Machine Learning Based Intelligent Fire Outbreak Detection in Wireless Sensor Networks

Dhiraj Chaurasia[1], Saikat Kumar Shome[2]

[1] National Institute of Technology Durgapur, M G Avenue, Durgapur – 713209 West Bengal, India
[2] CSIR-Central Mechanical Engineering Research Institute, M G Avenue, Durgapur – 713209 West Bengal, India

dc18u10737@btech.nitdgp.ac.in[1], saikatkshome@cmeri.res.in[2]

Abstract. The outbreak of fire is a serious hazard which is very likely to happen, resulting in loss of lives and property. However, it is arguably possible to predict the outbreak of fire using niches of learning techniques. The traditional smoke alarms generally make use of just one sensor and some threshold to trigger the alarm. Smoke is generated in several forms in daily lives and a single parameter is not reliable to detect an outbreak. This paper mainly focuses on intelligent use of appropriate array of sensors by deploying learning algorithms and AI techniques to reduce false alarm and increase efficiency while keeping cost in mind. Additionally, the application of Random Forrest Classifier has been demonstrated and analyzed in terms of detection rate, prediction score of the model, confusion matrix, logarithmic loss, and AUC. The performance results show that the model was able to predict the outbreak with an error rate of less than 0.2%. The complexity of computation has also been worked out.

Keywords: Fire Detection, Machine Learning, Random Forest, Wireless Sensor Network

1 Introduction

Fires outbreak can happen anywhere ranging from the bedroom to the office and the conditions can vary vastly. According to the statistics released in 2012 by the National Crime Records Bureau, India, fire accounts for about 5.9% (23,281) of the total deaths reported due to natural and un-natural causes. According to an estimate of the major losses reported by the Indian Insurance Companies in the year 2007-2008, about 45% of the claims were due to fire losses [1]. The easiest way to detect a fire is using the smoke detectors or any other similar sensors driven device, which are usually sensitive to ionization or obscuration [2]. While one set of sensing parameters may be suitable for a given preset, it may be almost impractical in another situation. This results in triggering of false alarm in a similarly stimulating situation such as smoking a cigarette or having a barbeque. From 2009 to 2012, excluding malicious triggers, 48% of the fire alarms were false [3]. Of all 6,684,500 fire accidents in the United States, 4,879,685 cases occurred where fire detection systems were installed. However, the evolution and availability of mobile technology, development and sophistication of sensing technologies backed with dynamic and robust learning algorithms brings about great opportunities for making smoke alarms smart and reliable. Let us briefly refer to and discuss the already made contributions in this domain.

Jun Hong et al. [4] proposed a new fire detection system with a multifunctional artificial intelligence framework and a data transfer delay minimization mechanism for the safety of smart cities. The framework includes a set of multiple machine learning algorithms and an adaptive fuzzy algorithm. In [5], the author incorporated Fire Weather Index (FWI) and a novel k-coverage algorithm to detect fires. K-coverage algorithm monitors each point by using k or more sensor nodes to improve fault tolerance. Therefore, some sensors can be put in standby mode to extend network lifetime. Qin Wu et al. [6] proposed an intelligent smoke alarm system that uses ZigBee transmission technology to build a wireless network, uses random forest to identify smoke, and uses E-charts for data visualization. Udak Umoh et al. [7] study employed Support Vector Machine (SVM) in the classification and prediction of fire outbreak based on fire outbreak dataset captured from the Fire Outbreak Data Capture Device (FODCD). Majid Bahrepour et al. proposed a work based on combination of appropriate sensors and deployment of feed forward neural network (FFNN), a Naïve Bayes Classifier. S.R Turns et at. [8] discussed how smoke dust, temperature, and pressure parameters can be used to determine a fire outbreak. Qingjie Zhang et al. [9] proposed a deep learning method for forest fire detection by training a patch fire classifier in a joined deep convolutional neural networks (CNN). Their detector obtained 97% and 90% accuracy on train and test data respectively.

