Contents lists available at ScienceDirect

Computer Communications

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

Anomaly detection in wireless sensor network using machine learning algorithm

I. Gethzi Ahila Poornima, Research Scholar [*], B. Paramasivan, Professor

Department of Information Technology, National Engineering College, India

ARTICLE INFO

Keywords:
Wireless sensor networks
Anomaly detection
Machine learning
Linear weighted projection regression
LWPR

ABSTRACT

Security in the Wireless Sensor Network(WSNs) is an essential and a challenging task. Anomaly detection is a key challenge to ensure the security in WSN. WSNs are vulnerable to various threats which may cause the node to get damaged and produce faulty measurements. The detection of such anomalous data is required to reduce false alarms. Machine learning algorithm based detection of anomalous data becomes popular now. Most of the current machine anomaly detection algorithms run in a stationary environment and require the entire training data to be kept in the node. In this paper, we formulate an Online Locally Weighted Projection Regression (OLWPR) for anomaly detection in Wireless Sensor Network. Linear Weighted Projection Regression methods are non parametric and the current predictions are performed by local functions that use only the subset of data. So, the computation complexity is low which is one of the requirements in Wireless Sensor Network. The dimensionality reduction in LWPR is done online by Principal Component Analysis (PCA) to handle the irrelevant and redundant data in the input data. After the prediction process, the dynamic threshold value is determined by a dynamic thresholding method to find the deviations of predicted value from the actual sensed value. OLWPR attains the detection rate of 86 percentage and very low error rate of only 16%.

1. Introduction

Wireless Sensor Network. Wireless Sensor Network (WSN) is a set of autonomous nodes grouped through the wireless channel and deployed in unmonitored or hazard-prone areas like dark forestation, desert, underwater or volcanoes [1,2]. WSN exploits a huge number of sensor nodes to collect aspects such as temperature, sound, pressure, humidity, light under different environments. Some of the applications of WSN include forest fire detection, environmental monitoring, mechanical stress recognition after an earthquake, biodiversity mapping for observing wildlife, or monitoring the patients in the intensive care unit. Because the sensor nodes are mostly deployed in unattended areas, monitoring every sensor node is unworkable. Hence, either these sensors can either fail at any given point of time or an intruder can attack the node thereby deteriorating the network and causing issues in collecting the data from the sensors. In addition to the failures of these sensors, we have constraints related to limited energy, memory, bandwidth, and communication.

Security techniques in WSN. WSN is mostly vulnerable to many kinds of attacks. Security is the major challenge in WSN due to the factors such as wireless medium, low transmission range, Ad hoc deployment, hostile environment, limited energy. To secure these sensors we have two different techniques available in WSN, which is prevention-based and detection based.

Prevention based techniques. The first line of defense against the security attacks in WSN is the prevention based techniques. The prevention-based technique mainly includes cryptography, which required more computation time and resources. Therefore, this is not a preferred technique for WSN.

Detection based detection. On the other hand, the detection-based techniques would be more suitable as this uses misuse/Signature or Anomaly detection, which consumes less time and resources.

Misuse/signature detection. It defines a set of abnormal behavior of the network in prior. Then it looks up for attacks for which the technique has defined already. As a result, the signature-based detection could not detect new attacks as the technique knows only the behavior of attacks that it has defined earlier.

Anomaly detection. On the contrary, the anomaly detection technique learns the behavior of the normal environment and creates a model for normal events in the network. The anomalies are the data/events that deviate from the normal data/events. The anomaly detection reveals the anomalies based on the predefined set of normal data/events. Hence this kind of variance detection could detect even unknown attacks. Even though the detection rate of anomaly detection is considerably high, it suffers high false alarm rates.

