Citation: Rahul Prasad, R K Baghel. Application of bagging ensemble model for fault detection in wireless sensor networks. *Journal of Harbin Institute of Technology (New Series)*. DOI:10.11916/j.issn.1005-9113.2024051

Application of Bagging Ensemble Model for Fault Detection in Wireless Sensor Networks

Rahul Prasad and R K Baghel*

(*Department of Electronics and Communication Engineering, Maulana Azad National Institute of Technology, Bhopal 462003, India*)

Abstract: A Wireless Sensor Network (WSN) comprises a series of spatially distributed autonomous devices, each equipped with sophisticated sensors. These sensors play a crucial role in monitoring diverse environmental conditions such as light intensity, air pressure, temperature, humidity, wind, etc. These sensors are generally deployed in harsh and hostile condition; hence they suffer from different kind of faults. However, identifying faults in WSN data remains a complex task, as existing fault detection methods, including centralized, distributed, and hybrid approaches, rely on the spatio-temporal correlation among sensor nodes. Moreover, existing techniques predominantly leverage classification-based machine learning methods to discern the fault state within WSN. In this paper, we propose a regression-based bagging method to detect the faults in the network. The proposed bagging method is consisted of GRU (Gated Recurrent Unit) and prophet model. Bagging allows weak learners to combine efforts to outperform a strong learner, hence it is appropriate to use in WSN. The proposed bagging method is at first trained at the base station, after which they are deployed at each SN (Sensor Node). Most of the common faults in WSN like transient, intermittent and permanent faults have been considered. The validity of the proposed scheme has been tested using a trusted online published dataset. Using experimental studies, compaed to the latest state-of-art machine learning models, the effectiveness of the proposed model is shown for fault detection. Performance evaluation in terms of false positive rate, accuracy, and false alarm rate shows the efficiency of the proposed algorithm.

Keywords: fault detection; GRU; prophet; deep learning; wireless sensor networks

CLC number: TP277 **Document code**: A **Article ID**: 1005-9113(2025)00-0000-12

0 Introduction

Wireless Sensor Networks (WSNs) are networks of sensor nodes equipped with versatile transducers and communication infrastructure, with the primary goal of monitoring and recording environmental conditions across various locations[1-4]. Each sensor node within a WSN is equipped with one or more sensor units to capture environmental data, which is then processed locally[5-9]. They serve to observe various environmental parameters including temperature, emissions, pressure, distance, and light intensity, transmitting gathered data to a sink node for processing. Renowned for their cost-effectiveness, computational capabilities, and bandwidth efficiency, WSNs have found extensive use in diverse applications. Despite their advantages, WSNs encounter challenges related to storage and processing capacity. WSNs are applied across a wide range of fields including military target tracking and monitoring, environmental hazard surveillance, health monitoring, habitat monitoring, and humanitarian disaster relief efforts[10-14].

Sensor nodes are typically densely deployed in an unstructured manner to fulfil tasks such as sensing, processing, and communication among themselves[2,12]. These unstructured or unordered deployment of the sensor nodes poses a significant challenge for maintaining network's reliability and connectivity. The remote and often inaccessible deployment locations of sensor nodes further exacerbate difficulties in monitoring node failures, especially in disaster-prone areas. Factors such as natural disasters, sensor component failures, battery depletion, and coverage area gaps contribute to the inherent fault-prone character of sensor nodes. This attribute is intensified by the placement of sensor

Received 2024-06-08.

* Corresponding author: Rahul Prasad, Ph.D candidate. Email: bkrahulpb@ gmail.com.

nodes in various and challenging environments. A fault in a sensor node potentially lead to imprecise measurements from the sensors, hence impacting the overall performance of the sensor network. At times, faults within the sensor network might spread to other parts of the network, resulting in a more substantial impact on the overall network. Fundamentally, sensor faults lead to deviation of the sensor values from the actual sensor values, which, if left unattended, can ultimately leads to sensor errors and sensor failures. The occurrence of these errors and failures in the sensor networks can significantly impact the overall computational efficiency of the WSN. Moreover, sensor networks failure in critical application such as health monitoring, enemy tracking, animal tracking, and environmental monitoring can pose a significant impact to environmental, human life and economic losses[2,12].

The faults in a WSN may be classified into three categories depending on the behaviour of the sensor nodes: permanent faults, intermittent faults, and transient faults[1,3,9]. Intermittent and transient faults can be grouped together as short-term faults. Permanent faults persist throughout the entire lifespan of a sensor module, resulting in consistent unwanted behaviour. intermittent fault, on the other hand, can infrequently lead to incorrect outcomes, making it challenging to gather and interpret data effectively. Moreover, due to the intermittent nature of this fault, it has higher likelihood of developing permanent fault in the future. Although less common, transient fault can occur infrequently throughout a sensor module life time. According to research, a sensor module is said to be suffering from permanent fault, if 81 to 100% of its sensor data are persistently incorrect over the entire time instance. A sensor module is said to be suffering from intermittent fault, if 31 to 80% of its sensor data are inaccurate across a given time instance. A transient fault occurs when the percentage of the faulty data ranges from 5 to 30 % for a certain time instance. Since, both intermittent fault and transient fault exhibits dynamic nature in the sensor networks, researchers faces significant challenges when it comes to detecting and diagnosing these faults in the network.

Existing fault detection methodologies, including centralized, distributed, and hybrid approaches, rely heavily on the spatio-temporal correlation among nodes to ascertain fault status[1,8,10,13]. However, as the prevalence of anomalous sensor modules escalates within the WSN, these techniques often falter in accurately discerning fault status. Moreover, to ascertain the faulty state of all the sensor modules in WSN, these fault detection approaches need to send their sensing information to either the adjacent sensor nodes, sink node or to their cluster head. Hence, as the number of nodes in the sensor networks increases, there is a corresponding increase in the energy consumption and detection latency of these approaches, which ultimately results in a decline of the overall network lifetime. Additionally, these methodologies predominantly lean on supervised learning-based classification algorithms for fault detection[3-15]. While these classification algorithms have shown effectiveness in certain contexts, they encounter limitations when applied to WSNs. WSNs inherently capture data from the monitoring field and provide continuous measurements, creating a vast and dynamic dataset. Consequently, addressing faults within WSNs can be framed as a regression problem, where the objective is to predict continuous values representing the state of sensor nodes. However, prevailing fault detection strategies primarily utilize classification-based methodologies, which not fully align with the nature of the data generated by WSNs. Moreover, the reliance on supervised learning approaches necessitates labelled training data, which can be challenging to obtain in real-world WSN deployments due to the dynamic and unpredictable nature of environmental conditions. In this paper, we propose a novel regression-based approach for fault detection, leveraging bagging techniques, aiming to effectively address the issues outlined above. Our proposed bagging method integrates Gated Recurrent Unit (GRU), and prophet models. Bagging, a robust ensemble learning technique, harnesses the collective power of multiple weak learners to surpass the performance of a single strong learner. It effectively mitigates variance, thereby reducing the risk of overfitting during model training. Moreover, our proposed technique involves training the bagging model at the base station and subsequently distributing the trained model to each sensor node within the network. This decentralized approach ensures that every sensor node actively participates in the fault diagnosis process using its locally collected data to determine its own fault status. By independently assessing its fault state based solely on its data, each sensor node contributes to the overall fault detection

process. Consequently, our approach represents an optimal solution for resource-limited WSNs.

