sensors

MDPI

Article

LT-FS-ID: Log-Transformed Feature Learning and Feature-Scaling-Based Machine Learning Algorithms to Predict the *k*-Barriers for Intrusion Detection Using Wireless Sensor Network

Abhilash Singh [1], J. Amutha [2], Jaiprakash Nagar [3], Sandeep Sharma [4,*] and Cheng-Chi Lee [5,6,*]

1 Fluvial Geomorphology and Remote Sensing Laboratory, Indian Institute of Science Education and Research Bhopal, Bhopal 462066, India; sabhilash@iiserb.ac.in or abhilash.iiserb@gmail.com
2 Department of Electronics and Communication Engineering, School of ICT, Gautam Buddha University, Greater Noida 201312, India; roniamutha@gmail.com
3 Subir Chowdhury School of Quality and Reliability, Indian Institute of Technology, Kharagpur 721302, India; jpnagar91@gmail.com
4 Deprtment of Electronics Engineering, Madhav Institute of Technology and Science, Gwalior 474005, India
5 Department of Library and Information Science, Research and Development Center for Physical Education, Health, and Information Technology, Fu Jen Catholic University, New Taipei 242, Taiwan
6 Department of Photonics and Communication Engineering, Asia University, Taichung 41354, Taiwan
* Correspondence: sandeepsvce@gmail.com (S.S.); cclee@mail.fju.edu.tw (C.-C.L.)

Abstract: The dramatic increase in the computational facilities integrated with the explainable machine learning algorithms allows us to do fast intrusion detection and prevention at border areas using Wireless Sensor Networks (WSNs). This study proposed a novel approach to accurately predict the number of barriers required for fast intrusion detection and prevention. To do so, we extracted four features through Monte Carlo simulation: area of the Region of Interest (RoI), sensing range of the sensors, transmission range of the sensor, and the number of sensors. We evaluated feature importance and feature sensitivity to measure the relevancy and riskiness of the selected features. We applied log transformation and feature scaling on the feature set and trained the tuned Support Vector Regression (SVR) model (i.e., LT-FS-SVR model). We found that the model accurately predicts the number of barriers with a correlation coefficient (R) = 0.98, Root Mean Square Error (RMSE) = 6.47, and bias = 12.35. For a fair evaluation, we compared the performance of the proposed approach with the benchmark algorithms, namely, Gaussian Process Regression (GPR), Generalised Regression Neural Network (GRNN), Artificial Neural Network (ANN), and Random Forest (RF). We found that the proposed model outperforms all the benchmark algorithms.

Keywords: WSNs; intrusion detection; machine learning; feature learning; support vector regression

Citation: Singh, A.; Amutha, J.; Nagar, J.; Sharma, S.; Lee, C.-C. LT-FS-ID: Log-Transformed Feature Learning and Feature-Scaling-Based Machine Learning Algorithms to Predict the *k*-Barriers for Intrusion Detection Using Wireless Sensor Network. *Sensors* **2022**, *22*, 1070. https://doi.org/10.3390/s22031070

Academic Editor: Antonio Guerrieri

Received: 11 December 2021
Accepted: 27 January 2022
Published: 29 January 2022

1. Introduction

These days, security is one of the primary concerns for every nation caused by highly unpredictable and noxious events taking place across the globe [1–3]. Every nation wants to secure and protect its borders from any kind of intrusion and attack by enemy forces. In addition, unauthorised and illegal entry is another vital matter that requires immediate attention from the concerned authorities [4]. In order to protect their international borders from enemies and unfriendly forces, several nations have their regular armies. These army soldiers patrol along the border stretches, but patrolling methods are conventional, periodic, and limited. Since a country may have international boundaries that are thousands of miles long, it is impossible to deploy soldiers at every single location. Consequently, there remains a large area along the international borders that is unguarded. Enemies may take advantage of these unguarded locations and enter the territories. They can likely steal some classified documents crucial to the security of a nation, decimate defence personnel, or

demolish crucial infrastructures. The surveillance and monitoring along the international borders and checkpoints can be achieved with the help of WSNs.

WSNs is a widely accepted and renowned technology because it is cheap, readily available, and can be installed on the fly in almost no time at any place [5,6]. In addition, WSNs consist of small and homogeneous sensors that work in a de-centralised fashion requiring no pre-installed foundation and communicating over wireless channels [7]. Therefore, WSNs are employed for many civilian and military applications such as precision agriculture, health monitoring, structural health monitoring, industrial monitoring, disaster management, rescue operations, wild animal monitoring, landslide monitoring, fire detection, monitoring and surveillance in border areas, and many more [8–11]. Furthermore, intrusion detection in border areas and unauthorised access detection in restricted areas and infrastructures is a pivotal application of WSNs. For example, a WSN can be deployed to form a sensor barrier for any possible intrusion path as shown in Figure 1. The studies conducted so far on intrusion detection issues can be divided into two categories; first, it is described as a monitoring or surveillance system to detect an invader or an unauthorised entry in the RoI. Secondly, it is assumed to be a component of a WSN system specifically designed and implemented to diagnose compromised and/or vulnerable sensors for avoiding false alarms and ensuring correct network behaviour [12]. In this work, we concentrate on the first category.

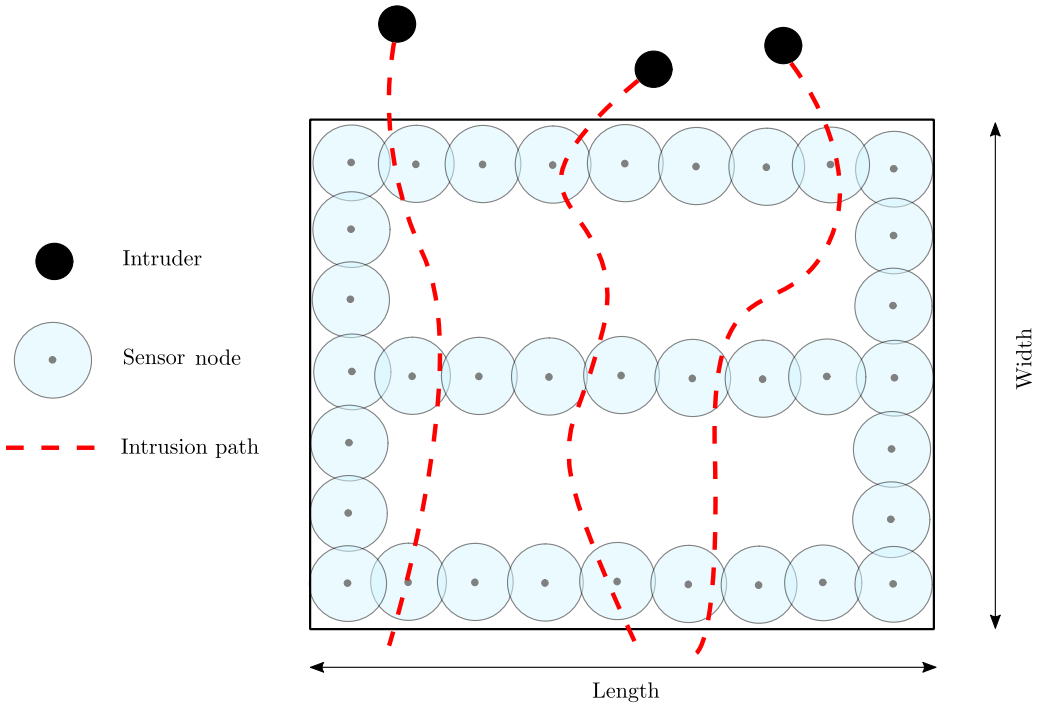

Figure 1. Illustration of 3-barrier coverage for each intrusion path.