[*] Corresponding author.
E-mail addresses: gethziphd@gmail.com (I.G.A. Poornima), bparamasivan@yahoo.co.in (B. Paramasivan).

https://doi.org/10.1016/j.comcom.2020.01.005
Received 13 August 2019; Received in revised form 19 December 2019; Accepted 2 January 2020
Available online 10 January 2020

The anomaly detection techniques [2] includes the approaches as such statistical-based, clustering-based, Classification based, and Artificial intelligence. Support vector machine (SVM) [3,4], One class Principal Component Classifier [5], Naive Bayes [6], Bayesian Network [7], and self-organizing map [8] based on neural networks are the Machine learning techniques used so far to reduce the false alarm rate to an extent. However, in all the aforementioned methods, adding new training data causes a drastic change in the outcome and the entire system gets collapsed. In addition to the above issues, the machine learning takes place on the static datasets in offline mode and as a result of the static learning, the prediction or classification befalls based on the predefined behaviors alone. Thus, setting up dynamic rules is not feasible in these techniques, and also these techniques required the presence of the entire datasets in the node where the prediction occurs which requires additional memory for storing the data at the node. As time goes, the data sensed by the sensors keep on increasing which leads to high memory consumption, and keeping huge training data in the sensor node is impractical.

Our objective is to find an online prediction based anomaly detection with reduced false alarm rates and very limited memory consumption. Even though some online techniques [9,8,10,11] are available for the online detection process, all the aforementioned state of art utilizes memory for both executing the detection algorithm and storing the entire dataset. Nevertheless, our work consumes memory only for executing the algorithm as the memory consumed for storing the model rather than the entire dataset is negligible. We propose an online Linear Weighted Projection Regression (OLWPR) based detection with a dynamic threshold method. We carry out the online detection of anomalous data in the sensor data captured in real-time in three phases. The first phase includes the data compression and the second includes prediction using LWPR. Finally, the detection phase uses the dynamic threshold method to detect anomalous data. The major contribution of the proposed scheme is twofold,

- Development of OLWPR model
- Detection of anomalous data by using dynamic threshold value.

The rest of the paper is organized as follows. Section 2 begins with an introduction to Linear Weighted Projection Regression technique and reviews some of the existing machine learning algorithm applied for detection of anomalous data/node in WSN. In Section 3, we present the proposed online LWPR along with the threshold calculation and detection process. Section 4 gives the experimental results of the proposed algorithm and comparison with the other existing models in this dataset. Section 5 concludes our work.

2. Related works

Secure data transmission within the limited resource in WSN is a great challenge for various applications. WSN possesses ad hoc behavior which means the network allows any nodes to enter and leave dynamically. As a result, the uncertainty of nodes exists in WSN causing the intruder to harass the nodes in the network. In addition to this, the sensors may fail due to environmental disasters or the depletion of battery power. Mohiuddin Ahmed et al. [4] surveyed the available approaches for the detection of network anomalies. However, they have not focused constraints of the WSN environments as such limited memory, computation power, energy, and bandwidth.

Nauman Shahid et al. [5], analyzed the characteristics such as energy conservation, minimum communication overhead, robustness against the dynamic nature of the network structure that should be satisfied by the algorithms to work under WSN. Pu Cheng et al. [10] proposed lightweight algorithms using one class Quarter Sphere Support vector machine (QSSVM) for a sort problem. They achieved the accuracy as same as QSSVM but the computational complexity is highly reduced. However, they have not focused on memory consumption.

The authors in [12] used K means clustering to detect the hybrid attack. Using K means, the anomalous patterns are constructed over the training data. The assumption made here is the training data is free from anomalies. The testing data contains both normal and anomalous data. The testing data is then compared with the anomalous pattern generated over the training data. The detection is done in a centralized manner. Anomalous nodes are detected exploiting the network traffic recorded offline which is the training data. The detection has been carried out in an online manner. The major disadvantage of this technique is the anomalous pattern generated over the training data is static, but it may change over time. Hence, it may cause false alarm rates and affect accuracy. On the other hand, S. Siripanadorn et al. [11] presented an anomaly detection algorithm for WSN scenario using Discrete Wavelet Transform (DWT) combined with Self Organizing Map (SOM). They exploit only the important feature of data instead of using the entire data thereby emphasizing more on memory consumption rather than computational power. However, some information loss may occur owing to the conversion of data to signal to reduce memory consumption.

In [6], Hyper grid KNN based Anomaly detection in the WSN environment has been proposed to detect cyber-attack and random faults. The detection region of conventional KNN is redefined as Hypercube. The parameters are estimated dynamically and adaptively. The major advantage of this method over other KNN is that it is not required to adjust the parameters manually. Also, it is scalable without a huge change in complexity. But the assumption made is that the training data do not contain any abnormal data. It is not always possible to collect pure data from any WSN environment.

