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. The traditional fire 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 the intelligent use of sensors by deploying learning algorithms and AI techniques to reduce false alarm and increase efficiency. Additionally, the application of such a system 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, Intelligent Fire Alarm

1 Introduction

Fires outbreak can happen anywhere ranging from the bedroom to the office and the conditions can vary vastly. According to a report published in 2012 by the National Crime Records Bureau, India, fire accounted for 5.9% (23,281) of the deaths reported. Another report of the biggest losses of the Indian Insurance Companies reported in 2007-2008, 45% of the claims were due to fire hazards [1]. The simplest way to detect a fire is by using smoke detectors, which is generally sensitive to ionization or obscuration [2]. While one set of sensing parameters may be suitable for a given preset, it may be impractical in another situation. This results in the triggering of false alarms. According to a report of NFPA, in the years 2009-2012, 48% of the fire alarms were false, excluding malicious triggers [3]. About 6,684,500 fire accidents happened in the US, and 4,879,685 of them had fire detection systems installed.

However, the evolution of sensors and robust learning algorithms brings about great prospects for making smoke alarms smart and reliable. Let us briefly discuss the contributions made in this domain. Jun H.P. et al. proposed a dependable fire detection system with a multifunctional AI framework which includes a set of machine learning algorithms and an adaptive fuzzy algorithm [4]. M. Bagheri et al. combined a novel k-coverage algorithm and the Fire Weather Index to detect fires [5]. To enhance fault tolerance and put unused sensors in standby, the algorithm uses k (or more) sensor nodes to screen every point. Qin Wu et al. proposed a smoke alarm system that uses an ensemble decision tree algorithm to detect smoke and ZigBee communication protocol to make a wireless network [6]. Udak Umoh et al. used Support Vector Machine to classify and predict fire outbreak [7]. S.R Turns et at. discussed how smoke, dust, temperature, and pressure parameters can be used to determine a fire outbreak [8]. Zhang et al. suggested a deep learning technique for forest fire detection by training a model of fire patch classifier in a deep joined CNN [9].

The works listed above follow two general approaches to make predictions more accurate. The first is to use only one kind of sensor but a complex algorithm to detect a fire outbreak. This approach can be seen in the work presented in [10], where a flame sensor is the only module used but a complex algorithm (fuzzy-wavelet classifier) is deployed. In contrast, the second approach is to use a set of sensor modules, but simple mathematics for detection of fire. An example of this approach can be seen in the work presented in [11], were CO concentration and ION sensors are used but a simple mathematic operation is used to judge fire outbreaks.

In this paper, the two methods have been combined to devise an approach that overcomes the shortcomings of the individual approaches like unreliability on only one kind of sensor and time taken to respond by a complex algorithm-driven system. An appropriate array of sensors was prepared based on the sensor variable importance and sensor feedback correlation. This array was then used on a contextually robust and fast machine learning algorithm-driven IoT system. The details are discussed in the following subheadings.

2 Methodology

The paper is focused on intelligent detection of fire outbreak by making use of machine learning techniques. Machine learning techniques provide our system with the ability to automatically learn and improve from experiences and removes the factor of explicit programming. To train the model a dataset obtained from an experiment [12], which imitated several fire hazard situations in a manufactured home, was used. 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

The "No Free Lunch" theorem states that there is no such algorithm that is optimal in all cases [13]. To make the system contextually robust, a classification algorithm that ensures the accuracy of prediction and has fast processing speed for a small volume dataset at the same time is required. To find the best-suited algorithm, several classification algorithm models were trained on the dataset and a cross-validation test was performed. The experiment was conducted on the dataset which is discussed in section 3.1.

Classification Algorithm	Error Rate	Error rate of train sets (5-fold CV)	Error rate of test sets (5-fold CV)
Random Forest	0.00917431	0.00733893	0.01560482
KNN	0.01284404	0.01330302	0.01929565
Decision Tree	0.01100917	0.01238506	0.02201854
Bagging	0.01100917	0.00940367	0.01835297
SVM	0.10893766	0.12001146	0.12363654

Table 1. Classification Algorithm effect comparison

A 5-fold CV method was used to fit each method. It can be observed that only bagging closes the accuracy of Random Forest as Random Forest is in fact a type of Bagging Algorithm, but it uses a subset of randomly selected features instead of all features like Bagging [14]. The margin is close in the experiment because there are only a few features. These results were also in agreement with the results of an experiment [15] that compared Random Forest and J48 (a decision tree generating algorithm) on a UCI ML repository dataset where a difference of 26.9% was seen. It can be established from these experiments that the Random Forest classifier is better suited for this experiment.