The main contributions of the paper are summarized as follows:

1) We propose a regression-based fault detection technique aimed at identifying various types of faults within the network, including permanent, intermittent, and transient faults.

2) We propose a bagging algorithm that does not depend on the sensor readings of neighboring nodes to detect faults in the network.

3) We implement the proposed fault detection algorithms using Google Colab with various parameter settings.

4) We evaluate the performance of the proposed algorithm using common metrics such as false positive rate, energy consumption, detection latency, accuracy, and false alarm rate, and conducted comparative assessments against existing state-of-the-art algorithms.

The rest part of this paper is organized as follows: Section 1 reviews relevant research pertaining to fault detection in WSNs. In Section 2, we present a taxonomy of common faults observed in WSNs. Our proposed fault detection algorithm is discussed in detailed in Section 3. Performance of the proposed method are presented in Section 4. At last, in Section 5, we conclude the paper with recommendations for future works.

1 Related Works

In this section, we discuss some significant contributions and key advancement made by the researchers in this domain.

Saeed et al.[11] proposed a hybrid-based fault detection approach using Extremely Randomized Trees (ET). ETs represent enhanced version of conventional random forest methods, incorporating increased randomness during the tree construction process. The performance of ET is based on methods, which are evaluated against commonly utilized fault detection methods such as random forests, decision tree, support vector machines, and multi-layers perceptron. Evaluation using real world WSN datasets demonstrated that ET performed better than other algorithms in terms of different parameters like accuracy, F1-score. The simulation results indicated that the ET based model effectively managed the inherent uncertainties and the complexities in WSN. However, this approach requires each cluster member to send their sensing information to their cluster heads. Hence, this approach is energy inefficient as significant amount of energy is dissipated in transmitting and receiving messages to the cluster head. Priya et al.[14] proposed a hybrid-based fault diagnosis approach for WSN using machine learning. The paper underscored the significance of feature selection in fault diagnosis for WSN with redundant features, classifying selection methods into filter, wrapper, and embedded types. This approach includes MRMR (Minimum Redundancy Maximum Relevance) for relevance and E-MRMR (Enhanced Minimum Redundancy Maximum Relevance) for feature set refinement, combining wrapper and filter methods. This paper employed SVM (Support Vector Machine) for detecting the anomalous nodes in WSN. The main drawback of this paper lies in its computational complexity. Given the limited resources of the sensor modules, this approach proved to be impractical due to its high computational demands. Swain et al.[1] proposed a fault diagnosis protocol to detect composite faults in the sensor networks. This protocol works in three distinct steps: Initially all the nodes are grouped into clusters, with each cluster is supervised by its cluster head. Subsequently, each cluster head collects data sensed by its cluster members, and leverages the spatial-temporal correlation between sensor modules, and applies statistical based Analysis of Variance (ANOVA) method to identify the presence of faults. In the following step, a PNN (Probabilistic Neural Network) is utilized to classify the type of detected composite fault. This algorithm specifically considers faults due to persistence, such as permanent fault, intermittent fault, and transient fault. While this approach offers effective fault detection for sensor networks, but it suffers from some major drawbacks. A notable drawback is in its implementation, as this approach relies on the cluster heads that must remain fault free for optimum operation.

Biswas et al.[12] proposed a hybrid-based fault diagnosis algorithm designed for sensor network, which leverages both SVM technique and Pearson's correlation coefficient. This algorithm uses the inherent spatial and temporal correlation present in the sensor data to detect the anomalous nodes in WSN. By incorporating Pearson's correlation coefficient with the

self-normalizing features, this model enhances its capabilities to the spatially temporally correlated sensor data in WSN. However, as the unit of the faulty sensor modules escalates in the sensor networks, this algorithm suffers from high false positive rate and lower accuracy. Swain et al.[9] proposed comprehensive fault diagnosis protocol tailored specifically for WSN, with a focus on effectively detecting and diagnosing composite faults like permanent fault, intermittent fault, and transient fault in the network. This fault diagnosis protocol combined a fusion of gradient decent based backpropagation and evolutionary strategies with a feed forward neural network, providing a robust mechanism for diagnosing faults within the WSNs. This protocol was implemented through a well-structured sequence of five stages: communication, clustering, isolation, classification, and fault detection. Noshad et al.[16] presented a novel fault diagnosis protocol in their paper. This research used different classifier models including convolutional neural network, probabilistic neural network, gradient decent, multilayer perceptron, and random forest to identify faults in the WSN. This paper detected different types faults in the networks such as spike, fixed bias, gain and out of bound faults. By using multiple classifier models, this research aimed to increase accuracy, reliability, and classification capabilities, thereby improving the fault detection in WSN. This approach suffers from significant detection delay as the number of nodes in the WSN increases, as the cluster head determines the fault status of each sensor module. Zidi et al.[8] proposed a comprehensive approach in WSN utilizing SVM for fault detection. This supervised learning-based approach works in two steps. Initially, the decision function and the support vectors are determined using the sensor data gathered from the neighbouring sensor modules. Subsequently, this decision function in then deployed on the cluster heads to classify the sensor modules into faulty and non-faulty categories. This research specifically considered faults such as offset, out of bound, gain and struck-at faults. However, this approach has a notable drawback in its energy efficiency, as it requires each cluster members to send their sensing data to the cluster heads.

