Fig. 7a). The variations are not well accorded because of the mild scattering. Further, to determine whether the linear model fits the data well, we have plotted the residual plot (Fig. 7b). The residual plot seems to be randomly scattered without following any pattern, and hence this linear regression plot can be considered as a good fit.

Table 2
Simulation parameters for GPR model.

Parameters	Values/methods	Description
Kernel (Covariance) Function	Squared exponential kernel	It has a separate length scale per predictor
length scale (l_s)	$\text{std}(X_{\text{train}})$	It is the standard deviation of the predictors
Signal standard deviation ($\sigma_f > 0$)	$\text{std}(Y_{\text{train}})$	It is the standard deviation of the predictand
Noise standard deviation ($\sigma_n > 0$)	$[10^{-4}, \max(10^{-3}, 10 \cdot \text{std}(Y_{\text{train}}))]$	It is a real value between the range
Basis Function	Constant	With basis matrix $H = 1; n \times 1$ vector of 1's.
Computation method	QR factorization	This methods provides better accuracy for computing log likelihood and gradient

5.2. Performance of the Method II

In this sub-section, we have compared the simulated k -barrier coverage probability with the predicted response of Method II. On comparing, we found that the predicted response is highly correlated with the simulated results and gathered along the regression line with $R = 0.79$ and $RMSE = 0.108$ (Fig. 8a). Also, the variations are well accorded.

Further, after plotting the residual plot (Fig. 8b), we found that the residuals are randomly scattered. This suggests that the linear plot is a good fit.

5.3. Performance of the Method III

Lastly, in this sub-section, we have compared the simulated k -barrier coverage probability with the predicted response of Method III. We observed a prodigiously good agreement between both with $R = 0.85$ and $RMSE = 0.095$ (Fig. 9a). In this case, also, the variations are well accorded. Finally, the residual plot is randomly scattered and hence suggesting the linear model is a good fit.

6. Discussion

In this section, we have discussed the time complexity of the proposed methods and compared the results with three different scenarios of the Monte Carlo simulation (Fig. 10). In doing so, we observed no obvious difference in the time complexity of all the proposed methods (*i.e.*, I, II, and III) with the method I having least time complexity followed by method III and method II. However, we found a significant difference when these results are compared with the Monte Carlo simulation for node density 100, 200, and 300. All other parameters are kept constant. The time complexity of the Monte Carlo simulation increases with increasing the node density. Hence, our proposed methods are very efficient with respect to the time complexity. The time complexity of GPR is $\mathcal{O}(N^3)$, where N represents the training samples. The time complexity increases significantly if N is greater than ten thousand, which is generally encountered with geospatial data (big data). In WSNs arena, we usually do not encounter such high training samples. Hence, GPR based machine learning models can be used without much concern with respect to the time complexity. However, to further reduce down the time complexity, various sparse GPR methods have been proposed (Quiñero-Candela & Rasmussen, 2005; Gu & Hu, 2012). These modified versions of GPR alleviate the time complexity further and much useful while working with big data.

Various other studies have been reported for improving the accuracy of intrusion detection based on fuzzy rule-based systems and ANN (Mittal, Saraswat, Iwendi, & Anajemba, 2019; Batiha, Prauzek, & Krömer, 2020). A detailed comparative analysis of the performance of various machine learning approach can be found in Baraneetharan (2020). All these studies have also reported a high accuracy for intrusion detection in WSNs. However, to ensure a fair evaluation, we need to compare the results of GPR models with other regression-based machine learning (Singh et al., 2020). In doing so, we have compared the results obtained through GPR with the corresponding variant of the benchmark algorithm Support Vector Regression (SVR) over the same data set (Singh et al., 2020). The corresponding variant of SVR are Scaling-SVR (S-SVR), Center mean-SVR (C-SVR), and SVR as shown in (Table 3). We have used R , $RMSE$, and time complexity as comparison metrics. On comparing, we found that method III has the best, and S-SVR has the worst R values. Also, the $RMSE$ is the lowest in method III as compared to the methods with high R values. Overall, method III has a good R with relatively lower $RMSE$ and with reasonable time complexity.