The work presented in [13] proposed a fusion algorithm with three levels of hierarchy to spot a passive mobile intruder. They have employed two crucial modalities, namely the sensing probability model and acoustic signal model, to ascertain the presence of an invader. In addition, the authors have also analysed the influence of the number of sensors, intruder speed on the probability of detection, detection accuracy, and false alarm rate and found that the proposed algorithm outperforms the other fusion algorithms. Another work presented in [14] proposed optimal trajectories for mobile sensors employed for intrusion detection in a given RoI. The proposed trajectories for mobile sensors will maximise the coverage area and reduce energy consumption, which would increase the lifetime of the sensor network, thus providing improved intrusion detection performance. A distributed border surveillance system is proposed in [15], where the performance of the system is estimated in terms of the number of barriers obtained for a possible intrusion path in shadowed

and non-shadowed environmental conditions. The authors found that the number of barriers obtained for shadowed environmental conditions is greater compared with the non-shadowed environmental conditions. Similarly, the work in [16] proposed a smart border surveillance system that uses ultrasonic, passive infra-red, and camera sensors to detect the presence of an intruder. The proposed system is capable of distinguishing between animal and human beings. The system sends an alert message and video streams to the control system as soon as it identifies an intruder. In Ref. [17], the authors have proposed a border surveillance system architecture that renders high energy efficiency and load balancing capabilities, thus, increasing the network lifetime. Furthermore, the proposed methodology needs less maintenance, involves low-cost installation, and delivers enhanced reliability. The authors claim that the proposed system outperforms other available intrusion detection systems and has an enhanced network lifetime. Another work provided in [18] presented an analytical model to detect a mobile intruder using mobile sensor networks. They have obtained an analytical formula to calculate the *k*-barrier coverage probability for an invader trying to cross a rectangular-belt region following a given path. They have also investigated the effect of network parameters such as sensor-to-intruder velocity ratio, sensing range, sensor count, and intrusion path angle on the performance metric. The proposed model is very effective in detecting an intrusion and tracking the enemy movements. Most recently, the authors in [19] proposed a remote surveillance system using robots with CCTV cameras. The authors claim that the proposed work will be useful for border surveillance and internal monitoring.

It is pivotal to mention that the above-discussed works [13–19] contribute significantly in the research domain. However, their models are validated through Monte Carlo simulation, which requires very high computation cost and time. For instance, it requires approximately 15 hours to achieve a single outcome through simulation runs at a given value of parameters. In addition, the simulation time increases exponentially with the increase in the number of sensors, sensing range and other network parameters. This is because of the fact that WSNs produce a large volume of data that requires plenty of time for its processing and analysis. Applications like infiltration in border regions are time-sensitive because a delay in seconds may cause catastrophes. Thus, it is vital to detect any kind of intrusion along the borders and around the prohibited regions as quickly as possible.

The problem at hand can be resolved by employing machine learning approaches that are exceptionally competent for computational time [20,21]. For instance, the work presented in [22] provided a mathematical framework to evaluate the *k*-barrier coverage probability for a given intrusion path using mobile WSNs. The authors have proposed three machine learning models based on the GPR algorithm to predict the *k*-barrier coverage probability to overcome the computational and time complexity problem. In doing so, they have considered sensing range, the number of sensors, sensor to intruder velocity ratio, mobile to static sensor ratio, required value of *k*, and intrusion path angle as potential features. The proposed machine learning model can predict the *k*-barrier coverage probability with higher accuracy than the other benchmark algorithms.

In this study, we proposed an efficacious machine learning-based approach to accurately predict the number of barriers for fast intrusion detection and prevention using relevant features. We extracted relevant features (i.e., the area of the RoI, sensing and transmission range of the sensor, and the total number of sensors) synthetically through Monte Carlo simulations. Subsequently, we applied feature transformation and scaling operations and trained a SVR model. We access the performance of the trained model by using R, RMSE, bias, and computational time complexity as the performance metrics. The main contributions of this paper are as follows:

- We introduced a synthetic data generation framework for a cost-effective solution.
- We estimated the relative importance score of each feature by using the regression tree ensemble approach.

- We performed the sensitivity analysis of the features using Partial Dependency Plot (PDP) analysis.
- We proposed a novel algorithm based on log-transformed feature learning and feature-scaling to accurately predict the number of barriers for fast intrusion detection and prevention. We also performed a sensitivity analysis of the proposed algorithm.

2. Material and Methods

2.1. Preparation of the Datasets

The performance of any machine learning model depends on the quality of datasets on which it is trained [23]. These datasets can either be field derived (obtained by direct measurements) or generated synthetically (obtained through simple rules, statistical modelling, and simulations) [24]. The use of synthetic data is increasing exponentially in the domain of healthcare [25,26], WSNs [22,27], and data privacy [28].

In this study, we extracted the datasets synthetically through simulations. To do so, we consider a finite number of sensors (N), distributed uniformly and randomly in a rectangular RoI. Each sensor is assumed to be homogeneous, i.e., sensing, transmission, and computational capabilities are identical for each sensor. The dimensions of the network deployment RoI are varied from 100×50 m^2 to 250×200 m^2. The entire dataset used for training and testing purposes is obtained through simulations using network simulator NS-2.35. The complete procedure for simulation outcomes is explained below.

Any two arbitrary sensors in the deployed WSN can communicate with each other, if they satisfy the condition, $R_{tx} \geq 2R_s$, where, R_{tx} and R_s indicates the transmission and sensing range of sensors respectively. Here, we have considered the most widely employed sensing range model known as the Binary Sensing Model (BSM) to estimate the performance of WSNs. According to BSM [29], a random sensor can detect a target with probability equal to one, if the target falls within the sensing range R_s of the sensor denoted by S_i. Otherwise, the target detection probability will be equal to zero. Mathematically, it can be represented by Equation (1).

$$P_{det} = \begin{cases} 1, \ if \ d(S_i, P) \leq R_s \\ 0, \ otherwise \end{cases} \tag{1}$$

where $d(S_i, P)$ represents the Euclidean distance between the sensor S_i and target point P. To identify the existence of intruders, a barrier is formed by connecting a sensor cluster over the entire RoI. To detect an intruder successfully, there should be at least one barrier for each possible intrusion path to ensure barrier coverage. The total number of sensors required to achieve the desired k-barrier coverage can be computed by $k = \lceil L/2R_s \rceil$ [1] and the maximum Barrier Paths (BP_{max}) that can be constructed for a given intrusion path is computed as: $BP_{max} = \lfloor N/k \rfloor$, where L indicates the length of the rectangular RoI. The k-coverage ensures that each point in the target RoI is monitored by k distinct sensors, where k is a positive integer having value greater than one. Table 1 shows different network parameters and their values used to get the simulation results.