Support Vector Machine is one of the techniques that are often used for the Anomaly detection system. The authors in [13] implemented a hyper-ellipsoidal one-class support vector machine to model the normal data. Then the outliers are detected in the distributed environment online. They conclude that the system generates a very low false alarm rate due to the updating of a model that was created at the time of the learning period. in [14,15] uses – class quarter – sphere support vector machine has been used to model the training data.

The authors in [7,14] proposed SVM and PCA based techniques for the detection of an anomaly. The first algorithm in this work is SVM with a slight modification in the RBF kernel to adapt for non-stationary time series data. The second one is the PCA based technique. To decrease the computational complexity due to the Eigen Decomposition (ED), the Orthonormal Projection Approximation subspace tracking approach is applied. Support Vector Data Description (SVDD) is modified in [15] so that the training complexity is reduced by reducing the number of sample points near the center of the sphere since they do not contribute much to the determination of hyper sphere. Thus the training time and memory are highly reduced but the problem is that the training data has to be kept in the node.

In all the existing methodology, the false alarm rate is not reduced enough. Also, the training data has to be kept in the respective nodes for further detection process. It leads to high memory consumption. As a result, the time required to detect by analyzing training data is high. The proposed scheme overcomes the aforementioned problems by keeping the model rather than the entire data in the nodes. Thus the detection time and memory consumption is very low.

3. Proposed online LWPR prediction based anomaly detection (LWPRAD)

In our work, cluster node WSN architecture is considered. The detection of anomalous data is carried out in three phases

1. Phase I: Data Compression.
2. Phase II: Linear Weighted Projection Regression (LWPR) based prediction.
3. Phase III: Detection of anomalous data using the dynamic thresholding method.

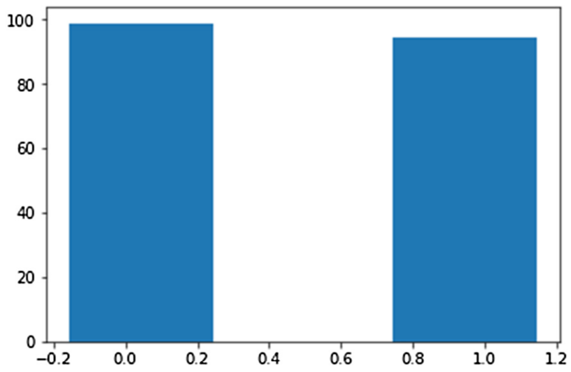

Fig. 1. Variance of PCA.

3.1. Data compression

Principal Component Analysis is used for dimensionality reduction. The need for Dimensionality Reduction is that the space required to store the data increases when the number of dimensions increases. Since the available memory is very limited in the WSN environment, the dimensionality reduction plays an important role here. Hence, lesser dimensions lend themselves to less computation or training time.

PCA is the technique that extracts a low dimensional set of features from a high dimensional data without deviating from the main objective which is to acquire as much information as possible. PCA solicits a linear combination of features such that maximum variance is extracted from the samples. Then it eliminates this variance and inquires a second linear combination that explains the maximum proportion variance of the remaining samples and so on. Before applying PCA, the data samples are normalized based on mean and variance.

Consider the data matrix Str_{mXn} with' observations with 'n' variables collected from the environment. Str_{mXn} is then normalized to zero mean and unit variance. The normalized matrix is $\overline{Str}_{\mathrm{mXn}}$. The covariance matrix CV is determined from the normalized matrix

$\overline{Str_{\mathrm{mXn}}}$ and the Single value decomposition algorithm is then carried out on $\overline{Str}_{\mathrm{mXn}}$ matrix. Then the matrix $\overline{Str}_{\mathrm{mXn}}$ is projected into two spaces namely Principal Component Space and Residual space. The former holds the components that retain most of the data variance but, the latter Residual Space holds the other components with lesser variance. This projection can be expressed as follows,

$$\overline{Str}_{\mathrm{mXn}} = \widehat{Str} + \widetilde{Str} \tag{1}$$

where $\widehat{Str}$ represents the Principal Component space and $\widetilde{Str}$ represents the residual space described in Eqs. (2) and (3) respectively.

$$\widehat{Str} = \hat{T}\hat{P}^T = \overline{Str}_{mXn}\hat{P}\hat{P}^T = \widehat{Str}_{mXn}\widehat{CV} \tag{2}$$

$$\widetilde{Str} = \tilde{T}\tilde{P}^T = \overline{Str}_{mXn}\tilde{P}\tilde{P}^T = \widetilde{Str}_{mXn}(1-\widetilde{CV}) \tag{3}$$