7. Conclusion

In this paper, we have presented a comprehensive framework for the

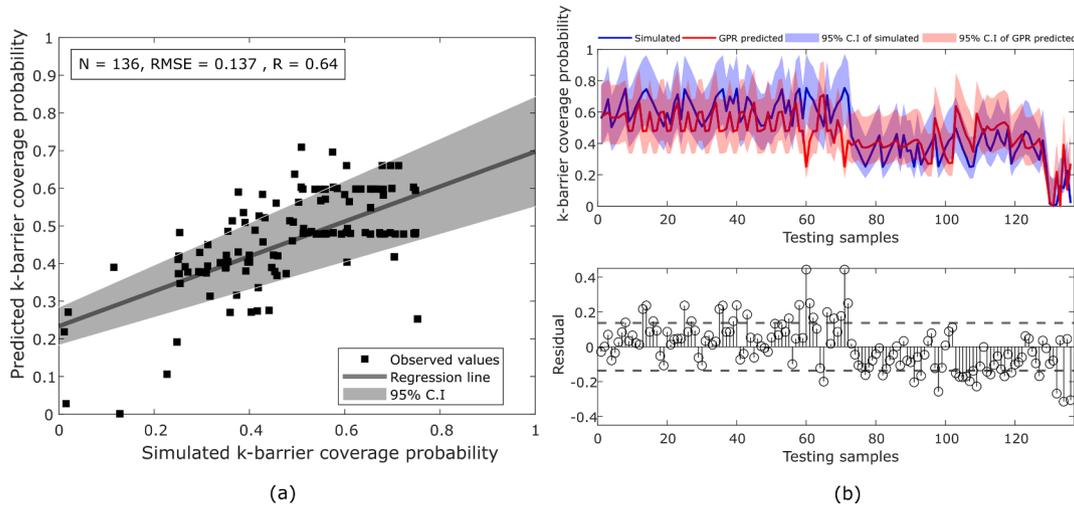


Fig. 7. Prediction results for k -barrier coverage probability using Method I. Fig. 7a shows the linear regression plot between the simulated and predicted response. The grey color represents the 95% C.I of the regression line. Fig. 7b shows the corresponding residual plot. The dashed line in the diagram represents RMSE.

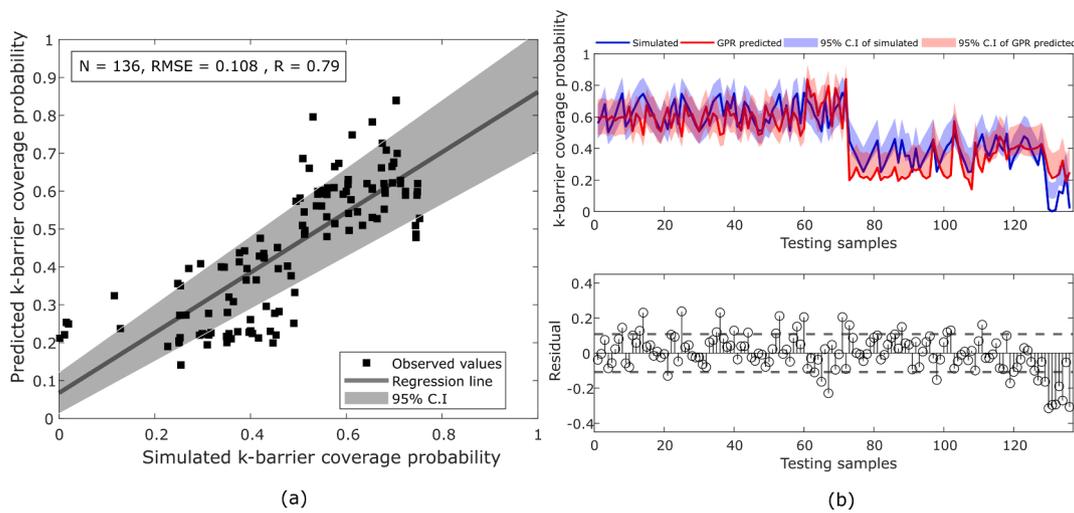


Fig. 8. Prediction results for k -barrier coverage probability using method II. Fig. 8a shows the linear regression plot between the simulated and predicted response. The grey color represents the 95% C.I of the regression line. Fig. 8b shows the corresponding residual plot. The dashed line in the diagram represents RMSE.