Table 1. Simulation parameters.

Parameters	Values
Simulator	NS-2.35
Network region	Rectangular RoI
Network area (m^2)	100×50 to 250×200
Number of sensors (N)	100 to 400
Sensing range (R_s)	15 to 40 m
Transmission range (R_{tx})	30 to 80 m
Sensor's deployment type	Uniform distribution
Sensing model	Binary sensing model

2.2. Calculation of Feature Importance and Sensitivity

To calculate each feature's relative importance score, we created a regression ensemble through boosting ensemble learning. We leverage LSBoost (Least Square gradient Boosting) algorithm to boost hundred regression trees, each having unity learning rate [22,30]. This algorithm assumes each decision tree as a weak learner and processes them individually by identifying their weak points. Afterward, the algorithm process the next weak learner by concentrating on the weak aspect of the previous learner. In this way, the algorithm iteratively formed an ensemble of weak learners. Once the ensemble is generated, we calculated the feature importance by summing the total change in the normalised node risk.

Further, we performed the Partial Dependency Plot (PDP) analysis to assess the impact of each individual feature on the predictand. It computes the partial dependency of the considered feature set on the predictand by marginalising the impact of remaining features [27,30]. We considered a set of two features and computed their partial dependency on the predictand. For a set of four features, we have a total of six pairs of features. We plotted the 2D and 3D variation profiles.

2.3. SVR Model Set-Up

In this section, we have discussed the modelling of SVR [31,32] for the prediction of the number of barriers (Figure 2). It is an effective algorithm to address prediction problems, solve sample issues, and provide significant generalisation performance [30,33]. Using a nonlinear mapping $\varphi(.) : \Re^n \rightarrow \Re^{n_h}$, the training sets (x_i, y_i), where $i = 1$ to n, are mapped into a high dimensional feature space, $\Re^{n_h}$. Then, a linear function, f, is used to express the nonlinear association among features and the response variable. The SVR function [34] is a linear function which is represented as:

$$f(x) = w^T \varphi(x) + B \tag{2}$$

where $f(x)$ indicates the forecasting values, $w \in \Re^{n_h}$ indicates the weighting matrix, and $B \in \Re$ indicates the bias term. The SVR approach intends to reduce the empirical risk as:

$$R_{em}(f) = \frac{1}{N} \sum_{i=1}^{N} \Theta_\epsilon(y_i, w^T \varphi(x_i) + B) \tag{3}$$

where $\Theta_\epsilon(y_i, w^T \varphi(x_i) + B)$ indicates the ϵ-insensitive loss function that determines the optimal hyper plane on a high-dimensional feature space to maximise the distance between two subsets of input dataset. It is determined by:

$$\Theta_\epsilon(y_i, w^T \varphi(x_i) + B) = \begin{cases} w^T \varphi(x_i) + B - y_i - \epsilon \; if \; w^T \varphi(x_i) + B - y_i \geq \epsilon \\ 0, \; otherwise \end{cases} \tag{4}$$

Hence, SVR is concerned with identifying the optimal hyper plane and decreasing the residual between the training datasets and the ϵ-insensitive loss function. Moreover, SVR reduces the total errors by:

$$\min_{w,B,\xi^*,\xi} R_\epsilon(w,\xi^*,\xi) = \frac{1}{2}w^T w + C\sum_{i=1}^{N}(\xi_i^* + \xi_i) \tag{5}$$

with the following constraints

$$\begin{aligned} y_i - w^T\varphi(x_i) - B \le \epsilon + \xi_i^*,\ i = 1,2,...,N \\ -y_i + w^T\varphi(x_i) + B \le \epsilon + \xi^i,\ i = 1,2,...,N \\ \xi_i^* \ge 0,\ i = 1,2,...,N \\ \xi_i \ge 0,\ i = 1,2,...,N \end{aligned}$$

Equation (4) normalises weight sizes, ensures regression function flatness, penalises *f(x)* and *y* training residuals by the ϵ-insensitive loss function, and *C* represents the penalty parameter. Training residuals above ϵ are represented as ξ_i^* and below $-\epsilon$ are represented as ξ_i. However, in the dual space, SVR function is represented as:

$$f(x) = \sum_{i=1}^{N}(\beta_i^* - \beta_i)K(x_i, x_j) + B \tag{6}$$

where $K(x_i,x_j)$ represents the kernel function. It is the inner product of x_i and x_j vectors in the feature space $\varphi(x_i)$ and $\varphi(x_j)$, respectively. We have used polynomial kernel (Equation (7)) as it belongs to the group of the non-stationary kernel that performs effectively over standarised and transformed features [35].

$$K(x_i, x_j) = \gamma((x_i \cdot x_j) + 1)^\omega \tag{7}$$

where γ and ω are the kernel function's structural parameter and polynomial degree, respectively. The prediction accuracy of an SVR model is governed by the good tuning of hyperparameters (*C* and ϵ). If the residual between the observed and predicted value is greater than the hyperparameter ϵ then the other hyperparameter *C*, penalises the model. Hence, a high value of *C* results in under-fitting, and a lower value leads to high computational complexity [27].

In this study, we applied the universal grid optimisation algorithm [36] to optimise the hyperparameters. We selected the most frequently used Mean Square Error (MSE) function [37] as the objective function given by:

$$\frac{1}{n}\sum_{i=1}^{n}(f_i - \widehat{f}_i)^2 \tag{8}$$

where *n* is the sampling size, f_i is the observed and $\widehat{f}_i$ is the predicted values. We iteratively optimised *C* for all possible ϵ by considering the MSE function as the objective function. We found the optimal value of *C* = 0.1 and ϵ = 0.01. Afterward, we applied log transformation (LT) [38] and mean z-score scaling (Equation (9)) on the input features. Where x_f is the input feature set, $\overline{x_f}$ is the mean of the feature set, and σ is the standard deviation of the feature set.