Gouda et al.[17] proposed a distributed based approach to diagnose faults in the wireless sensor network. This research focuses on identifying the temporary fault like intermittent fault in resource constrained WSN. This paper utilized the inherent spatial-temporal correlations that existed between the nodes to determine the anomalous nodes in the network. This method used the likelihood ratio test to analyse the statistical characteristics of the sensor readings, facilitating informed decision about fault diagnosis. This research offers effective solution for detecting and diagnosing faults in WSN. Sun et al.[18] addressed the challenge of fault diagnosis in WSNs by proposing a centralized based belief rule-based technique. This research acknowledged the inherent uncertainty and variability in sensor readings within the sensor networks. The use of rule-based reasoning and belief function theory provided a robust framework for handling complex scenarios, leading to lower false positive rate. This technique is not scalable for large scale sensor networks. Chanak et al.[15] proposed a comprehensive method for detecting data fault in sensor network. Each sensor module transmits its sensing information to its cluster head, which then computes mean and standard deviation using this data. Using this data, cluster head computes the two-population z test to detect the anomalous nodes in WSN. Sensor node having z score greater than a certain threshold is tentatively marked as soft fault. Cluster members then participate in voting to determine the anomalous status of each cluster member. A node is deemed as faulty if it gets more than half of the cluster members vote as faulty. Singh et al.[19] proposed a novel approach for fault detection, harnessing the capabilities of an autoencoder classifier. This algorithm demonstrated superior performance compared to existing algorithms across various evaluation metrics. The autoencoder architecture consisting of an encoder and a decoder, effectively transformed the input data into lower dimensional space through the encoding process. In the subsequent step, this data was reconstructed to minimize the reconstruction error. This mechanism enabled the algorithm to effectively capture the underlying pattern in the data. By optimizing the autoencoder using Adam optimizer and employing the mean square error loss function, the algorithm accurately approximated the target values, leading to reliable fault detection outcomes. This robust training methodology ensured that the autoencoder effectively learned and captured the complex relationships with the sensor networks data, facilitating accurate fault

detection with lower false positive rate.

2 Network Model

Faults in the sensor networks can be classified based on their duration, indicating the timespan of the fault. These faults can be further classified into various categories, including in Refs. [1, 3, 9]:

2.1 Transient Faults

Transient fault, prevalent in sensor networks, are characterized by their temporary and short timespan nature. These faults occur due to temporary disturbance and disruptions within the systems, which includes factors such as power fluctuation, battery issue, electromagnetic interference, or environmental influences. Unlike permanent fault, transient fault have a short time duration and typically do not persists over an extended period. transient fault can be mathematically represented by the following Eq.(1):

$$\psi_\tau \left[\{ \in_{(\beta_a)} \{ s_i(k_{\beta_a}) \} \neq \{ s_i(\sigma_{\beta_a}) \} \} \& \{ \in_{(\beta_r)} \{ s_i(k_{\beta_r}) \} = \{ s_i(\sigma_{\beta_r}) \} \} \right] \quad (1)$$

where, k_{β_a} denotes the sensor node values that diverge from the actual sensor values σ_{β_a} during the transient time span $\beta_a = \{1,2,3,\ldots,t_s\}$. Conversely, during regular time intervals $\beta_r = \{1,2,3,\ldots,t_r\}$, k_{β_r} sensor node values align with the actual sensor values σ_{β_r}. It is noteworthy that $\beta_a \ll \beta_r$, indicating that the transient divergence occurs over a shorter duration compared to the regular convergence. Additionally, the relationship $\beta = \{1, 2, 3, \ldots, t_r\}$ for the sensor node s_i.

2.2 Intermittent Faults

Intermittent faults in WSNs stem from unpredictable, erratic, and irregular malfunctions of the sensors. These faults manifest in an on-and-off behaviour and may arise due to various factors such as faulty components, unstable ambient conditions, or insufficient specifications. Owing to their random nature, detecting intermittent faults can be challenging; however, patterns of errors can be identified through statistical analysis or consistent monitoring of the parameters. Mathematically, Eq.(2) represents intermittent faults as follows[1]:

$$\psi_\tau \left[\{ \in_{(\beta_1)} \{ s_i(k_{\beta_1}) \} \neq \{ s_i(\sigma_{\beta_1}) \} \} \& \{ \in_{(\beta_2)} \{ s_i(k_{\beta_2}) \} = \{ s_i(\sigma_{\beta_2}) \} \} \right] \quad (2)$$

In the time set $\beta_1 = \{1,2,3,\ldots,t_1\}$, the sensor node values k_{β_1} deviate from the actual sensor value σ_{β_1}. Conversely, during the time set $\beta_2 = \{1,2,3,\ldots,t_2\}$, the sensor node value k_{β_2} equals the actual sensor value σ_{β_2}. These sets, $\{\beta_1, \beta_2\}$, are subsets of the comprehensive time instance set $\beta = \{1, 2, 3, \ldots, t\}$ pertaining to the sensor node s_i. The duration of intermittent faults is typically longer compared to transient faults.

2.3 Permanent Faults

Permanent faults in WSNs are characterized by consistent discrepancies that persist in the system and may exhibit variability in behavior over time. These faults can arise due to factors such as aging, wear and tear, or eventual degradation of components within the sensors. Over time, permanent faults can significantly impact the accuracy, reliability, and integrity of the sensed data, with their behavior being either erratic or progressively deteriorating. Mathematical relations for permanent faults can be expressed through Eq.(3), which outline the underlying characteristics and dynamics of these faults[1].

$$\psi_\tau \{ s_i(k_\beta) \} \neq \psi_\tau \{ s_i(\sigma_\beta) \} \quad (3)$$

Each sensor node s_i is defined by sensor values k_β for each time instance $\beta = \{1, 2, 3, \ldots, t\}$. These sensor modules values k_β drift from the actual corresponding sensor values σ_β due to the presence of permanent faults, which results in the reduced overall performance of the network, inaccurate data communication, and diminished long-term stability. Therefore, mitigating and addressing these faults effectively is important to ensure the reliability and integrity of the data sensed by the nodes in the WSN.

3 Methodology

This section provides the proposed ensemble-based bagging model and the proposed scheme.

3.1 Bagging

The bagging regression is an ensemble－based regression algorithm that randomly selects subsets of the modeling data and combines each regressor's prediction through the averaging method to get the final forecast[20-21]. By introducing randomizing approach into the prediction construction process, bagging regression effectively minimizes bias as well as variance stemming from the underlying regression procedure. Hence, this process demonstrates improved performance, particularly when dealing with complex model with low bias and variance. Bagging operates as

a parallel ensemble approach aimed at mitigating model dispersion by incorporating additional training data. This approach involves non - probabilistic sampling with data replacement from the original set, where certain observations may be repeated in newly created datasets. Despite the size of the training dataset, the predictive power is minimally affected, and fluctuations can be significantly reduced by adjusting predictions to align with desired outcomes. Each dataset is routinely used to train new models, and the ensemble of these models utilizes the average of all forecasts. Because of the distinctive attributes of the aforementioned algorithm, this research combines two techniques, GRU and prophet, within the bagging algorithm to capitalize on their individual advantages. The flowchart of the bagging algorithm provides a clear roadmap of the steps involved in achieving the desired output, as shown in Fig.1.