The two main focus of this paper are to make the prediction more reliable and convey the alarm robustly and in time. The research works discussed above make use two general methods to make predictions more accurate. The first approach uses one type of sensor and detects the fire outbreak by a complex algorithm. An example of this approach is the work presented in [10], which uses a flame detection sensor and a fuzzy-wavelet classifier. In contrast, the second approach uses multiple sensors and performs the detection by a simple mathematical operation. The work presented in [11] is an example of the second approach, which uses CO and ionization (ION) sensors and a simple mathematic operation. We have decided to combine these two methods and design by combining multiple sensors to make an appropriate array and use an appropriate algorithm. Furthermore, to serve the communication purpose, after training the model and extracting the variable importance, a web/mobile app may be developed using appropriate API and notification system may be integrated to alarm the admin/provide video-footage/call fire department as required.

The rest of the paper has the following structure. Section 2 discusses the methodology, algorithm proposition and system architecture. Section 3 contains empirical results and discussions. Section 4 consists of the conclusions and Section 5 includes the references.

2 Methodology

The paper is focused on intelligent detection of fire outbreak by making use of machine learning techniques. Machine learning techniques provides our system the ability to automatically learn and improve from experiences and removes the factor of explicit programming. Using machine learning method, a fire outbreak binary classification model can be trained using labelled positive and negative samples. To train the model a dataset obtained from an experiment [12], which imitated several fire hazard situations in a manufactured home, was used. The experiment was monitored in terms of various sensor parameters. These sensor data were feature engineered to obtain an annotated dataset and a classifier model was trained which would classify the real time data to predict an outbreak. A detailed discussion on algorithm and design follows in the subsections below.

2.1 Algorithm Proposition

In this section, there is a brief comparison of various Machine Learning algorithms which are best suited for the purpose of the paper. In order to improve the alarm speed of the system, a classification algorithm is needed to ensure the accuracy of the same time with faster processing speed. According to this requirement, a paper [6], that analyzed the algorithms with excellent classification accuracy and processing speed in the case of small data volume was referred to. More precisely, Bagging, SVM, Decision tree, k-nearest neighbor and random forest classifiers were compared. Below is a brief citation of the results obtained.

The experiment was conducted on a set of cardiotocography (CTG) data [13] as an example. The data has 2126 observations and 23 variables, including the fatal heartrate (FHR) and the uterine contraction (UC) characteristics classified by experts based on guardianship records.

Classification Algorithm	Error Rate	Error rate of train sets (5-fold CV)	Error rate of test sets (5-fold CV)
Decision tree	0.01599247	0.01599253	0.01599337
Bagging	0.01599247	0.01564398	0.01599337
Random forest	0.00047037	0.00047038	0.01270257
SVM	0.01081844	0.01046568	0.01740293
K-nearest neighbor	0.00423330	0.00282222	0.02916432

Table 1. Classification Algorithm effect comparison

A 5-fold cross-validation (CV) method was used to fit each method. To balance the dependent variables, the sample size is arbitrarily divided into 5 substitutions, 5 classification methods' error rate, and 5-fold cross-validation results. The results obtained is cited in the table 1. The results obtained were in agreement with the results of an experiment [14] that compared the classification results of Random Forest and the J48 (a decision tree generating algorithm) on a UCI ML repository dataset where a difference of 26.9% was seen. It can be clearly established from these experiments that the error rate of Random Forest classifier is better than the rest and hence best suited for this experiment. In addition, Random Forest overcomes the problem of overfitting, are less sensitive to outlier data, and eliminates the need for pruning trees. It also decides the variable importance and accuracy automatically.

2.2 Random Forest Classifier

It is an ensemble tree-based learning algorithm. The Random Forest Classifier is a set of decision trees from randomly selected subset of training set. It aggregates the votes from different decision trees to decide the final class of the test object. The random forest algorithm makes good use of randomness (including randomly generated subsample sets, random selection of sub-features), minimizes the relevance of the trees, and improves the overall classification performance, and because the time of each tree is very short and the forest can be parallelized, the random forest classification is very fast.