$$x_{sf} = \frac{x_f - \overline{x_f}}{\sigma} \tag{9}$$

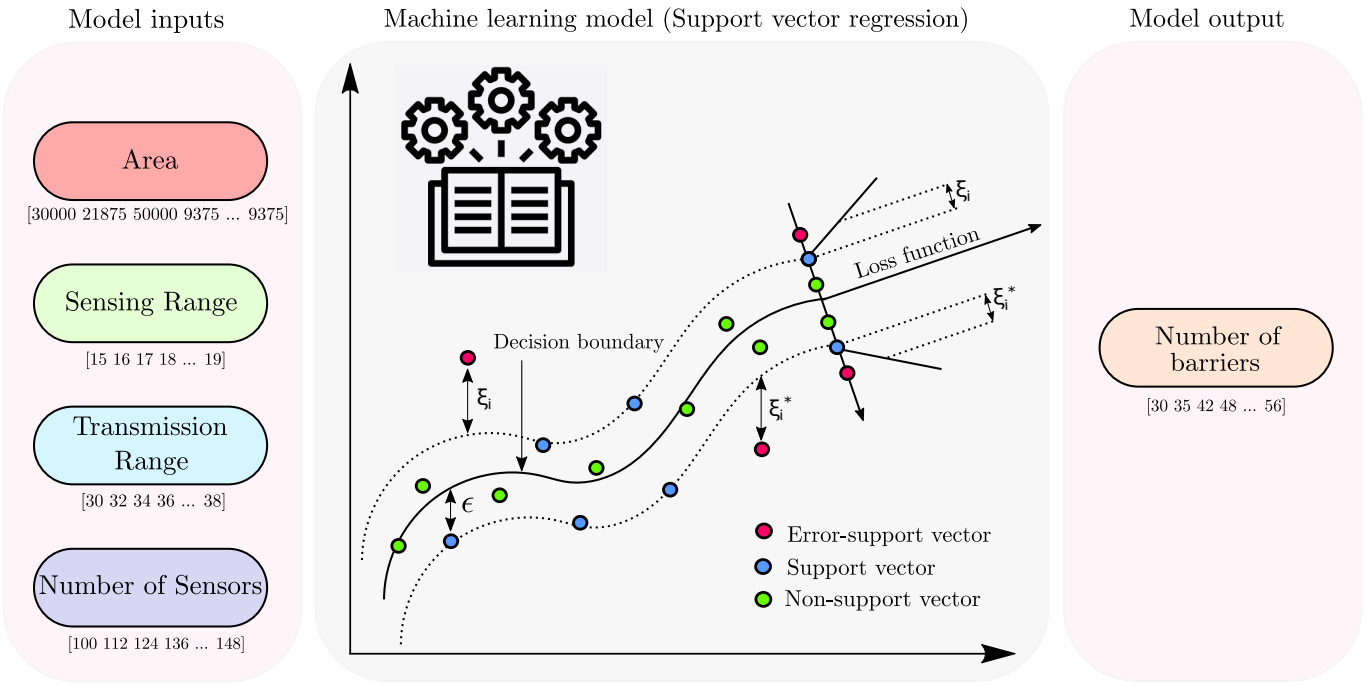

Figure 2. Illustration of the support vector regression with all input features and the corresponding response variable.

Once we applied feature pre-processing, we trained and evaluated the SVR model in an 80:20 ratio. The datasets are divided randomly using Mersenne Twister random generator. We illustrated the complete methodology in Figure 3 and also enumerated the complete process into the following steps;

1. We synthetically generated the input features (i.e., area of the RoI, sensing range of the sensors, transmission range of the sensor, and the number of sensors) through Monte Carlo simulations.
2. We trained a regression tree ensemble to estimate each feature's relative feature importance score.
3. We leverage PDP analysis to perform the sensitivity analysis of each feature.
4. We applied feature scaling on the selected features post log transformation.
5. We used the Mersenne Twister generator with a random seed to randomly divide the datasets for training and testing the model in a ratio of 80:20.
6. We used 80% of the datasets to set up the machine learning model.
7. We used the remaining 20% of the datasets to test the performance of the trained model.
8. We performed the sensitivity analysis of the trained model.
9. We performed the error analysis using error histogram analysis to understand the distribution of the errors.
10. We compared the performance of the trained model with the benchmark algorithms (i.e., ANN, GRNN, GPR, and Random Forest).

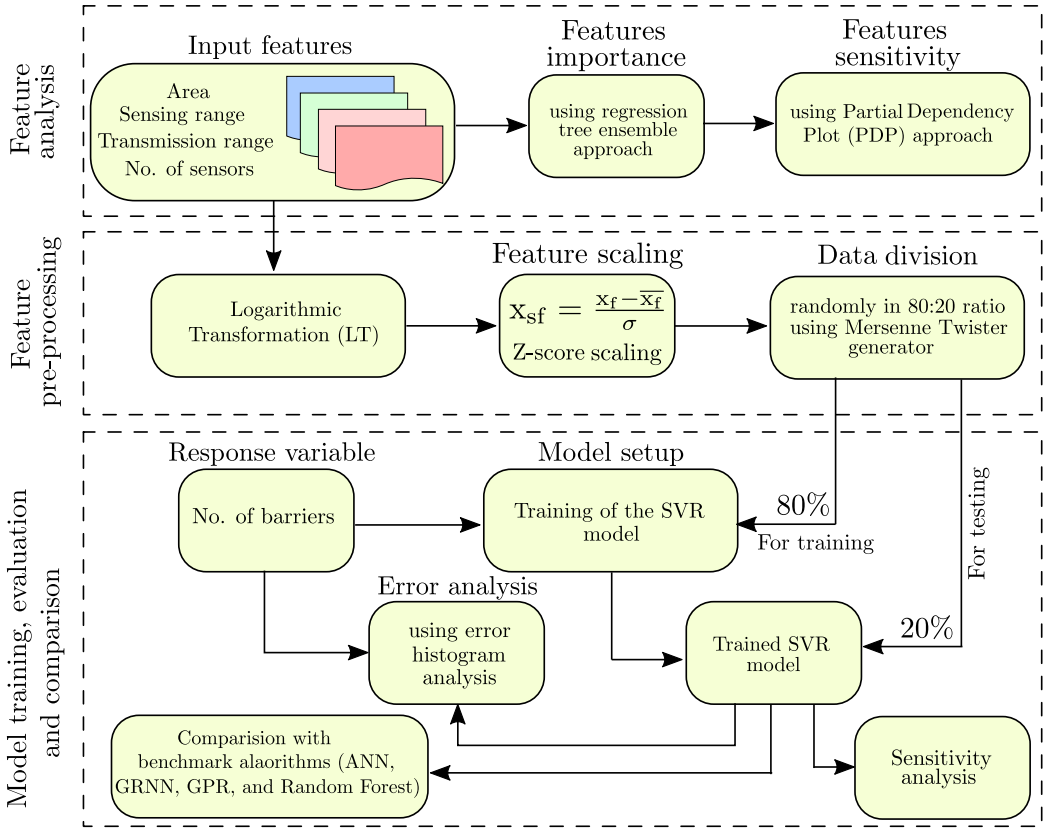

Figure 3. Flowchart of the proposed methodology.

3. Results

In this section, firstly, we discuss the results of feature importance and sensitivity analysis. Afterward, we discuss the performance of the proposed model.

3.1. Feature Importance and Sensitivity

We evaluated the prominence of each feature through the regression tree ensemble approach. The bars in Figure 4 show the relative feature importance score of each feature. The feature importance score of all four features ranges between 60 to 140. We found that the area of the RoI has the least feature importance among all, indicating that area of the rectangular region is the least relevant feature in predicting the number of barriers for fast intrusion detection and prevention. Surprisingly, we found that the sensing range of the sensor, the transmission range of the sensor, and the number of sensors have the same and highest feature importance score, indicating that they are the most relevant features in predicting the number of barriers.

Further, we performed the feature sensitivity analysis of all the four features through the Partial Dependency Plot (PDP) analysis (Figure 5). We observed that area of the rectangular region has a negative repercussion on the response variable (i.e., number of barriers). In contrast, the sensing range of the sensors, the transmission range of the sensors, and the number of sensors have a positive repercussion on the response variable.