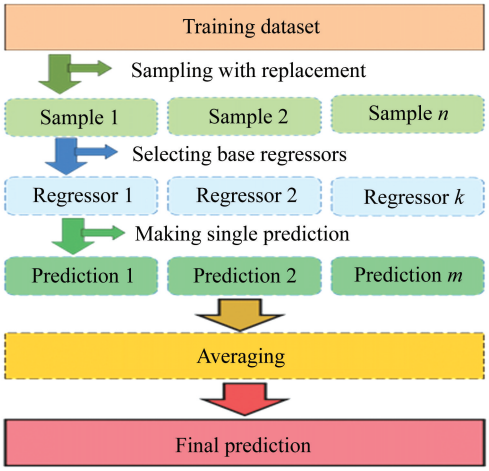

Fig. 1 The structure of the ensemble bagging deep learning model[20]

3.1.1 *Gated Recurrent Unit (GRU)*

The GRU was initially introduced by Cho et al.[22] in 2014, emerging as a prominent algorithm within the realm of Recurrent Neural Networks (RNNs)[23-24]. Primarily, GRU addresses the issue of vanishing gradients inherent in standard RNNs. It is often regarded as a variant of LSTM (Long Short-Term Memory) due to their comparable efficacy in various scenarios. GRU composed of two sigmoid layers: the update gate (z_t) and the reset gate (r_t). These gates are pivotal in easing the vanishing gradient problem and determining the model's output. Fig.2 gives the detailed flowchart of GRU structure, showing the computational process involved in the hidden state and flow of information through the reset and update gates.

1) Update gate

Data processing in GRU initiates with the update gate, where the update gate z_t is calculated for timestep t using the following mathematical formula:

$$z_t = \sigma(w_z \cdot [h_{t-1}, x_t]) \tag{4}$$

here, input (x_t), hidden state (h_{t-1}) are multiplied with their respective weights (w_z) and summed together. In the next step, a sigmoid (σ) is used to normalize the result between 0 and 1. z_t plays an important role in determining which past information should be retained or propagated to the future timesteps.

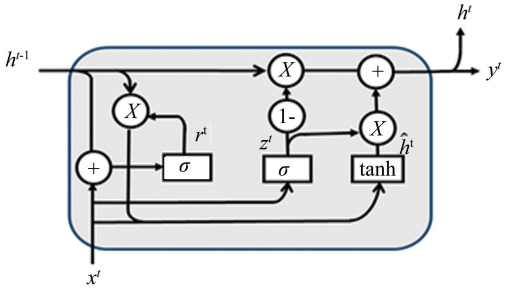

Fig. 2 GRU structure diagram

2) Reset gate

The reset gate r_t is computed at timestep t using the following mathematical Eq.(5).

$$r_t = \sigma(w_r \cdot [h_{t-1}, x_t]) \tag{5}$$

where, x_t, h_{t-1} are multiplied with their respective weights (w_r) and summed together. Then, a σ is used to normailze the output between 0 and 1. r_t assists the GRU model in determining, how much the past information should be disregarded.

3) Current memory content

This process involves the r_t and entails introducing a new memory content to utilize the r_t and retain pertinent information from the past. Mathematically, this is represented by the Eq.(6).

$$\hat{h}_t = tanh(w \cdot [r_t * h_{t-1}, x_t]) \tag{6}$$

here $\hat{h}_t$ denotes the candidate hidden state. The computation begins with the multiplication of the input x_t by its corresponding weight (w). Next, element-wise multiplication is performed between the r_t and the previous output h_{t-1}, allowing only the relevant past information to be retained. Subsequently, the results of these calculations are added together, and a hyperbolic tangent function (*tanh*) function is applied.

4) Final memory at current time step

Finally, the unit computes the h_t vector, which contains information for the current unit and passes it further down the network. The z_t plays a pivotal role in this process, as represented by the mathematical equation:

$$h_t = (1 - z_t) * h_{t-1} + (z_t * \hat{h}_t) \quad (7)$$

in this calculation, if the vector z_t is close to 0, a significant portion of the current content will be disregarded as it is deemed irrelevant for prediction. Conversely, since z_t will be close to 0 at this time step, $(1 - z_t)$ will be close to 1, allowing the majority of the past information to be retained.

3.1.2 Prophet

The prophet model, developed by the Facebook team, is an open-source library utilized for time series forecasting[25-26]. It operates on a decomposable model, offering impressive performance compared to traditional forecasting methods, even with simple parameters. The model has the capability to consider custom seasons and holidays, providing flexibility in modeling complex time series features. Based on Bayesian curve fitting, Facebook prophet is an additive time series model that accommodates various seasonality's, including yearly, monthly, weekly, and daily patterns, as well as holiday effects. Eq.(8) illustrates the four primary components of a Facebook prophet model, which account for trend, seasonality, and holidays.

$$y(t) = g(t) + s(t) + k(t) + e(t) \quad (8)$$

where, $y(t)$ denotes the output value, $g(t)$ represents a trend function that captures non-periodic changes in the time series, $s(t)$ signifies periodic changes such as weekly, monthly, or yearly seasonality, and $k(t)$ accounts for the effects of holidays. The error term $e(t)$ encompasses any irregular or random features in the dataset that cannot be explained by the model. To represent the trend component, a piecewise linear growth model is employed. This model is depicted in Eq.(9).

$$g(t) = (p + a(t)T\delta)t + (m + a(t)T\gamma) \quad (9)$$

The logical growth model is expressed as Eq.(10):

$$g(t) = \frac{C(t)}{1 + exp(-(p + a(t)T\delta)(t - (m + a(t)T\gamma)))} \quad (10)$$

The logical growth model is defined by Eq.(10), where $C(t)$ denoting the non-constant carrying capacity; p denotes the growth rate; $a(t)$ is the adjustment rate that defines the effect of trend change points on the growth model; t represents the time steps or the timeline for the model; δ is the rate of adjustment at the trend change points, allowing flexibility in modelling shifts in the data trends; T indicates trend change points or specific intervals for model adjustments; γ means trend change points; and m denotes an offset parameter. During parameter determination, candidate values are enumerated, and the optimal model parameters are established using the grid search with cross-validation method.

The second term in $y(t)$ denoted as $s(t)$, embodies the seasonal component, which captures cyclic variations occurring on a weekly, monthly, or annual basis. The Facebook prophet model utilizes Fourier series to furnish a versatile model of periodic effects, as depicted in Eq.(11).