Assume a random forest classifier $\{h_j\ (x, \theta_i, i = 1, ..., N)\}$; the class label of the classification result is obtained by each decision tree $h_j\ (x, \theta_i)$ and probability averaging for the test instance x. The environmental information (temperature (TCA_1…TCA_N), smoke-obstruction (SMA/B/C/D/E/F_1…SMA/B/C/D/E/F_N), CO concentration, et. al) is collected by the system; the prediction class tag c_p includes non-fire smoke and fire smoke [6]. The classification process is shown below:

$$c_p = \underbrace{argmax}_{c}\left(\frac{1}{N}\sum_{i=1}^{N} I\left(\frac{n_{h_j,c}}{n_{h_j}}\right)\right),$$

$$c_p = \underbrace{argmax}_{c}\left(\frac{1}{N}\sum_{i=1}^{N} I\left(\frac{n_{h_j,c}}{n_{h_j}}\right)\right),$$

where argmax denotes the parameter c with the highest score, N denotes the number of decision trees in the random forest, $I(*)$ denotes the exponential function, n_{h_j}, c denotes the classification result of the decision tree for the class c, n_{h_j} represents the number of leaf nodes of the decision tree h_j, and w_i represents the weight of the i tree in the random forest [15].

2.3 System Architecture

In fire outbreak detection system an array of temperature, smoke obscuration, CO, CO_2 and O_2 concentration sensors has been used. These sensors are connected to a micro-controller unit. The microcontroller interfaces with the system peripherals (sensors) through the input/output (I/O) pins. Values sensed by these sensors are read program running on the microcontroller's processor. The read parameter values are sent in a request parameter as a sequence of text via a Wi-Fi module to a web interface that captures, pre-processes, stores the data for use by the system. The stored dataset is visualized in order to descend their relationship. 80% of the data are used for training dataset while 20% of the dataset are employed for testing and validation. The conceptual architecture of the fire outbreak detection system is presented in Figure 1.