3.2. Model Performance

Once our model is trained, we evaluate its performance by using R, RMSE, and bias as the performance metrics. To do so, we fed the testing datasets into the trained model's input and obtained the predicted response from the model. Afterward, we plotted a linear fit line between the observed and predicted response variable in Figure 6a. In doing so, we observed that the predicted values accord well with the observed values (R = 0.98, RMSE = 6.47, and bias = 12.35). All the data points lie around the regression line, with very few (especially the lower values) lying beyond the 95% Confidence Interval (C.I.).

However, the presence of positive bias indicates that the model is slightly overestimating the response variable.

Further to understand the distribution of errors in the model, we have plotted the error histogram of the model using 10 bins (Figure 6b). We fitted a continuous Gaussian fit on the error distribution and found that the error follows left-skewed distribution (also called negatively skewed distribution). The error ranges from -7.4 (leftmost bin) to 21.4 (rightmost bin). Negative errors (left to the zero error line) represent the underestimated region, and positive errors (right to the zero error line) represent the overestimated region. The peak of the distribution lies in the overestimated region, indicating the presence of positive bias.

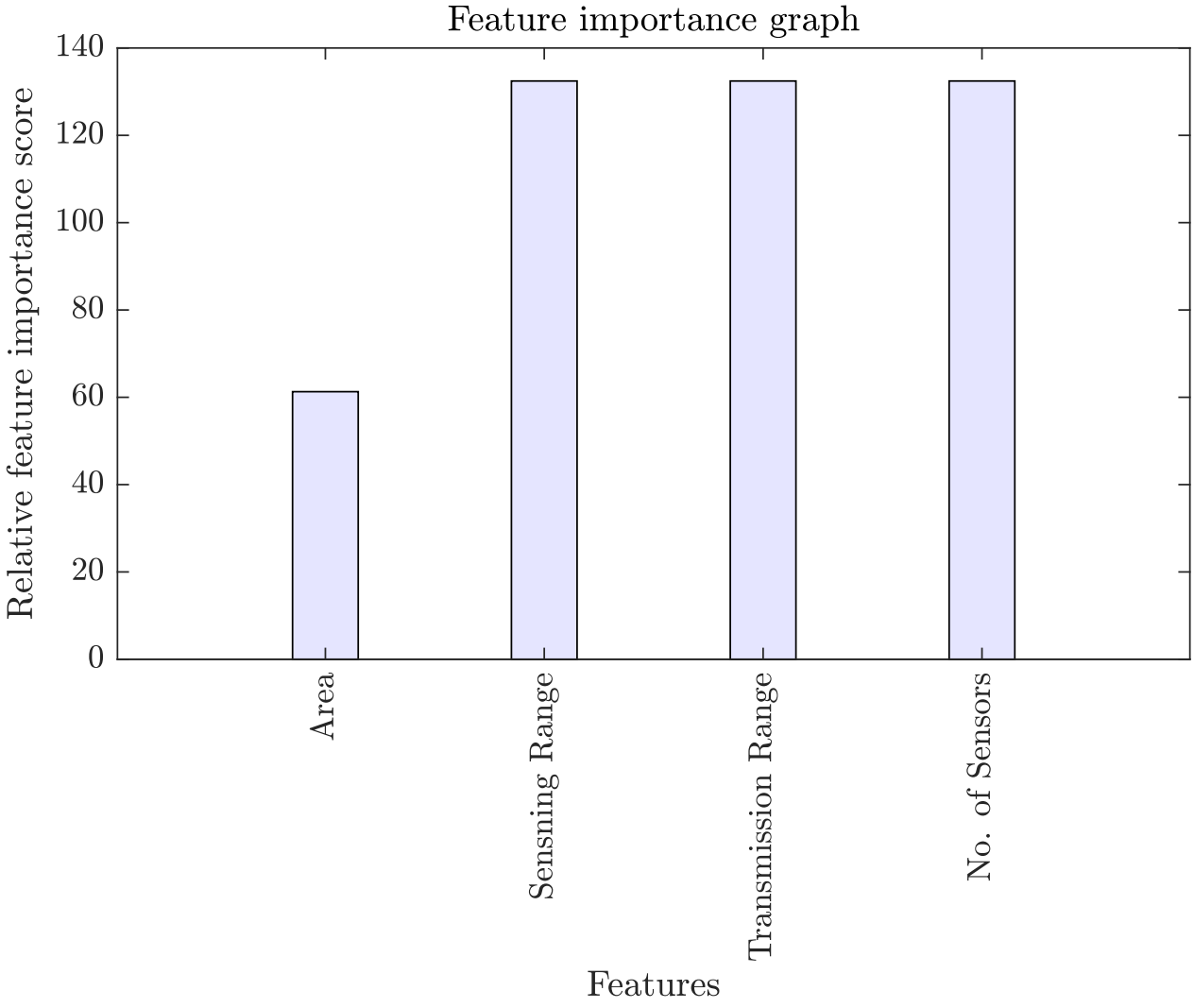

Figure 4. Bar graph illustrating each feature's relative feature importance score estimated through regression tree ensemble approach.