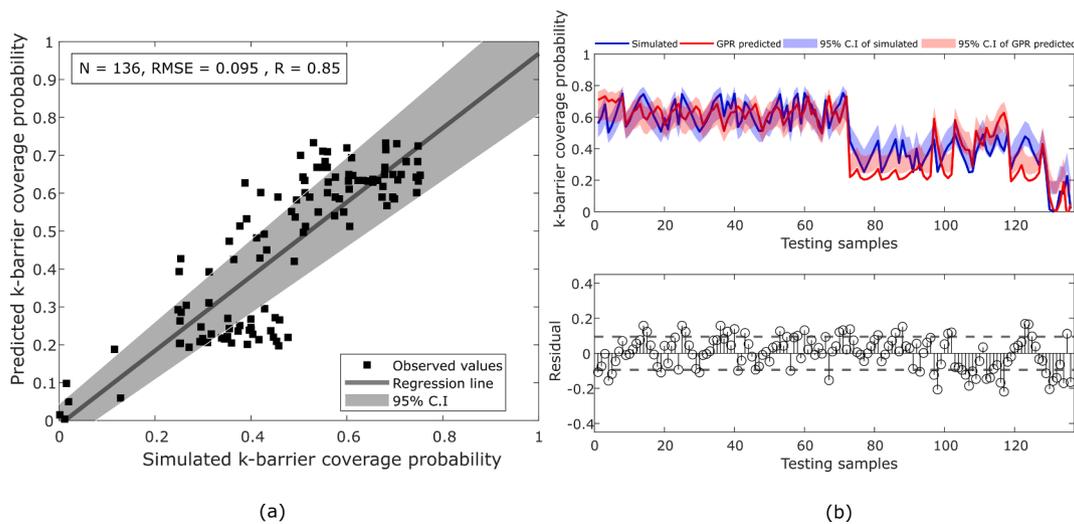


Fig. 9. Prediction results for k -barrier coverage probability using method III. Fig. 9a shows the linear regression plot between the simulated and predicted response. The grey color represents the 95% C.I of the regression line. Fig. 9b shows the corresponding residual plot. The dashed line in the diagram represents RMSE.

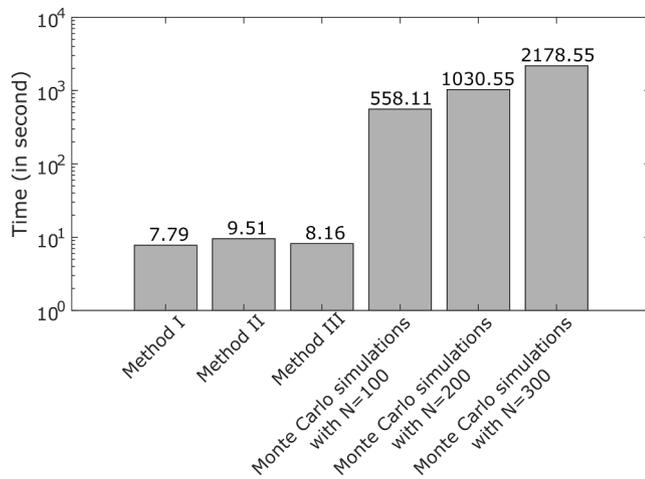


Fig. 10. Comparison of the computation time complexity for method I, II, III with three different scenarios of the Monte Carlo simulations. In this figure, the time axis is in log scale.

Table 3

Comparison of the proposed methods with the benchmark.

Parameters	Methods					
	(S-GPR)	C-GPR	(GPR)	S-SVR	C-SVR	SVR
R	0.64	0.79	0.85	0.25	0.27	0.28
RMSE	0.137	0.108	0.095	0.051	0.068	0.045
Time (s)	7.79	9.51	8.16	3.36	3.50	3.37

accurate prediction of k -barrier coverage probability. We have proposed three GPR based machine learning models based on the data standardisation. We trained all the three models using the squared exponential kernel. Afterwards, we have evaluated and compared the performance of all the three models. To ensure a fair comparison, we have selected R, RMSE, and time complexity parameters as metrics. On Comparing, we observed that the native GPR model (*i.e.*, method III) results in the highest R with relatively lower RMSE. Further, method III outperforms the corresponding SVR variants in terms of accuracy. Hence, this method can be used for accurate prediction of k -barrier coverage probability with less time complexity.

Code availability

The code for this work will be made available on a reasonable request to the corresponding author.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

We would like to acknowledge IISER Bhopal, IIT Kharagpur, Gautam Buddha University Greater Noida, for providing institutional support. We thank the editor and all the anonymous reviewers for providing helpful comments and suggestions.