The loading matrix P is $[\hat{P}\tilde{P}]$, where both $\hat{P}$ and $\tilde{P}$ contains the set of l and $n-l$ Eigenvectors of the covariance matrix CV. $\widehat{CV}$ and $\widetilde{CV}$ are the results of projection on the Principal Component Analysis and Residual space respectively. The goal of PCA is to find the weighted matrix W such that Eq. (4) gets minimized.

$$F\left(\overline{Str}[t].W\right) = \frac{1}{N}\sum_{t=1}^{N}\left\|\overline{Str}[t] - WW^T\overline{Str}[t]\right\|^2 \tag{4}$$

where N is the number of observations and $1 \le t \le N$. While solving, we get a set of first eigenvectors of the correlation matrix $StrStr^T$ in the decreasing order of the degree of variance got by eigenvector. The variance of the Principal components selected is as shown below. (See Fig. 1.)

3.2. Linear weighted projection regression-based prediction

LWPR is one of the categories of LWL. The LWL comes with memory-based locally weighted learning and purely incremental learning.

3.2.1. Locally weighted learning

Memory-based Locally Weighted Learning (LWL) is a regression technique where prediction is carried out by creating a local model around the point of interest. Support vector machine, Gaussian and linear regression techniques use a global model to predict the query points. Also, these methods use the entire training dataset for creating the global model based on the neighboring data of the query points. But in LWL, it creates a local model and uses only a subset of data for prediction. Each data point in the training set becomes a weighting factor that demonstrates the level of influence of the data points for the process of prediction. A weighing factor of a data point describes how far the particular data point is relevant for the ongoing prediction. Since all the training data has to be kept in memory, it is not applicable for WSN. So, we focused more on the purely incremental method. The algorithm is explained below.

Algorithm: Locally Weighted learning based on memory

Input:

- *Datapoint* x_{pred}
- *Training points*(x_{ij}, y_i), *0≤ i ≤n & 0≤j≤* n_a *where n is the number of training points and*n_a *is the number of attributes or features selected.*

Prediction:

- *Construct matrix X* $\begin{bmatrix} x_{11} & x_{21} & x_{31} & \cdots & \cdots & x_{n1} \\ x_{12} & x_{22} & x_{32} & \cdots & \cdots & x_{n2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & . & . & . & . \\ x_{1n_a} & x_{2n_a} & x_{3n_a} & \cdot\cdot & . & x_{nn_a} \end{bmatrix}$
- *Construct vector* $y=(y_1, y_2, y_3, \ldots\ldots, y_n)^T$
- *Determine Weight matrix (diagonal) W where each diagonal element is*

$$w_{i,i} = exp\left(-\frac{1}{2}\left(x_i - x_{pred}\right)^T D\left(x_i - x_{pred}\right)\right)$$

- *The predicted value is calculated as follows,*

$$y_{pred} = [x_{pred}^T\ 1]\beta_{pred}$$

where $\beta_{pred} = (X^T W X)^{-1} X^T W y$

3.2.2. LWPR

LWPR is one of the purely incremental learning methods. It considers the local model rather global model as in linear regression. It follows non parametric learning method and so it is very fast with second order learning methods. For the applications where linear model is not sufficient to model the data, LWPR could be used. This is because LWPR algorithm could create non linear model to model the data. The prediction process is carried out here by the weighted combination of local models that are linear. to model the data, LWPR could be used. This is because LWPR algorithm could create non linear model to model the data. The prediction process LWPR learning method is explained in the next section in detail. The proposed online distributed locally weighted projection regression is executed in each node of the WSN. A finding of the locality of the linear model from the data plays an important role in the prediction process. This method does not use the entire dataset and also all the training data is not required to be kept in the memory. Only the local model is kept in the memory and updated whenever a test data enter.