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right) \quad (11)$$

$$\zeta = [a_1, b_1, a_2, b_2, \cdots, a_N, b_N] \quad (12)$$

In the prophet model, the smoothing prior for seasonality is imposed using $\zeta \sim \text{Normal}(0, \sigma 2)$. Here, P is the period (time duration for one full cycle); N is the number of harmonics used to approximate the signal; ζ is the vector of the coefficients for the cosine and sine terms, which defines the shape and characteristics of the signal. The estimation of the holiday term $k(t)$ relies on an indicator function to denote whether time t falls within a holiday event i. Each holiday is associated with a parameter u_i, which governs the adjustments to the forecast corresponding to the holiday. Moreover, the Facebook prophet model demonstrates resilience to challenges posed by outliers and missing data.

3.2 Proposed Scheme

Once the bagging algorithm predicts a value, the fault status of the WSN is determined by comparing the actual output with the bagging approximation model. If the disparity between them exceeds a predefined threshold, a sensor fault is identified. Specifically, for a sensor node i, denoted by its real output $\text{output}_i(t)$, and the bagging model output $\text{bagging}_i(t)$, if $\text{bagging}_i(t) - \text{output}_i(t) \geqslant \eta_i$, then a fault is detected at sensor node i.

To classify faults, each node compares its values with the predicted values from the bagging algorithm over the entire time instance. The classification criteria are based on the percentage of erroneous values within a given time instance. If more than 81% to 100% of

the values are erroneous, the node is identified as having a permanent fault. If 31% to 80% of the values are erroneous, the node is classified as having an intermittent fault. If 5% to 30% of the values are erroneous, the node is classified as experiencing a transient fault. This method ensures precise fault classification, enabling appropriate fault management strategies. Table 1 shows the default simulation parameters. The proposed scheme consists of two phases. Fig.3 presents the workflow of the proposed scheme. Initially, the bagging-based prediction model is trained at the sink node. Subsequently, this trained model is deployed at each module to detect its individual anomalous status.

Table 1 Simulation parameters

Parameter	Value
Base estimators	GRU, Prophet
Max_features	1
Max_samples	1
Random_state	None
Verbose	0
Training iterations (GRU)	100
Batch size (GRU)	25
Sequence length (GRU)	60
GRU layers (GRU)	2
Units in GRU layers (GRU)	50 each
Dense layers (GRU)	2
Units in dense layers (GRU)	25, 1
Loss function (GRU)	Mean squared error
Optimizer (GRU)	Adam
Prediction period (Prophet)	30

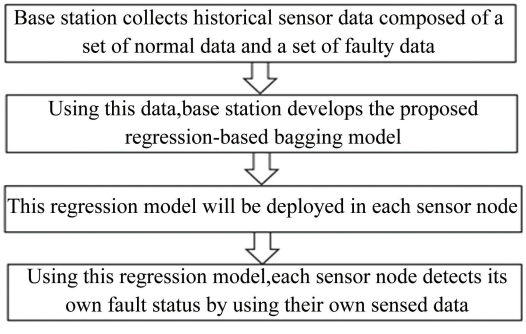

Fig. 3 Workflow of the proposed scheme

4 Results and Discussion

The proposed bagging-based ensemble method is evaluated in this section. First, a brief description of the datasets used is given. The next subsection presents, analyzes, and compares the outcomes of the proposed technique with the existing algorithms.

4.1 Dataset

Dataset published by University of North Carolina-This labelled dataset was published by researchers at the University of North Carolina[8,11,16]. In this dataset, the researchers collected data that consists of measurements of temperature and humidity. For a 6-hour period, these data were measured every 5 s. These data were collected using the TelosB mote from a multi-hop and a simple single-hop WSN. The labelled dataset that we have prepared is a set of sensor measurements into which we have injected different types of faults.

4.2 Result

The results presented in this article were simulated using Matlab and Google Colab. The performance of the proposed scheme has been evaluated by comparing it with the most recent fault detection techniques in WSN. These techniques were outlined and discussed in the literature section, which are ET[11], SVM[8], and PNN (Probabilistic Neural Network)[1]. This comparison is essentially based on five metrics: accuracy, FPR (False Positive Rate), FAR (False Alarm Rate), energy consumption, and detection latency. At each point of fault probability, the average value of that metric has been plotted after injecting each type of fault in our dataset. The first metric is accuracy, and it is defined as below[1,9]:

$$A_{accuracy} = E/S \tag{13}$$

where E represents number of faulty nodes detected, while S represents total number of faulty nodes.

The second comparison metric is FPR, which is defined as below:

$$B_{FPR} = F/S \tag{14}$$

here F stands for number of faulty nodes detected as fault free.

The third comparison metric is FAR, which is defined as below:

$$C_{FAR} = J/K \tag{15}$$

where J stands for number of fault free nodes detected as faulty, K represents total number of faulty free nodes.

The fourth comparison matric is energy consumption, which is defined as the aggregated energy utilized in the fault probability process[1,9]. The final comparison metrics is detection latency, which is defined as the time taken by the sensor network to determine the fault status of all the sensor module in the network.

Fig.4 represents the graph between accuracy versus fault probability. Fig. 5 represents the graph between FPR versus fault probability. Fig. 6 represents the graph between FAR versus fault probability. By increasing the fault probability, the accuracy of all four algorithms decreases. By increasing the fault probability, the FRP, and FAR of all four algorithms rises respectively. The ET, PNN and SVM models are machine learning based classification algorithms. These algorithms fail to detect faulty sensor nodes when the faulty data closely resembles data from normal functioning. So, the accuracy of these algorithms is lower than the proposed algorithm. The ET, PNN and SVM algorithms performed well with permanent faults. However, these algorithms fail to detect transient and intermittent faults due to their high resemblance with the non-faulty data samples. So, these algorithms show a lower detection accuracy and higher FPR than the proposed algorithm. Moreover, ET, PNN and SVM algorithms depend on the sensor readings of the neighboring sensor nodes. So, these algorithms show a lower fault detection accuracy, and higher FAR, FPR than the proposed algorithm. The proposed algorithm employs a regression-based bagging approach to detect faults in WSNs by leveraging the inherent correlations in the sensor data. This method utilizes each sensor node's own data to determine its fault status (does not depend upon neighboring sensor modules data), leading to superior accuracy in fault detection. By aggregating multiple regression models, the bagging algorithm reduces variance and enhances prediction robustness. Hence, this approach outperforms existing algorithms in terms of accuracy, FPR, and FAR.