References

- Ali, S., Qaisar, S. B., Saeed, H., Khan, M. F., Naeem, M., & Anpalagan, A. (2015). Network challenges for cyber physical systems with tiny wireless devices: A case study on reliable pipeline condition monitoring. *Sensors*, *15*, 7172–7205.
- Amutha, J., Sharma, S., & Nagar, J. (2020). Wsn strategies based on sensors, deployment, sensing models, coverage and energy efficiency: Review, approaches and open issues. *Wireless Personal Communications*, *111*, 1089–1115.
- Ansley, C. F. & Kohn, R. (1985). Estimation, filtering, and smoothing in state space models with incompletely specified initial conditions. *The Annals of Statistics*, (pp. 1286–1316).
- Aponte-Luis, J., Gómez-Galán, J. A., Gómez-Bravo, F., Sánchez-Raya, M., Alcina-Espigado, J., & Teixido-Rovira, P. M. (2018). An efficient wireless sensor network for industrial monitoring and control. *Sensors*, *18*, 182.
- Assad, N., Elbhiri, B., Faqih, M. A., Ouadou, M., & Aboutajdine, D. (2016). Efficient deployment quality analysis for intrusion detection in wireless sensor networks. *Wireless Networks*, *22*, 991–1006.
- Baraneetharan, E. (2020). Role of machine learning algorithms intrusion detection in wsns: A survey. *Journal of Information Technology*, *2*, 161–173.
- Batiha, T., Prauzek, M., & Krömer, P. (2020). Intrusion detection in wireless sensor networks by an ensemble of artificial neural networks. In *Intelligent decision technologies 2019* (pp. 323–333). Springer.
- Camp, T., Boleng, J., & Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, *2*, 483–502.
- Dey, N., Ashour, A. S., Shi, F., Fong, S. J., & Sherratt, R. S. (2017). Developing residential wireless sensor networks for ecg healthcare monitoring. *IEEE Transactions on Consumer Electronics*, *63*, 442–449.
- Friedman, J.H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, (pp. 1189–1232).
- Ghosh, K., Neogy, S., Das, P. K., & Mehta, M. (2018). Intrusion detection at international borders and large military barracks with multi-sink wireless sensor networks: an energy efficient solution. *Wireless Personal Communications*, *98*, 1083–1101.
- Gu, D., & Hu, H. (2012). Spatial gaussian process regression with mobile sensor networks. *IEEE Transactions on Neural Networks and Learning Systems*, *23*, 1279–1290.
- Han, L., Zhou, M., Jia, W., Dalil, Z., & Xu, X. (2019). Intrusion detection model of wireless sensor networks based on game theory and an autoregressive model. *Information Sciences*, *476*, 491–504.
- Hassan, M. M., Gumael, A., Alsanad, A., Alrubaiyan, M., & Fortino, G. (2020). A hybrid deep learning model for efficient intrusion detection in big data environment. *Information Sciences*, *513*, 386–396.
- He, H. & Siu, W. -C. (2011). Single image super-resolution using gaussian process regression. In *CVPR 2011* (pp. 449–456). IEEE.
- He, S., Chen, J., Li, X., Shen, X., & Sun, Y. (2013). Mobility and intruder prior information improving the barrier coverage of sparse sensor networks. *IEEE Transactions on Mobile Computing*, *13*, 1268–1282.
- Jawad, H. M., Nordin, R., Gharghan, S. K., Jawad, A. M., & Ismail, M. (2017). Energy-efficient wireless sensor networks for precision agriculture: A review. *Sensors*, *17*, 1781.
- Keung, G. Y., Li, B., & Zhang, Q. (2012). The intrusion detection in mobile sensor network. *IEEE/ACM Transactions on Networking*, *20*, 1152–1161.
- Kumar, S., Dutttagupta, S., Rangan, V. P., & Ramesh, M. V. (2020). Reliable network connectivity in wireless sensor networks for remote monitoring of landslides. *Wireless Networks*, *26*, 2137–2152.
- Liu, B., Brass, P., Dousse, O., Nain, P., & Towsley, D. (2005). Mobility improves coverage of sensor networks. In *Proceedings of the 6th ACM international symposium on mobile ad hoc networking and computing* (pp. 300–308).
- Mittal, M., Saraswat, L. K., Iwendi, C., & Anajemba, J. H. (2019). A neuro-fuzzy approach for intrusion detection in energy efficient sensor routing. In *2019 4th international conference on internet of things: Smart innovation and usages (IoT-SIU)* (pp. 1–5). IEEE.
- Mohapatra, S. K., Sahoo, P. K., & Wu, S.-L. (2016). Big data analytic architecture for intruder detection in heterogeneous wireless sensor networks. *Journal of Network and Computer Applications*, *66*, 236–249.
- Nagar, J., Chaturvedi, S. K., & Soh, S. (2020). An analytical model to estimate the performance metrics of a finite multihop network deployed in a rectangular region. *Journal of Network and Computer Applications*, *149*, Article 102466.
- Nagar, J., & Sharma, S. (2018). k -barrier coverage-based intrusion detection for wireless sensor networks. In *Cyber security* (pp. 373–385). Springer.
- Nancy, P., Muthurajkumar, S., Ganapathy, S., Kumar, S. S., Selvi, M., & Arputharaj, K. (2020). Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks. *IET Communications*, *14*, 888–895.
- Nguyen, M. T., & Kim, K. (2020). Genetic convolutional neural network for intrusion detection systems. *Future Generation Computer Systems*, *113*, 418–427.
- Noel, A. B., Abdaoui, A., Elfouly, T., Ahmed, M. H., Badawy, A., & Shehata, M. S. (2017). Structural health monitoring using wireless sensor networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *19*, 1403–1423.
- O'Hagan, A. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society: Series B (Methodological)*, *40*, 1–24.
- Østergård, T., Jensen, R. L., & Maagaard, S. E. (2018). A comparison of six metamodelling techniques applied to building performance simulations. *Applied Energy*, *211*, 89–103.
- Otoun, S., Kantarci, B., & Moutafah, H. T. (2019). On the feasibility of deep learning in sensor network intrusion detection. *IEEE Networking Letters*, *1*, 68–71.
- Quiñero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, *6*, 1939–1959.