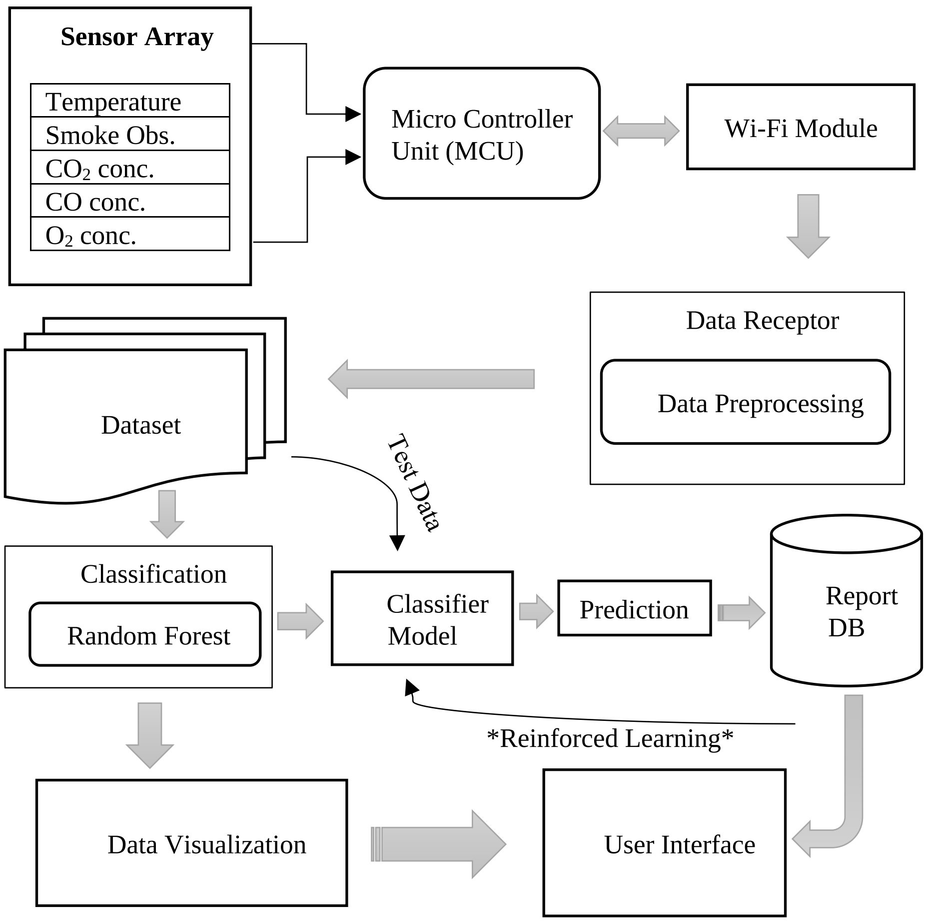

Fig. 1. Conceptual Architecture of the Outbreak Detection System

3 Empirical Results and Discussion

To evaluate the proposed approach, the set of data was fed to the trained model and the obtained results were analyzed as discussed below. Subsection 3.1 discusses the dataset, 3.2 consists of the performance metrics employed, 3.3 consists of the experimental reports, while Subsection 3.4 discusses the computational complexity consideration.

3.1 The Dataset

The Dataset used in the research was obtained from the NIST Website `https://www.nist.gov/el/nist-report-test-fr-4016`. Different fire hazard situations like smoldering chair, flaming mattress, cooking oil fire etc. were recreated in controlled experimental and the concentrations of CO, CO_2, and O_2, smoke obscuration, and temperature at multiple locations in the structure were recorded. A total of 7 datasets (sdc01...sdc07) were combined to obtain a total of 5450 entries, 80% of which was used for training the model and 20% for testing the model. The compiled csv can be obtained from `https://github.com/dch239/Fire-Outbreak-Detection/blob/master/sdcCompiled.csv`. It may be noted that only 6 sensor data (TCB_1, TCFIRE, SMB_1, GASB_1, GASB_3 and GASB_6) were fed to the model based on the correlation with output and feature importance graph as shown in Fig. 2 and Fig. 3. The hookup to the column headings and units can be found here `https://www.nist.gov/document/hookupmh1csv`.

3.2 The Performance Metrics

The basic score evaluation of the model was done based on the model score, logarithmic loss, and the AUC ROC curve. To further visualize the performance of the model, a confusion matrix was obtained.

$$\text{Model Score} = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}$$

Logarithmic Loss penalizes the false classifications. Log loss nearer to 0 indicates higher accuracy. If there are N samples belonging to M classes (M=2 for binary classification), then the Logarithmic Loss is calculated as below.

$$\text{Logarithmic Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * \log\,(p_{ij})$$

where,
y_{ij}, indicates whether sample i belongs to class j or not
p_{ij}, indicates the probability of sample i belonging to class j

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the 'signal' from the 'noise'. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

When AUC = 1, then the classifier can perfectly distinguish between all the Positive and the Negative class points correctly. When 0.5<AUC<1, there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values. When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points. Meaning either the classifier is predicting random class or constant class for all the data points.

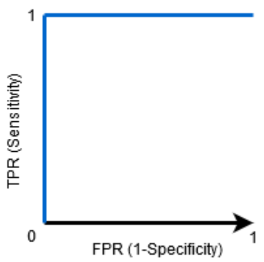

Fig. 2. AUC = 1

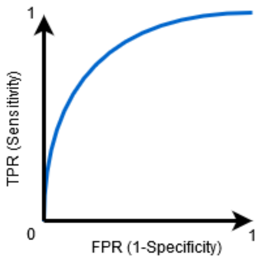

Fig. 3. 0.5 < AUC < 1

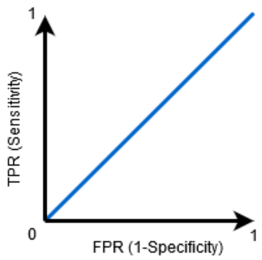

Fig. 4. AUC = 0.5

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.
Mathematically,

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \text{ ,}$$

Where,
True Positive (TP): Observation is positive and is predicted to be positive.
False Negative (FN): Observation is positive but is predicted negative.
True Negative (TN): Observation is negative and is predicted to be negative.
False Positive (FP): Observation is negative but is predicted positive.