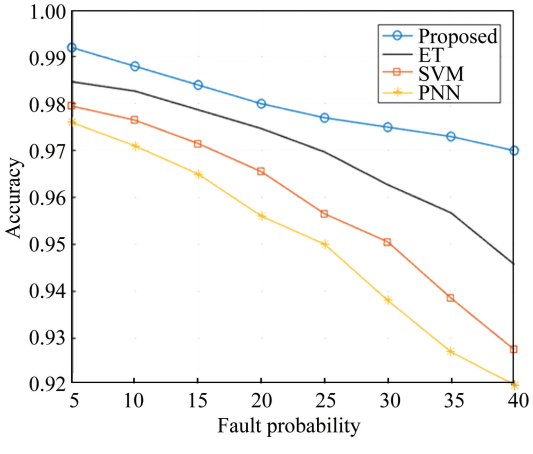

Fig. 4 Accuracy versus fault probability

Figs.7, 8, and 9 depict the performance metrics of accuracy, FPR, and FAR concerning the number of sensor modules, respectively. The proposed model exhibits superior performance compared to existing models in terms of accuracy, false alarm rate, and false positive rate, as illustrated in Figs.7, 8 and 9. Notably, as the number of modules within the WSN increases, the accuracy decreases while the false alarm rate, and false positive rate increase. These results emphasize that the proposed model outperforms existing model, particularly with increasing sensor modules occurrences in the network. Regarding accuracy, the proposed model achieves an average of 97.6%, outperforming the ET algorithm with an average of 96.7%, as well as SVM and PNN algorithms with averages of 95.5% and 95%, respectively. In terms of average FAR, the proposed algorithm, along with ET, SVM, and PNN algorithms, achieves 0.6%, 1.1%, 1.3%, and 1.4%, respectively. Additionally, for average FPR, the proposed algorithm, ET, SVM, and PNN algorithms achieve 2.2%, 3.2%, 4.5%, and 5%, respectively.

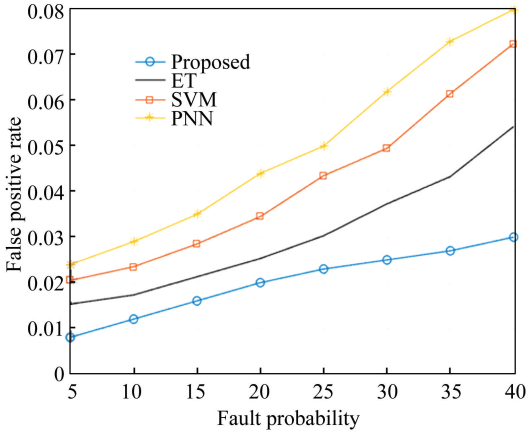

Fig. 5 FPR versus fault probability

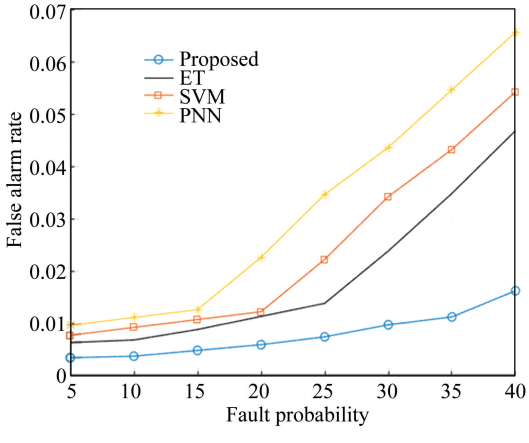

Fig. 6 FAR versus fault probability

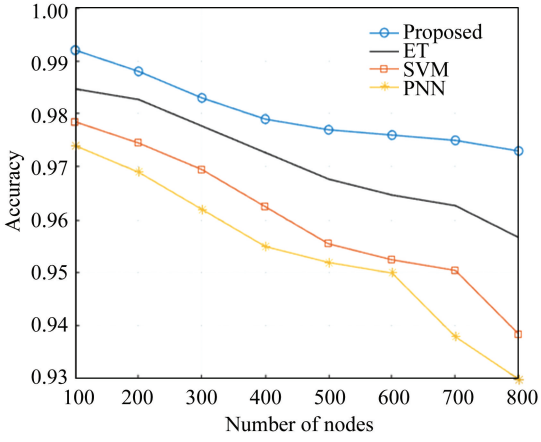

Fig.7 Accuracy versus number of sensor modules

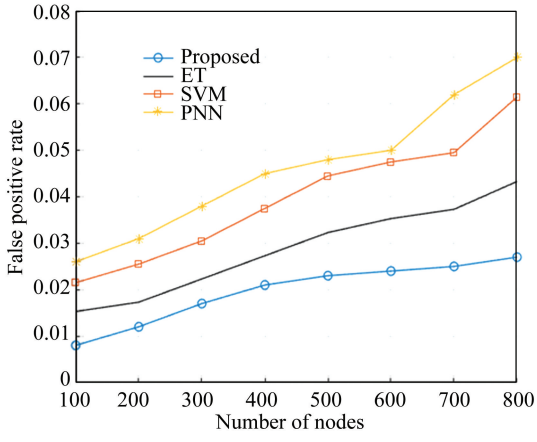

Fig. 8 FPR versus number of sensor modules

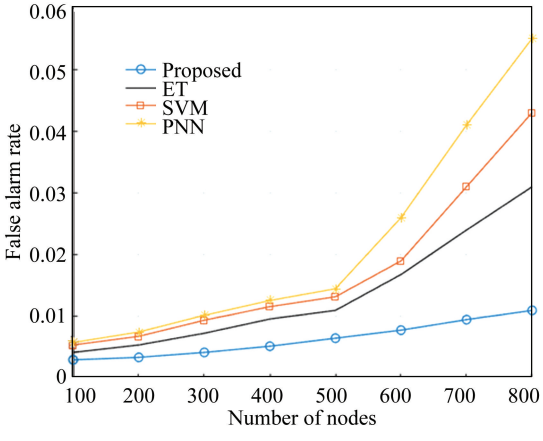

Fig. 9 FAR versus number of sensor modules

Fig.10 illustrates the relationship between accuracy and fault probability for various types of soft faults within the network. As observed, an increase in fault probability leads to a decrease in accuracy across all soft faults, including permanent, intermittent, and transient faults. For instance, the accuracy of permanent fault declines from 0.99 to 0.97, intermittent fault accuracy declines from 0.99 to 0.965, and transient fault accuracy declines from 0.98 to 0.965 with the rise in fault probability. Notably, among the soft faults, permanent fault exhibits the highest accuracy compared to intermittent and transient fault. Figs.11 and 12 present the FPR and FAR against fault probability for different soft faults, respectively. These figures demonstrate that an increase in fault probability corresponds to an increase in both FAR and FPR across all fault types. Particularly, transient fault exhibits higher FAR and FPR compared to other faults due to its unpredictable behaviour. Conversely, intermittent fault, characterized by uniformly random behaviour, shows higher FAR and FPR than permanent fault but lower than transient fault. Permanent fault, known for its certain and continuous behaviour, demonstrates lower FAR and FPR compared to other faults. Analysing the average fault alarm rate values, permanent, intermittent, and transient fault exhibit 0.7%, 0.9%, and 1.5%, respectively. Similarly, the average fault positive rate for permanent, intermittent, and transient fault are observed to be 2%, 2.5%, and 3%, respectively.