- Richter, P., & Toledano-Ayala, M. (2015). Revisiting gaussian process regression modeling for localization in wireless sensor networks. *Sensors*, *15*, 22587–22615.
- Riyaz, B., & Ganapathy, S. (2020). A deep learning approach for effective intrusion detection in wireless networks using cnn. *Soft Computing*, 1–14.
- Saha, S., Saha, B., Saxena, A., & Goebel, K. (2010). Distributed prognostic health management with gaussian process regression. In *2010 IEEE aerospace conference* (pp. 1–8). IEEE.
- Sharma, A., & Chauhan, S. (2020). Sensor fusion for distributed detection of mobile intruders in surveillance wireless sensor networks. *IEEE Sensors Journal*.
- Sharma, S., & Nagar, J. (2020). Intrusion detection in mobile sensor networks: A case study for different intrusion paths. *Wireless Personal Communications*, 1–21.
- Singh, A., Kotiyal, V., Sharma, S., Nagar, J., & Lee, C.-C. (2020). A machine learning approach to predict the average localization error with applications to wireless sensor networks. *IEEE Access*, *8*, 208253–208263.
- Singh, A., Sharma, S., & Singh, J. (2021). Nature-inspired algorithms for wireless sensor networks: A comprehensive survey. *Computer Science Review*, *39*, Article 100342.
- Singh, A., Sharma, S., Singh, J., & Kumar, R. (2019). Mathematical modelling for reducing the sensing of redundant information in wsns based on biologically inspired techniques. *Journal of Intelligent & Fuzzy Systems*, *37*, 6829–6839.
- Sun, S., & Xu, X. (2010). Variational inference for infinite mixtures of gaussian processes with applications to traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, *12*, 466–475.
- Tan, X., Su, S., Huang, Z., Guo, X., Zuo, Z., Sun, X., & Li, L. (2019). Wireless sensor networks intrusion detection based on smote and the random forest algorithm. *Sensors*, *19*, 203.
- Wahba, G. (1985). A comparison of gcv and gml for choosing the smoothing parameter in the generalized spline smoothing problem. *The Annals of Statistics*, (pp. 1378–1402).
- Wang, W., Huang, H., Li, Q., He, F., & Sha, C. (2020). Generalized intrusion detection mechanism for empowered intruders in wireless sensor networks. *IEEE Access*, *8*, 25170–25183.
- Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian processes for machine learning* (Vol. 2). MA: MIT press Cambridge.
- Yabansu, Y. C., Isakov, A., Kapustina, A., Rajagopalan, S., & Kalidindi, S. R. (2019). Application of gaussian process regression models for capturing the evolution of microstructure statistics in aging of nickel-based superalloys. *Acta Materialia*, *178*, 45–58.