LWPR learning mechanism. The kernel is created and weights $w_{\mathrm{ke,i}}$ are calculated for each data point (x_{i}, yi) which resides inside the kernel based on the distance from the center c_k of the kernel in each local model. The Gaussian kernel is determined as follows,

$$w_{k,1} = \exp(-\frac{1}{2}\left(x_i - c_k\right)^T D_k(x_i - c_k)) \tag{5}$$

D_k represents distance metrics that implies the size of the region of validity of the model. The region of validity is called Receptive field (RF). The distance metric D is changed and optimized for every receptive field. This optimization can be carried out using a stochastic leave-one-out cross-validation criterion. The prediction output of the LWPR method is as follows, (by combining N_k locally weighted models).

$$y_{pred} = \frac{\sum_{k=1}^{N_k} w_k y_k}{\sum_{k=1}^{N_k} w_k} \tag{6}$$

where $y_k = \beta_k s$ and s is the projected input. The weight w_k of the local model k is calculated using (5). w_k is the Receptive Field (RF) of the local model k is calculated using (5). w_k is the Receptive Field (RF) of the local model k. The parameters to be learned are projection directions, regression parameter β (local) and distance metric D. D can be built as

$$D = hdiag([n_1, n_2, \ldots .n_n]) \tag{7}$$

with the scaling parameter 'h'. The learned parameters are updated for the subsequent updating of local models.

LWPR local model structure. When the receptive field achieves the maximum allowable input to be trained, the distance metric D is updated using a stochastic leave-one-out cross-validation method so that the prediction error/cost is minimized. The LWPR model is created by tuning the input parameters of the sensed data. The number of input and output dimensions and the initial distance matrix are initialized. Leave one out cross-validation cost function is,

$$J = \frac{1}{\sum_{i=1}^{M} w_i} \sum_{i=1}^{M} w_i (y - \hat{y}_{i,-i})^2 + \frac{\gamma}{N} \sum_{i.j=1}^{N} D_{ij}^2 \tag{8}$$

Where M — number of data points in the input training dataset

$\left\{x_i, y_i\right\}_{i=1}^{M}$ — Training Data

$\sum_{i=1}^{M} w_i (y - \hat{y}_{i,-i})^2$ — Mean Leave-one-out-cross-validation error of the local model.

$\frac{\gamma}{N} \sum_{i.j=1}^{N} D_{ij}^2$ — Penalty term

The penalty term shows that receptive fields cannot shrink more when a large amount of training data is used. The algorithm for LWPR model updating during training is presented below.

```
Initialize LWPR with no model
For each training sample (x, y) do
for each local model k do
Activation weight $w_k = \exp(\frac{-1}{2}(x - c_k)^T D_k((x - c_k)))$
(i)    Update the mean and model parameters
$x_0^{M+1} = \frac{\lambda W^M x_0^M + wx}{W^{M+1}}$  $\beta_0^{M+1} = \frac{\lambda W^M \beta_0^M + wy}{W^{M+1}}$
where $W^{k+1} = \lambda W^k + w$ and $x_0^0 = 0, u_k^0 = 0, \beta_0^0 = 0, W^0 = 0$
(ii)   Update Model parameters
  $z_0 = x - x_0^{M+1}$ $res_o = y - \beta_0^{M+1}$
Update projection
$u_k^{M+1} = \lambda u_k^M + wz_{k-1} res_{k-1}$ -Determine Projection

  $s_k = s_{k-1}^T u_k^{M+1}$ – Project input data
$SS_k^{M+1} = \lambda SS_k^M + ws_k^2$
$SR_k^{M+1} = \lambda SR_k^M + ws_k^2 res_{k-1}$

$SZ_k^{M+1} = \lambda SZ_k^M + wz_{k-1}^2 s_k$
$\beta_k^{M+1} = SR_k^{M+1} / SS_r^{n+1}$
$p_k^{M+1} = SZ_k^{M+1} / SS_r^{n+1}$ Regress input data against current projection
$z_k = z_{k-1} - s_k p_k^{M+1}$ Reduce input space
$res_k = res_{k-1} - s_k \beta_k^{M+1}$
$MSE_k^{M+1} = \lambda MSE_k^M + wres_i^2$ Mean Square Error
Check whether the number of projection R has to be increased or not.
End for
End for
```

The algorithm to perform the LWPR based prediction during the testing period.

```
 Input: the query point array Xq – Test Data
 Initialize: $x_q \epsilon X_q = x_q - x_0$
 Prediction:

 For r=1 to R
   Compute latent variable $s_r = u_r^T x_q$
   update prediction $\hat{y}_q = \hat{y}_q + s_r \beta_r$
   reduce input space $x_q = x_q - s_r p_r^n$
 End for
                          $\hat{Y}_q = \hat{Y}_q . Add(\hat{y}_q)$
End for
```

Thus, the prediction is carried out and got the Yq array with the prediction values.