3.3 Experimental Results

The evaluation of the proposed algorithm on the above-mentioned dataset is discussed in this section. Let us first analyze the dataset. To visualize the relationship between the sensor input and the labelled output of the dataset, a correlation matrix was obtained which is shown in Fig. 5. The correlation ranges from -1 to 1 which corresponds to maximum negative correlation to maximum positive correlation. The sensor input correlation in this experiment ranges between -0.11 to 0.30 as shown. Random Forest Classifier decides the variable importance automatically which can be visualized using Fig. 6. The rise pattern of cumulative importance curve helps us understands that the temperature sensor place right above the burning object and smoke shares the greater weight and the rest follows as shown.

The dataset had 5450 data records, which were divided to 4360 training data and 1090 test data. The training dataset was fed to the model and after training, the performance metrics were employed for evaluation. A model score of 0.9899082568807339 and a logarithmic score of 0.34855792618245773 was recorded. To null out the seed factor the experiment was performed several times and the mean data was used the metric. An AUC ROC curve was obtained with an AUC score of 0.9965160067102365. The curve is shown in Fig. 7. It is clear from the curve that the model has very efficiently classified the test data with a very less error rate which can be seen in the confusion matrix obtained. Fig. 8 shows the confusion matrix of the prediction where TP = 797, FN = 282, TN = 9 and FP = 2, which corresponds to an accuracy near about the mean model score.