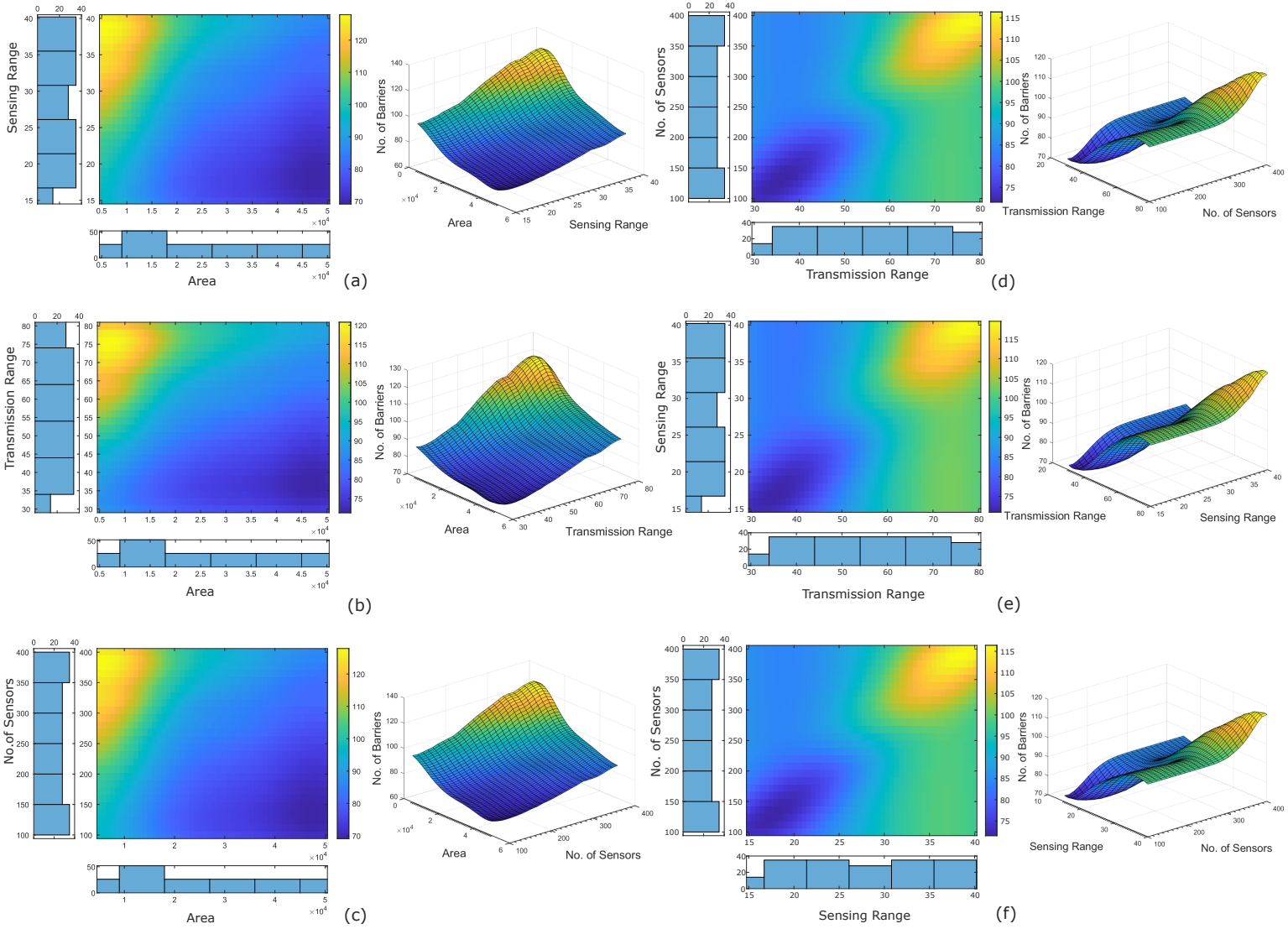

Figure 5. Feature sensitivity analysis through partial dependency plot. Two features are considered at a time (a total of six pairs from **a**–**f**). The left image shows the 2-D variation profile (with histogram) for each pair, and the right image shows the corresponding 3-D variation profile.

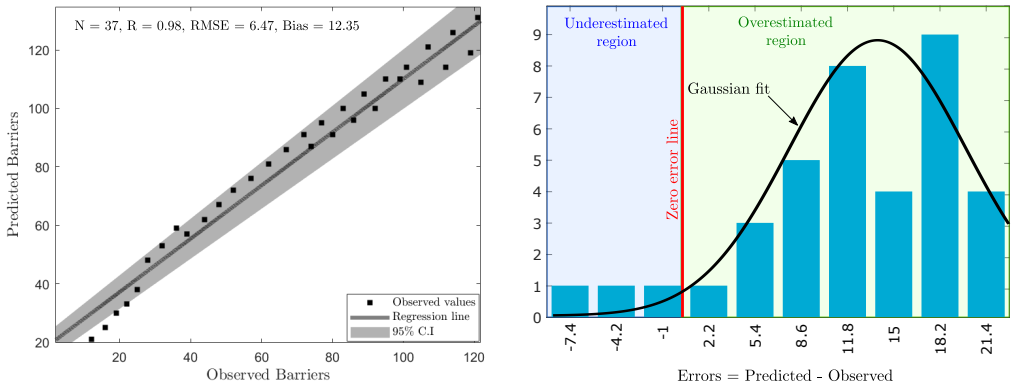

Figure 6. (**a**) Linear regression curve between the predicted result of LT-ZM-SVR model and observed values. (**b**) Error distribution analysis though error histogram.

4. Discussion

4.1. Comparison with Other Scaling Methods

We also evaluated and compared the performance of other scaling approaches. We considered Center Mean (CM) scaling and Min-Max scaling along with Z-score scaling. We also considered the Non-Scaled (NS) version for an appropriate comparison. After log transformation of the features, we applied these scaling techniques and trained the SVR model. We reported the performance of LT-NS-SVR, LT-CM-SVR, LT-ZM-SVR, and LT-MM-SVR in Table 2. Interestingly, we found that the predicted barriers accord well with the observed values for all the variants. However, the RMSE, MSE, bias, and computational time complexity of LT-NS-SVR is worst among all.

Table 2. Comparison of the performance of Z-score scaling (i.e., LT-ZM-SVR) with other scaling methods (i.e., LT-NS-SVR, LT-CM-SVR, and LT-MM-SVR).

Performance Metrics	LT-NS-SVR	LT-CM-SVR	LT-ZM-SVR	LT-MM-SVR
R	0.96	0.94	0.98	0.97
RMSE	12.66	2.39	6.47	4.59
MSE	160.15	5.727	41.87	21.10
Bias	36.30	6.24	12.35	15.62
Time (s)	2.21	0.59	0.65	0.51

4.2. Comparison with Benchmark Algorithms

To ensure an unbiased conclusion, we compared the performance of the proposed approach with different benchmark algorithms. In doing so, we evaluated the performance of ANN [39], GRNN [40], GPR [41,42], and Random Forest (RF) [43] over the same datasets after performing LT and z-score scaling on the features (Table 3). These models are selected based upon their performance in different applications such as remote sensing [30], WSNs [44], IoT [45], and blockchain [46]. We selected R, RMSE, MSE, bias, and computational time as the performance metrics. In comparing, we found that the proposed approach outperforms the benchmark algorithms in terms of RMSE, MSE, and bias. Additionally, LT-ZM-SVR emerges as the computationally efficient approach. Surprisingly, we found that the RF has the best R; however, with a poor RMSE. We observed a positive bias (i.e., overestimation tendency) in GRNN, GPR, RF, and LT-ZM-SVR. In contrast, a negative bias (i.e., underestimation tendency) is observed with ANN.

Table 3. Comparison of the proposed model with the benchmark algorithms.

Performance Metrics	Methods				
	LT-ZM-SVR	**ANN**	**GRNN**	**GPR**	**Random Forest**
R	0.98	0.38	0.96	0.94	0.99
RMSE	6.47	46.37	57.56	63.83	32.15
MSE	41.87	2150.20	3312.00	4074.7	1033.6
Bias	12.35	-36.12	49.62	50.96	28.62
Time (s)	0.65	1.81	2.02	1.71	2.70

4.3. Sensitivity Analysis of the LT-ZM-SVR

Finally, we performed the sensitivity analysis of the LT-ZM-SVR model to evaluate its robustness in the presence of uncertainty in input features. To do so, we introduced a fixed amount of variation in any one of the input features, keeping others constant. We performed this iteratively for all the features and reported the percentage change in the response variable in Figure 7. From the heat map, we found that overall the model is quite stable in the presence of small uncertainty. Relatively, the model is more vulnerable to the uncertainly present in the number of sensors.