3.3. Setting threshold to differentiate the normal and abnormal sensed data

The output of the LWPR predictor with the predicted values is fed to the threshold method to detect the anomalous data. Here instead of giving a threshold value as input to the detection algorithm, we presented a dynamic threshold generation method in the detection procedure itself. The parameters under consideration will not be a constant for all the times because it may depend on some physical factor such as time and seasonal changes. The temperature may increase in the noontime but a little bit low in the morning and evening time. Hence the threshold should be dynamic to reduce the false rates. The sliding window concept is used to determine the dynamic threshold value. The number of samples taken for the sliding window is 40. So, the threshold value is calculated from the recent past 40 samples. The pseudo-code for the threshold determination algorithm is as given below

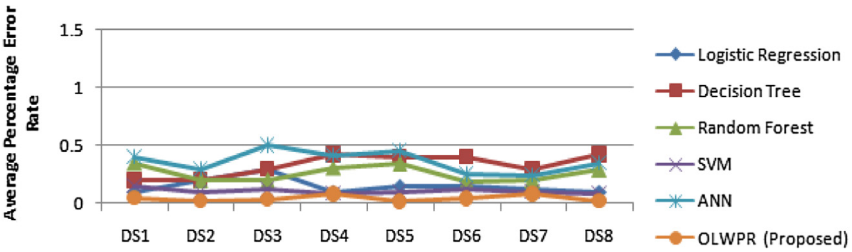

Fig. 2. Average percentage error.

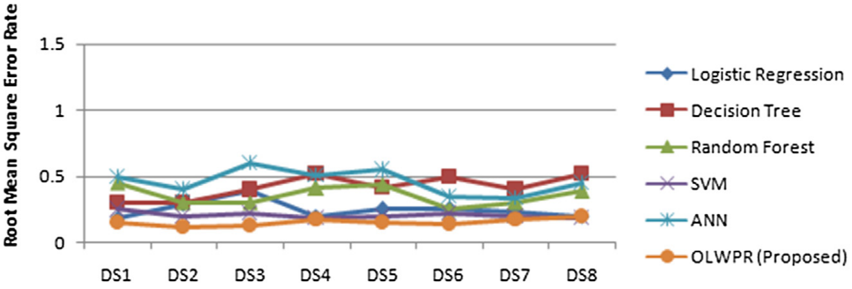

Fig. 3. Average RMSE.

```
Procedure: Finding Dynamic threshold
  Input: $D_a$ ,n
  Output: $S_t$,$T_t$- Standard Deviation, Threshold.
$\mu$=0;
$S_t$ = 0;
$T_t$=0;
For i=1:1: n
   Update $\mu$
   Update $S_t$
   Update $T_t$

End for
 $T_t = \mu$ -1 * $S_t$
```

4. Results and discussion

4.1. Datasets used

The dataset used in this work is collected by the researchers in Intel Berkeley Research Lab (IBRL) from 54 mica sensors. The data consists of the date, time, epoch id, mote id, temperature, humidity, light and voltage measurements measured every 31 s. Before starting the procedure, the missing records are handled first. After the removal of duplicate and null values, the attribute selected are time, mote id, temperature, humidity, and light. The data sensed by the mote 1, 2, 3, 4 and 5 are considered for this experiment to evaluate our proposed predictive model to detect the anomalies. Some abnormal values are injected into the dataset at a random position.

For node 1, we chose 40,000 real-time sensed records. The first 10000 records are used to build the LWPR predictive model. We built this predictive model offline. These records are training records. The rest of the data (30,000) are the testing data. These 30,000 data are split into 8 sets of data as below,

10001-13750-DS1
13751-17500-DS2
17501-21250-DS3
21251-25000-DS4
25501-28750-DS5
28751-32500-DS6
32501-36250-DS7
36251-40000-DS8

The testing is done online at each node. Prediction and comparison with the actual sensed value using the thresholding method are carried out. Similarly, we experimented with our proposed prediction-based anomaly detection in the nodes 1, 2, 3 and 4. The major advantage of our prediction is that it does not require to store all the training data in the node memory. Only the model is kept in the node and the prediction is carried out.