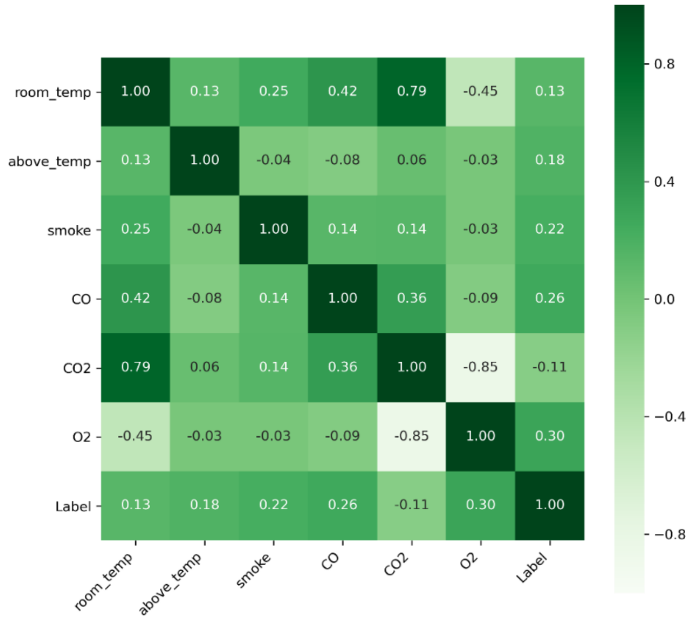

Fig. 5. Correlation Matrix Plot

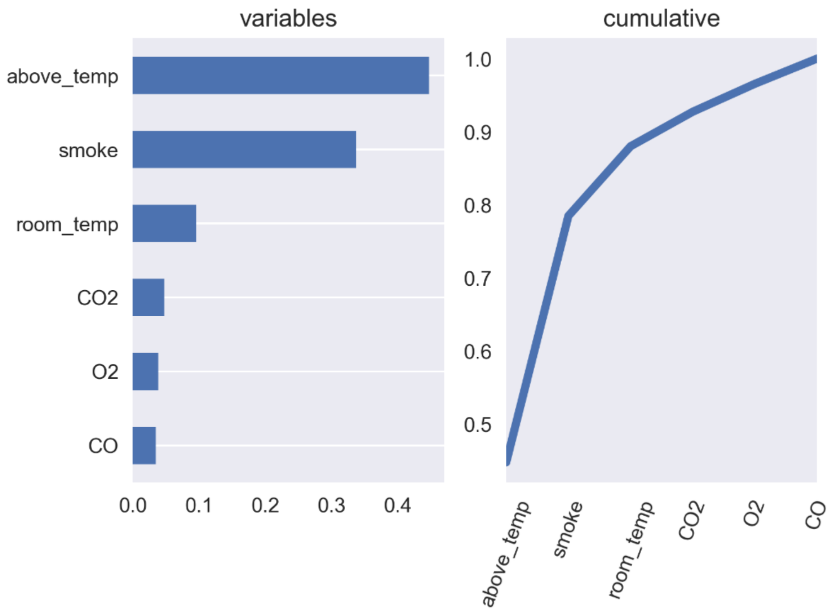

Fig. 6. Features Importance

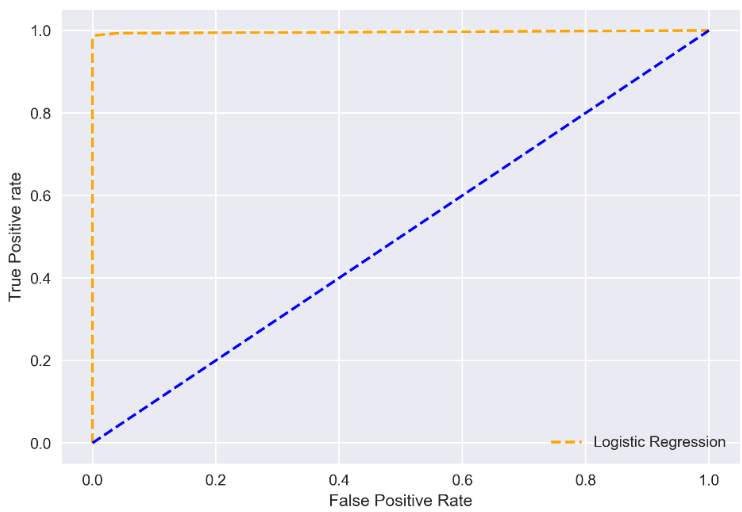

Fig. 7. AUC-ROC Curve of the classifier

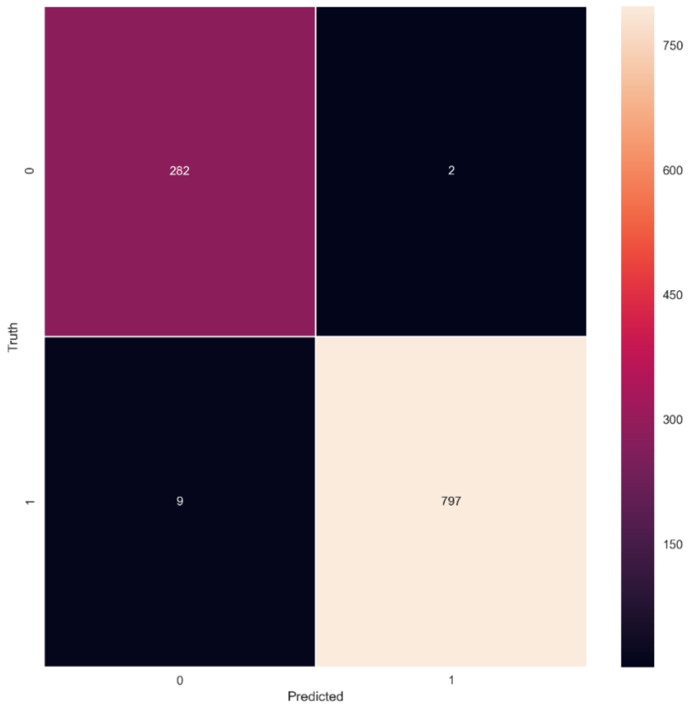

Fig. 8. Confusion Matrix of the Classified Test Dataset

3.4 Computational complexity consideration

Random forests or random decision forests are an ensemble learning method for classification. It works by constructing a multitude of decision trees. Let us calculate the time complexity for building a complete decision tree which is not pruned. If n is the number of records and v is the number of variables/attributes, we get O (v * n log(n)). Assuming the number of trees to be built in random forest ensemble is n_{tree} and at each node m_{try} variables are to be sampled, the complexity to build one tree would be O (m_{try} * n log(n)). If a random forest having n_{tree} number of trees is to be built, the time complexity would be O (n_{tree} * m_{try} * nlog(n)). Assuming the depth of the tree is O (n log(n)), which is the worst-case scenario the above result is obtained. But often, the build process of a tree terminates before this as computational complexity increases.