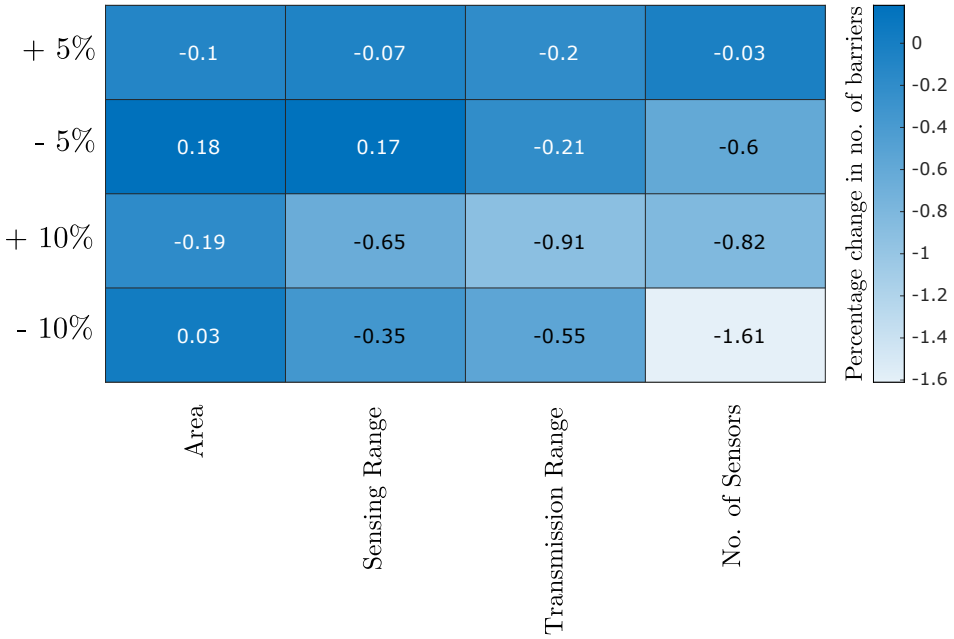

Figure 7. Sensitivity analysis of LT−ZM−SVR for ±5% and ±10% uncertainty in the input feature.

5. Conclusions

This study proposed a novel approach to estimate the number of barriers required for intrusion detection. To do so, we extracted relevant features from the network parameters through Monte Carlo simulations. We evaluated the relevancy of each feature through feature importance analysis. We found the area of the RoI to be the least relevant feature in estimating the number of barriers. All other features (*i.e.*, the sensing range, the transmission range, and the number of sensors) equally carry the highest relevancy. Additionally, to measure the impact of each feature on the response variable, we performed a feature sensitivity analysis. We observed that except for the area of the RoI, all other features positively impact the response variable. Afterward, we applied log transformation and scaling operations on the selected features. After feature pre-processing, we applied the tuned SVR algorithm as an interpretable data-driven model. Once our model was

trained, we evaluated its performance on the testing datasets using R, RMSE, MSE, bias, and computational time complexity as performance metrics. We found that the proposed approach accurately and timely predicts the number of barriers for fast intrusion detection and prevention.

For a robust conclusion, we compared the performance of the proposed approach with different scaling and benchmark algorithms. We found that the proposed methodology outperforms all the benchmark algorithms. However, the limitation of the proposed algorithm is that it assumes the values of the input features to be a positive real number. This study is a step towards fast intrusion detection and prevention using WSNs. Our approach can be employed for near-real-time applications such as border surveillance.

Author Contributions: A.S. developed the models, J.N. and J.A. extracted the datasets, S.S. and C.-C.L. analysed the results. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the Ministry of Science and Technology (MOST), Taiwan, R.O.C., under contract no.: MOST 110-2410-H-030-032.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: The computer algorithms originated during the current study can be made available from the corresponding author or first author on a reasonable request.

Data Availability Statement: The datasets generated during and/or analysed during the current study can be downloaded from https://www.kaggle.com/abhilashdata/intrusion-data-wsn (accessed on 26 January 2022).

Acknowledgments: We would like to acknowledge IISER Bhopal, Madhya Pradesh, India; Gautam Buddha University, Uttar Pradesh, India; IIT Kharagpur, West Bengal, India; MITS Gwalior, Madhya Pradesh, India; Fu Jen Catholic University, Taiwan; Asia University, Taiwan, for providing institutional support.

Conflicts of Interest: The author states that there is no conflict of interest. All the sources are cited, referred and acknowledged.

References

1. Mostafaei, H.; Chowdhury, M.U.; Obaidat, M.S. Border surveillance with WSN systems in a distributed manner. *IEEE Syst. J.* **2018**, *12*, 3703–3712. [CrossRef]
2. Lee, S.; Jain, S.; Yuan, Y.; Zhang, Y.; Yang, H.; Liu, J.; Son, Y.J. Design and development of a DDDAMS-based border surveillance system via UVs and hybrid simulations. *Expert Syst. Appl.* **2019**, *128*, 109–123. [CrossRef]
3. Sharma, M.K.; Singal, G.; Gupta, S.K.; Chandraneil, B.; Agarwal, S.; Garg, D.; Mukhopadhyay, D. INTERVENOR: Intelligent Border Surveillance using Sensors and Drones. In Proceedings of the 2021 6th International Conference for Convergence in Technology (I2CT), Pune, India, 2–4 April 2021; pp. 1–7.
4. Komar, C.; Donmez, M.Y.; Ersoy, C. Detection quality of border surveillance wireless sensor networks in the existence of trespassers' favorite paths. *Comput. Commun.* **2012**, *35*, 1185–1199. [CrossRef]
5. Nagar, J.; Chaturvedi, S.K.; Soh, S. An analytical framework with border effects to estimate the connectivity performance of finite multihop networks in shadowing environments. *Cluster Comput.* **2021**, *25*, 187–202. [CrossRef]
6. Singh, A.; Sharma, S.; Singh, J.; Kumar, R. Mathematical modelling for reducing the sensing of redundant information in WSNs based on biologically inspired techniques. *J. Intell. Fuzzy Syst.* **2019**, *37*, 6829–6839. [CrossRef]
7. Nagar, J.; Chaturvedi, S.K.; Soh, S. Wireless Multihop Network Coverage Incorporating Boundary and Shadowing Effects. *IETE Tech. Rev.* **2021**, 1–16. [CrossRef]
8. Singh, A.; Sharma, S.; Singh, J. Nature-inspired algorithms for wireless sensor networks: A comprehensive survey. *Comput. Sci. Rev.* **2021**, *39*, 100342. [CrossRef]
9. Kandris, D.; Nakas, C.; Vomvas, D.; Koulouras, G. Applications of wireless sensor networks: An up-to-date survey. *Appl. Syst. Innov.* **2020**, *3*, 14. [CrossRef]
10. Kotiyal, V.; Singh, A.; Sharma, S.; Nagar, J.; Lee, C.C. ECS-NL: An Enhanced Cuckoo Search Algorithm for Node Localisation in Wireless Sensor Networks. *Sensors* **2021**, *21*, 3576. [CrossRef] [PubMed]
11. Amutha, J.; Sharma, S.; Sharma, S.K. Strategies based on various aspects of clustering in wireless sensor networks using classical, optimization and machine learning techniques: Review, taxonomy, research findings, challenges and future directions. *Comput. Sci. Rev.* **2021**, *40*, 100376. [CrossRef]