4.2. Performance analysis

To analyze the effectiveness of the proposed approach, some kind of anomalies are injected into the dataset. The LWPR prediction method is compared with the prediction technique Linear Regression (LR), Gaussian, and Sequential Minimal Optimization (SMO) Regression. The metrics used to compare the LWPR with the other techniques are Average Root Mean Square Error (RMSE) and Average Percentage Error. Also the parameters Accuracy, Accuracy error, F1-score, Sensitivity, Specificity, recall are determined for the proposed model and compared with the existing classification techniques like Logistic Regression, Decision Tree, Random forest, Adaboost, SVM, and ANN. The Table 1 shows the aforementioned parameters for both the conventional classification techniques and the proposed detection model. These results are obtained by taking six parameters such as Epoch Id, Datetime, temperature, humidity, light, and voltage.

Figs. 2 and 3 show the average percentage error and Root Mean Square Error of our proposed model.

The Detection rate is calculated using the formula as below,

$$Detection\ Rate = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{9}$$

True positive is the number of positive samples predicted as positive and the false negative is the number of positive samples predicted as negative. The false-positive rate is defined as follows,

$$FalsePositiveRate = \frac{FalsePositive}{FalsePositive + TrueNegative} \tag{10}$$

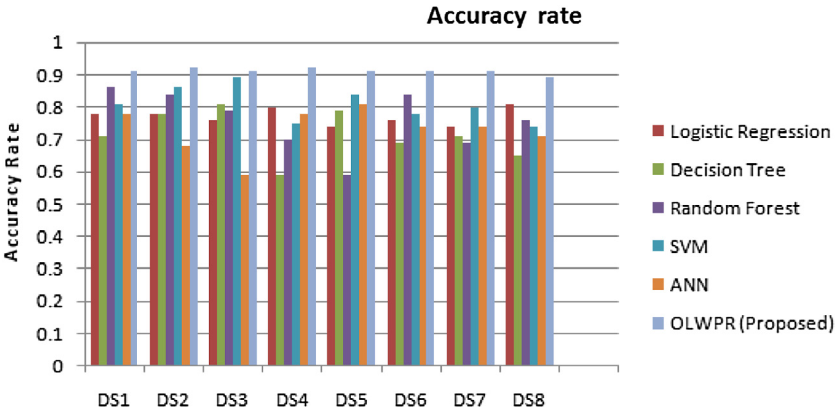

Fig. 4. Average Accuracy rate.

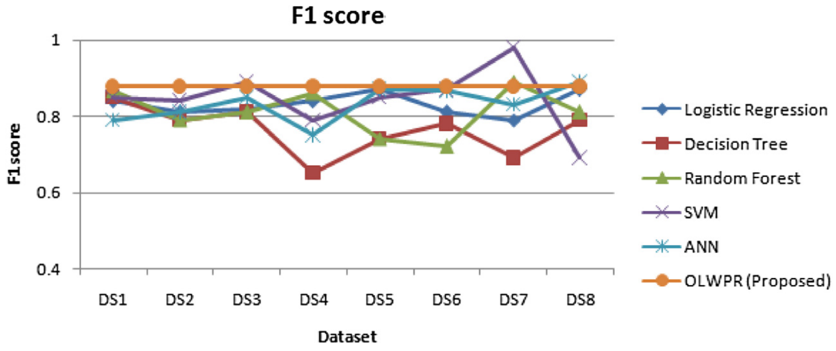

Fig. 5. F1 score.

Table 1
Comparison with algorithms like LR, DT, RF, SVM, and ANN.

S. No	Technique	Accuracy	Precision	Detection rate/Recall	F1 Score	AUC
1	Linear Regression	0.87	0.84	0.75	0.792	0.7944
2.	Decision Tree	0.82	0.71	0.75	0.729	0.633
3.	Random Forest	0.79	0.69	0.69	0.69	0.7283
4.	Support Vector Machine	0.89	0.81	0.84	0.795	0.500
5.	Artificial Neural Network	0.75	0.72	0.75	0.734	0.564
6.	OLWPR	0.91	0.85	0.86	0.86	0.54

False Positive is the number of negative samples predicted as positive and the true negative is the number of negative samples predicted as negative. 10%–40% of anomalies are injected to the testing data. Also the parameters Accuracy, Accuracy error, F1-score, Sensitivity, Specificity, recall, training and testing time are determined for the proposed model and compared with the existing classification techniques like Logistic Regression, Decision Tree, Random forest, Adaboost, SVM, and ANN. Figs. 4, and 5 show the accuracy rate, F1 score respectively.