Furthermore, the depth of the trees in our random forest can also be restricted. If the maximum depth of our tree is restricted to "d", then the complexity calculations can be optimized to O (n_{tree} * m_{try} * d * n). Considering the complexity for random selection of variables that needs to be done at each node, an additional O (v * d * n_{tree}) may be factored.

4 Conclusions

In this paper, an implementation method of learning algorithm-based smoke alarm system, including system architecture, discussion of the classification algorithm deployed, and visualization of the results obtained has been demonstrated. The central purpose of the paper was to find a solution for the false alarm of the tradition alarm system which has been addressed. It may also be mentioned that the dataset has about 118 parameters which is not feasible to build an efficient alarm system, but the model can be engineered to use the desired set of parameters suitable to the preset. The results indicate that the Random Forest Classifier-based fire outbreak detection has been able to provide a solution to the problems associated with fire outbreak detection by providing continuous monitoring of environmental changes and deploying self-learning intelligent algorithm which can save lives and property. The sensor array can also be updated with time as nothing is hardcoded and the algorithm is flexible.

5 REFERENCES

1. R.R Nair. (2012) Fire Safety in India – An Overview. Safety and Health Information Bureau, Vashi, Navi Mumbai.
2. Bahrepour, Majid & Meratnia, Nirvana & Havinga, Paul. (2009). Use of AI Techniques for Residential Fire Detection in Wireless Sensor Networks. IEEE Journal of Quantum Electronics - IEEE J QUANTUM ELECTRON. 475. 311-321.
3. Karter M.J. False Alarm Activity in the US 2012. National Fire Protection Association; Quincy, MA, USA: 2013.
4. Park, Jun & Lee, Seunggi & Yun, Seongjin & Kim, Hanjin & Kim, Won-Tae. (2019). Dependable Fire Detection System with Multifunctional Artificial Intelligence Framework. Sensors. 19. 2025. 10.3390/s19092025.
5. Bagheri, M., Efficient K-Coverage Algorithms for Wireless Sensor Networks and Their Applications to Early Detection of Forest Fires, in Computing Science. 2007, SIMON FRASER UNIVERSITY. p. 75.
6. Wu, Qin & Cao, Jiashuo & Zhou, Chuang & Huang, Ji & Li, Zhuo & Cheng, Shin-Ming & Cheng, Jun & Pan, Guanghui. (2018). Intelligent Smoke Alarm System with Wireless Sensor Network Using ZigBee. Wireless Communications and Mobile Computing. 2018. 1-11. 10.1155/2018/8235127.
7. Umoh, Uduak & Udo, Edward & Emmanuel, Nyoho. (2019). SUPPORT VECTOR MACHINE-BASED FIRE OUTBREAK DETECTION SYSTEM. International Journal on Soft Computing, Artificial Intelligence and Applications. 08. 01-18. 10.5121/ijscai.2019.8201.
8. S.R.Turns, An Introduction to Combustion,vol.287, McGraw-Hill, New York, NY, USA, 1996.
9. Zhang, Qingjie & Xu, Jiaolong & Xu, Liang & Guo, Haifeng. (2016). Deep Convolutional Neural Networks for Forest Fire Detection. 10.2991/ifmeita-16.2016.105.
10. Thuillard, M. Application of Fuzzy Wavelets and Wavelets in Soft Computing Illustrated with the Example of Fire Detectors in Wavelet Applications VII. 2000. 6.
11. Gottuk, D.T., et al., Advanced fire detection using multi-signature alarm algorithms. Fire Safety Journal, 2002. 37(4): p. 381-394.

12. Richard D. Peacock, Jason D. Averill, Richard W. Bukowski, & Paul A. Reneke. Home Smoke Alarm Project, Manufactured Home Tests at Building and Fire Research Laboratory, National Institute of Standards and Technology.
13. L. Breiman, "Bagging predictors" Machine Learning, vol.24, no.2, pp. 123–140, 1996.
14. Ali, Jehad & Khan, Rehanullah & Ahmad, Nasir & Maqsood, Imran. (2012). Random Forests and Decision Trees. International Journal of Computer Science Issues (IJCSI). 9.
15. J. S. Lee, Y. W. Su, and C. C. Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society, IECON 2007, pp. 46–51, IEEE, 2007.