12. Wang, Y.; Fu, W.; Agrawal, D.P. Gaussian versus uniform distribution for intrusion detection in wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *24*, 342–355. [CrossRef]
13. Sharma, A.; Chauhan, S. Sensor fusion for distributed detection of mobile intruders in surveillance wireless sensor networks. *IEEE Sens. J.* **2020**, *20*, 15224–15231. [CrossRef]
14. Nurellari, E.; Licea, D.B.; Ghogho, M.; Rivero-Angeles, M.E. On Trajectory Design for Intruder Detection in Wireless Mobile Sensor Networks. *IEEE Trans. Signal Inf. Process. Netw.* **2021**, *7*, 236–248. [CrossRef]
15. Amutha, J.; Nagar, J.; Sharma, S. A distributed border surveillance (dbs) system for rectangular and circular region of interest with wireless sensor networks in shadowed environments. *Wirel. Pers. Commun.* **2021**, *117*, 2135–2155. [CrossRef]
16. Singh, R.; Singh, S. Smart border surveillance system using wireless sensor networks. *Int. J. Syst. Assur. Eng. Manag.* **2021**, 1–15. [CrossRef]
17. Vadivelan, N.; Taware, M.S.; Chakravarthi, M.R.R.; Palagan, C.A.; Gupta, S. A border surveillance system to sense terrorist outbreaks. *Comput. Electr. Eng.* **2021**, *94*, 107355. [CrossRef]
18. Sharma, S.; Nagar, J. Intrusion detection in mobile sensor networks: A case study for different intrusion paths. *Wirel. Pers. Commun.* **2020**, *115*, 2569–2589. [CrossRef]
19. Karthy, G.; Harish, M.; Harish, R.; Srivarshan, R.N.; Sridhar, B. BORS (Border Patrol Search) ROBOT by using Wireless Technology. In Proceedings of the 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 4–6 August 2021; pp. 449–456.
20. Roman, R.C.; Precup, R.E.; Petriu, E.M. Hybrid data-driven fuzzy active disturbance rejection control for tower crane systems. *Eur. J. Control* **2021**, *58*, 373–387. [CrossRef]
21. Zhu, Z.; Pan, Y.; Zhou, Q.; Lu, C. Event-triggered adaptive fuzzy control for stochastic nonlinear systems with unmeasured states and unknown backlash-like hysteresis. *IEEE Trans. Fuzzy Syst.* **2020**, *29*, 1273–1283. [CrossRef]
22. Singh, A.; Nagar, J.; Sharma, S.; Kotiyal, V. A Gaussian process regression approach to predict the k-barrier coverage probability for intrusion detection in wireless sensor networks. *Expert Syst. Appl.* **2021**, *172*, 114603. [CrossRef]
23. Schmidt, J.; Marques, M.R.; Botti, S.; Marques, M.A. Recent advances and applications of machine learning in solid-state materials science. *Npj Comput. Mater.* **2019**, *5*, 1–36. [CrossRef]
24. Nikolenko, S.I. Synthetic data for deep learning. *arXiv* **2019**, arXiv:1909.11512.
25. Chen, R.J.; Lu, M.Y.; Chen, T.Y.; Williamson, D.F.; Mahmood, F. Synthetic data in machine learning for medicine and healthcare. *Nat. Biomed. Eng.* **2021**, 1–5. [CrossRef] [PubMed]
26. Rankin, D.; Black, M.; Bond, R.; Wallace, J.; Mulvenna, M.; Epelde, G. Reliability of supervised machine learning using synthetic data in health care: Model to preserve privacy for data sharing. *JMIR Med. Inform.* **2020**, *8*, e18910. [CrossRef] [PubMed]
27. Singh, A.; Kotiyal, V.; Sharma, S.; Nagar, J.; Lee, C.C. A machine learning approach to predict the average localization error with applications to wireless sensor networks. *IEEE Access* **2020**, *8*, 208253–208263. [CrossRef]
28. Abay, N.C.; Zhou, Y.; Kantarcioglu, M.; Thuraisingham, B.; Sweeney, L. Privacy preserving synthetic data release using deep learning. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Dublin, Ireland, 10–14 September 2018; pp. 510–526.
29. Zou, Y.; Chakrabarty, K. Sensor deployment and target localization in distributed sensor networks. *ACM Trans. Embed. Comput. Syst. (TECS)* **2004**, *3*, 61–91. [CrossRef]
30. Singh, A.; Gaurav, K.; Rai, A.K.; Beg, Z. Machine learning to estimate surface roughness from satellite images. *Remote Sens.* **2021**, *13*, 3794. [CrossRef]
31. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
32. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
33. Garg, R.; Kumar, A.; Bansal, N.; Prateek, M.; Kumar, S. Semantic segmentation of PolSAR image data using advanced deep learning model. *Sci. Rep.* **2021**, *11*, 1–18. [CrossRef] [PubMed]
34. Hong, W.C. Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm. *Energy* **2011**, *36*, 5568–5578. [CrossRef]
35. Heinonen, M.; Mannerström, H.; Rousu, J.; Kaski, S.; Lähdesmäki, H. Non-stationary gaussian process regression with hamiltonian monte carlo. In *Artificial Intelligence and Statistics*; PMLR: New York, NY, USA, 2016; pp. 732–740.
36. Syarif, I.; Prugel-Bennett, A.; Wills, G. SVM parameter optimization using grid search and genetic algorithm to improve classification performance. *Telkomnika* **2016**, *14*, 1502. [CrossRef]
37. Reed, R.; MarksII, R.J. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*; MIT Press: Cambridge, MA, USA, 1999.
38. Zhan, T.; Gong, M.; Jiang, X.; Li, S. Log-based transformation feature learning for change detection in heterogeneous images. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1352–1356. [CrossRef]
39. Benardos, P.; Vosniakos, G.C. Optimizing feedforward artificial neural network architecture. *Eng. Appl. Artif. Intell.* **2007**, *20*, 365–382. [CrossRef]
40. Specht, D.F. A general regression neural network. *IEEE Trans. Neural Netw.* **1991**, *2*, 568–576. [CrossRef] [PubMed]
41. Rasmussen, C.E. Gaussian Processes in Machine Learning. In *Advanced Lectures on Machine Learning*; Summer School on Machine Learning; Springer: Berlin/Heidelberg, Germany, 2003; pp. 63–71.

42. Quinonero-Candela, J.; Rasmussen, C.E. A unifying view of sparse approximate Gaussian process regression. *J. Mach. Learn. Res.* **2005**, *6*, 1939–1959.
43. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
44. Zhang, D.; Zhang, X.; Xie, F. Research on Location Algorithm Based on Beacon Filtering Combining DV-Hop and Multidimensional Support Vector Regression. *Sensors* **2021**, *21*, 5335. [CrossRef]
45. Gupta, N.; Khosravy, M.; Patel, N.; Dey, N.; Crespo, R.G. Lightweight Computational Intelligence for IoT Health Monitoring of Off-Road Vehicles: Enhanced Selection Log-scaled Mutation GA Structured ANN. *IEEE Trans. Ind. Inf.* **2021**, *18*, 611–619. [CrossRef]
46. Dibaei, M.; Zheng, X.; Xia, Y.; Xu, X.; Jolfaei, A.; Bashir, A.K.; Tariq, U.; Yu, D.; Vasilakos, A.V. Investigating the prospect of leveraging blockchain and machine learning to secure vehicular networks: A survey. *IEEE Trans. Intell. Transp. Syst.* **2021**. [CrossRef]