Table 1 shows the comparison of the parameters accuracy, precision, recall and F1 score obtained in the proposed scheme with other existing techniques.

5. Conclusions

In this paper, an online anomaly detection system is presented to detect anomalous data in WSN. The proposed system predicts the sensor value based on the LWPR regression. The predicted value is then compared with actual sensed value and the error is calculated by using the dynamic threshold. This presented approach is implemented in Python and tested through the real-time dataset collected in IBRL. The experimental results show that the proposed approach has a very high detection rate and low false alarm rates. The proposed method is compared with existing methods like Logistic Regression, Decision Tree, Random forest, Ada boost, SVM, and ANN. It found to be better than the existing in terms of Average RMSE, Average Percentage Error, F1 score and accuracy rate.

Compliance with ethical standards

No fund has been received for this work done.

Ethical approval

This article does not contain any studies in with human participants or animals performed by any of the authors.

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.

References

[1] J.X.Z.S.W. Harish Radhappa, Lei Pan, Practical overview of security issues in wireless sensor network applications, Int. J. Comput. Appl. (2017) 1–12.
[2] S.S. Walaa Elsayed, Mohamed Elhoseny, A. Riad, Self-maintenance model for wireless sensor networks, Comput. Electr. Eng. (2017) 1–14.

[3] M.A.R. Nurfazrina Mohd Zamry, Anazida Zainal, Unsupervised anomaly detection for unlabelled wireless sensor networks data, Int. J. Adv. Soft Comput. Appl. (2018) 172–191.

[4] J.H. Mohiuddin Ahmed, Abdun Naser Mahmood, A survey of network anomaly detection techniques, J. Netw. Comput. Appl. (2016) 19–31.

[5] S.B.Q. Nauman Shahid, Ijaz Haider Naqvi, Characteristics and classifi cation of outlier detection techniques for wireless sensor networks in harsh environments: a survey, Artif. Intell. Rev. (2015) 193–228.

[6] Ye Yuan, Shouzheng Li, Xingjian Zhang, Jianguo Sun, A comparitive analysis of SVM, Naive Bayes and GBDT for Data Faults Detection in WSNS, in: 2018 IEEE International Conference on Software Quality, Reliability and Security Companion, 2018, pp. 394–399.

[7] W.Z.L. Yang, M.Wu, An improved distributed pca-based outlier detection in wireless sensor network, in: Cloud Computing and Security ICCCS 2018, 2018, pp. 37–49.

[8] Colin O'Reilly, Alex Gluhak, Muhammad Imran, Sutharshan Rajasegarar, Online anomaly rate parameter tracking for anomaly detection in wireless sensor networks, in: Mesh and Ad Hoc Communications and Networks (SECON), 2012, pp. 191–199.

[9] P. Gil, H. Martins, F. Januário, Outliers detection methods in wireless sensor networks, Artif. Intell. Rev. (2018) 2411–2436.

[10] P. Cheng, Z. M, Lightweight anomaly detection for wireless sensor networks, Int. J. Distributed Sens. Netw. (2015) 1–8.

[11] S.S.W. Hattagam, N. Teaumroong, Anomaly detection in wireless sensor networks using self- organizing map and wavelets, Int. J. Commun. (2010) 74–83.

[12] W. M, A. Das, An efficient hybrid anomaly detection scheme using k-means clustering for wireless sensor networks, Wirel. Pers. Commun. (2016) 1971–2000.

[13] Yang Zhang, Nirvana Meratnia, Paul J.M. Havinga, Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine, Ad Hoc Netw. (2013) 1062–1074.

[14] X.M.Y.L.H. Zhao, C. Li, Distributed online one-class support vector machine for anomaly detection over networks, IEEE Trans. Cybern. (2019) 1475–1488.

[15] F.L. Zhen Feng Jingqi Fu, Dajun Du, S. Sun, Dajun du s. sun a new approach of anomaly detection in wireless sensor networks using support vector data description, Int. J. Distributed Sens. Netw. (2017) 1–14.