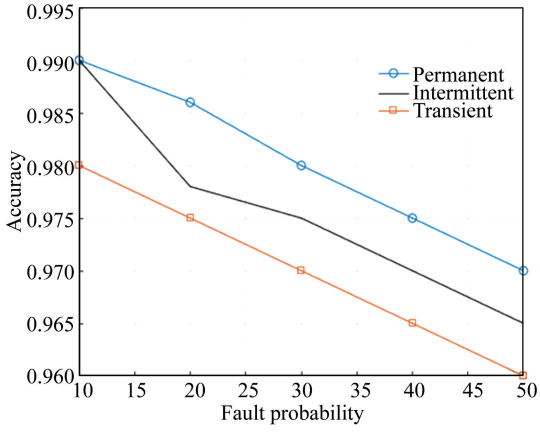

Fig.10 Accuracy of SNN according to fault type

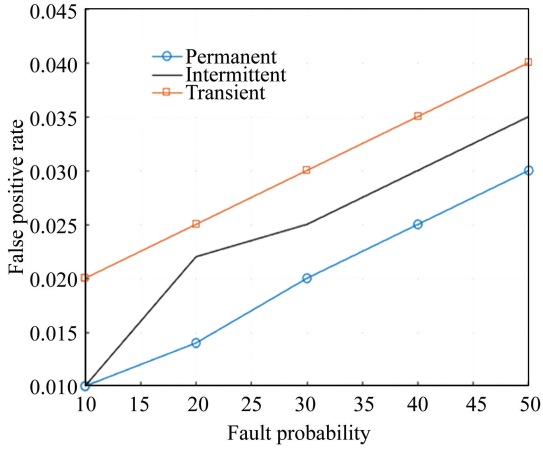

Fig.11 FPR of SNN according to fault type

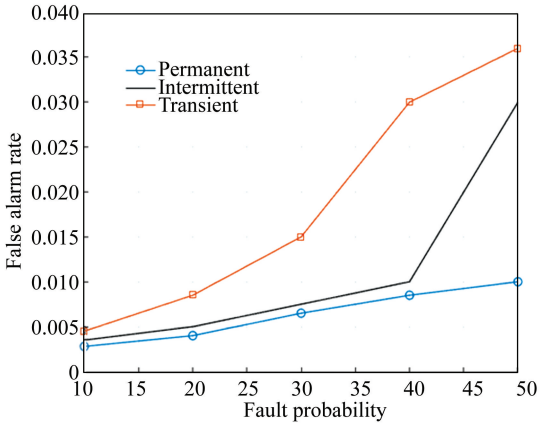

Fig.12 FAR of SNN according to fault type

Table 2 shows the average energy consumption and average fault detection latency for different algorithms (using 100 nodes). ET, SVM, and PNN utilizes hybrid-based fault detection protocols. In hybrid algorithms, sensor modules are organized into clusters with a cluster head. Here, the cluster head collects the sensing information from their cluster members, and using these sensed data can decide the fault status of all its cluster members. Energy consumption of these algorithms depends upon the volumes of messages transmitted to the cluster head and the amount of energy required for data processing. The energy utilized in message transmitting and receiving far exceed that used for the data processing[27-28]. Hence, the average energy consumption for these algorithms are higher compared to the proposed algorithm. In the proposed algorithm each sensor nodes utilizes their own data to determine its fault status (does not depend upon the spatial-temporal correlation between neighbouring sensor modules data). Hence, this algorithm operates without the need of extensive messages exchanges between the nodes and utilizes only energy required for data processing. Hence, the energy consumption of the proposed algorithm is lower than existing algorithms. Similarly, detection latency of these existing algorithm like ET, SVM, PNN depends upon the time required in message transmitting, and data processing. Hence, detection latency of these algorithms is higher than the proposed algorithm. detection latency of the proposed algorithm depends only upon the time required for data processing. Hence, its detection latency is lower than these existing algorithms.

Table 2 Comparison of average energy consumption and detection latency for different algorithms

Algorithm	Average energy consumption	Average detection latency
Proposed	0.0356	0.7
ET	0.424	1.65
SVM	0.445	1.7
PNN	0.462	1.9

5 Conclusions

In this paper, we proposed a novel regression-based bagging diagnostic framework designed to efficiently detect and diagnose faults in WSNs in a timely manner. The proposed bagging method comprises Gated Recurrent Unit (GRU) and prophet models. Leveraging the ensemble learning technique of bagging allows weaker learners to collaborate and outperform a single strong learner, making it particularly suitable for the dynamic and complex nature of WSNs. Furthermore, we conduct a comprehensive performance comparison of our proposed scheme with several machine learning classification algorithms, including SVM, ET, and PNN. Performance analysis is conducted using an online dataset widely trusted within the research community, published by researchers at the University of North Carolina. This dataset encompasses sensor measurements collected from both single-hop and multi-hop networks and encompasses various common faults encountered in WSNs, such as transient, intermittent, and soft permanent faults. Through extensive experimental studies and comparisons with state-of-the-art machine learning models, we demonstrate the effectiveness of our proposed model for fault detection. Performance evaluation metrics, including FPR, energy consumption detection latency, accuracy, and FAR, highlight the efficiency and reliability of our proposed approach.

In the future, our research will explore several ideas to enhance fault detection in WSNs. Firstly, we plan to execute a mobile base station mechanism for fetching sensor module status reports. This approach will mitigate the impact of topological changes on the fault detection process, ensuring the reliability and accuracy of fault detection even in dynamic network environments. Secondly, we will explore the integration of advanced fault detection algorithms to enhance the system's resilience against various sources of noise and disturbances in the network. This will

involve developing innovative techniques that can effectively mitigate fault detection processes.

References

[1] Swain R R, Khilar P M, Bhoi S K. Heterogeneous fault diagnosis for wireless sensor networks. Ad Hoc Networks, 2018, 69: 15-37. DOI: 10.1016/j.adhoc.2017.10.012.

[2] Muhammed T, Shaikh R A. An analysis of fault detection strategies in wireless sensor networks. Journal of Network and Computer Applications, 2017, 78: 267-287. DOI: 10.1016/j.jnca.2016.10.019.

[3] Swain R R, Dash T, Khilar P M. Automated fault diagnosis in wireless sensor networks: A comprehensive survey. Wireless Personal Communications, 2022, 127(4): 3211-3243. DOI: 10.1007/s11277-022-09916-3.

[4] Regin R, Rajest S, Singh B. Fault detection in wireless sensor network based on deep learning algorithms. EAI Endorsed Transactions on Scalable Information Systems, 2021, 8(32): e8. DOI: 10.4108/eai.3-5-2021.169578.

[5] Fan F, Chu S C, Pan J S, et al. An optimized machine learning technology scheme and its application in fault detection in wireless sensor networks. Journal of Applied Statistics, 2023, 50(3): 592-609. DOI: 10.1080/02664763.2021.1929089.

[6] Mahmood T, Li J, Pei Y, et al. An intelligent fault detection approach based on reinforcement learning system in wireless sensor network. The Journal of Supercomputing, 2022, 78(3): 3646-3675. DOI: 10.1007/s11227-021-04001-1.

[7] Rajan M S, Dilip G, Kannan N, et al. Diagnosis of fault node in wireless sensor networks using adaptive neuro-fuzzy inference system. Applied Nanoscience, 2023, 13(2): 1007-1015. DOI: 10.1007/s13204-021-01934-0.

[8] Zidi S, Moulahi T, Alaya B. Fault detection in wireless sensor networks through SVM classifier. IEEE Sensors Journal, 2017, 18(1): 340-347. DOI: 10.1109/JSEN.2017.2771226.

[9] Swain R R, Khilar P M. Composite fault diagnosis in wireless sensor networks using neural networks. Wireless Personal Communications, 2017, 95: 2507-2548. DOI: 10.1007/s11277-016-3931-3.

[10] Loganathan S, Arumugam J, Chinnababu V. An energy-efficient clustering algorithm with self-diagnosis data fault detection and prediction for wireless sensor networks. Concurrency and Computation: Practice and Experience, 2021, 33(17): e6288. DOI: 10.1002/cpe.6288.

[11] Saeed U, Jan S U, Lee Y D, et al. Fault diagnosis based on extremely randomized trees in wireless sensor networks. Reliability Engineering & System Safety, 2021, 205: 107284. DOI: 10.1016/j.ress.2020.107284.

[12] Biswas P, Samanta T. A method for fault detection in wireless sensor network based on Pearson's correlation coefficient and support vector machine classification. Wireless Personal Communications, 2022, 123(3): 2649-2664. DOI: 10.1007/s11277-021-09257-7.

[13] Swain R R, Khilar P M, Bhoi S K. Underlying and persistence fault diagnosis in wireless sensor networks using majority neighbors co-ordination approach. Wireless Personal Communications, 2020, 111: 763-798. DOI: 10.1007/s11277-019-06884-z.

[14] Priya P I, Muthurajkumar S, Daisy S S. Data fault detection in wireless sensor networks using machine learning techniques. Wireless Personal Communications, 2022, 122(3): 2441-2462. DOI: 10.1007/s11277-021-09001-1.

[15] Chanak P, Banerjee I, Bose S. An intelligent fault-tolerant routing scheme for Internet of Things-enabled wireless sensor networks. International Journal of Communication Systems, 2021, 34(17): e4970. DOI: 10.1002/dac.4970.

[16] Noshad Z, Javaid N, Saba T, et al. Fault detection in wireless sensor networks through the random forest classifier. Sensors, 2019, 19(7): 1568. DOI: 10.3390/s19071568.

[17] Gouda B S, Panda M, Panigrahi T, et al. Distributed intermittent fault diagnosis in wireless sensor network using likelihood ratio test. IEEE Access, 2023, 11: 6958-6972. DOI: 10.1109/ACCESS.2023.3236880.

[18] Sun G W, He W, Zhu H L, et al. A wireless sensor network node fault diagnosis model based on belief rule base with power set. Heliyon, 2022, 8(10): e10879. DOI: 10.1016/j.heliyon.2022.e10879.

[19] Singh Y, Rathi R, Prasad R, et al. Fault detection in wireless sensor networks using autoencoder classifier. 2023 6th International Conference on Contemporary Computing and Informatics (IC3I). Piscataway: IEEE, 2023, 6: 1563-1568. DOI: 10.1109/IC3I59117.2023.10397796.

[20] Sharafati A, Asadollah S B H S, Al-Ansari N. Application of bagging ensemble model for predicting compressive strength of hollow concrete masonry prism. Ain Shams Engineering Journal, 2021, 12(4): 3521-3530. DOI: 10.1016/j.asej.2021.03.028.

[21] Amin M N, Iftikhar B, Khan K, et al. Prediction model for rice husk ash concrete using AI approach: Boosting and bagging algorithms. Structures, 2023, 50: 745-757. DOI: 10.1016/j.istruc.2023.02.080.

[22] Chung J, Gulchhre C, Cho K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014. arXiv: 1412.3555. DOI: 10.48550/arXiv.1412.3555.

[23] Li W, Wu H, Zhu N, et al. Prediction of dissolved oxygen in a fishery pond based on gated recurrent unit (GRU). Information Processing in Agriculture, 2021, 8(1): 185-193. DOI: 10.1016/j.inpa.2020.02.002.

[24] Islam M S, Hossain E. Foreign exchange currency rate prediction using a GRU-LSTM hybrid network. Soft Computing Letters, 2021, 3: 100009. DOI: 10.1016/j.socl.2020.100009.

[25] Chaturvedi S, Rajasekar E, Natarajan S, et al. A comparative assessment of SARIMA, LSTM RNN and Fb Prophet models to forecast total and peak monthly energy demand for India. Energy Policy, 2022, 168: 113097. DOI: 10.1016/j.enpol.2022.113097.

[26] Guo L, Fang W, Zhao Q, et al. The hybrid PROPHET-SVR approach for forecasting product time series demand with seasonality. Computers & Industrial Engineering, 2021, 161: 107598. DOI: 10.1016/j.cie.2021.107598.

[27] Panda M, Khilar P M. Distributed self fault diagnosis algorithm for large scale wireless sensor networks using modified three sigma edit test. Ad Hoc Networks, 2015, 25: 170-184. DOI: 10.1016/j.adhoc.2014.10.006.

[28] Panda M, Khilar P M. Distributed Byzantine fault detection technique in wireless sensor networks based on hypothesis testing. Computers & Electrical Engineering, 2015, 48: 270-285. DOI: 10.1016/j.compeleceng.2015.06.024.