2.2 Random Forest Classifier

Random Forest is an ensemble learning algorithm that builds a multitude of decision trees from a randomly selected subset of the training set with different features. The key idea is to build a large number of uncorrelated decision trees and sum up the votes from all decision trees to decides the class of the test object. As the time taken by each tree to spit out the result is now lesser and the forest can be parallelized, the algorithm can classify at a much faster rate. Besides, Random Forest overcomes the problem of overfitting, is less sensitive to outlier data, and eliminates the need for pruning trees. It also decides the variable importance and accuracy automatically.

Let us assume such a classifier $\{h_j(x, \theta_i, i = 1, ..., N)\}$, where the label classification is attained by each decision tree $h_j(x, \theta_i)$ and the probability averaging for the test object is x. The prediction class tag c_p outputs 0 or 1, 1 being fire outbreak.

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

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

where argmax implies the parameter c with the maximum score, N is the total number of decision trees constructed at training time, $I(*)$ is the exponential function, (n_{h_j}, c) is the classification result for the object class c, n_{h_j} denotes the number of leaf nodes of the decision tree h_j, and w_i denotes the weight of the i^{th} tree in the forest [16].

2.3 System Architecture

In the outbreak detection system, an array of temperature, smoke obscuration, CO, CO_2, and O_2 concentration sensors has been used. The array of these sensor modules is connected to a microcontroller unit through the I/O pins. The values obtained from these sensors are fed to a program installed on the microcontroller's processor. The data is then sent in a request parameter as a string through a Wi-Fi module to a data receptor (web interface) that captures and pre-processes the data. The data is then fed to the trained classifier model which makes a real-time prediction triggering the alarm and reports the results to a Report Database which can be accessed from the User Interface(UI). The UI can be used for visualization or reinforcing the learning algorithm.

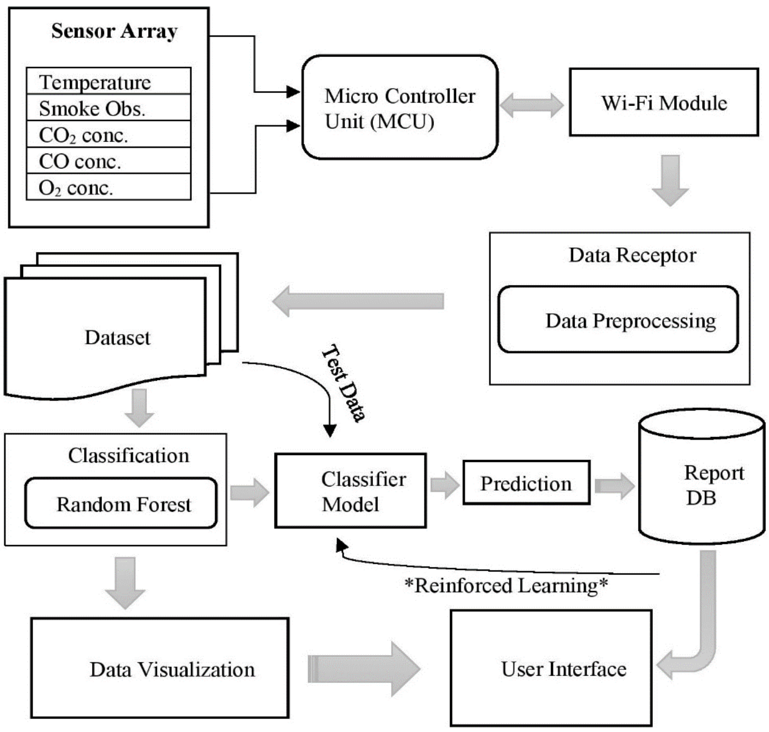

Fig. 1. Conceptual Architecture of the Outbreak Detection System

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 a smoldering chair, flaming mattress, cooking oil fire, etc. were recreated in controlled experimental and the concentrations of CO, CO2, and O2, 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 at https://github.com/dch239/Fire-Outbreak-Detection/blob/master/sdcCompiled.csv. It may be noted that only 5 sensor parameters were fed to the model based on the correlation with output and feature importance graph as discussed in the results.

3.2 The Performance Metrics

The 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.

Logarithmic Loss penalizes false classifications. Log loss nearer to 0 indicates higher accuracy. If N samples are 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 and p_{ij} indicates the probability of sample i belonging to class j.

The Receiver Operator Characteristic (ROC) curve is a metric for the assessment of binary classification models. It is a plot of True Positive Rate against False Positive rate at the threshold values and basically tells the signal from the noise. The Area Under the Curve is a measure of the probability of detection or classification. Given we have only two classes, positive (1) and negative (0), AUC = 1 implies that the classifier has perfectly classified all the positive and negative test objects. AUC between 0.5 and 1 implies a good chance of correct classification. An AUC of 0.5 implies that the classifier cannot distinguish between positive and negative class objects which means that either the classifier is predicting randomly or static class for all the test objects.

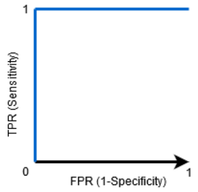

Fig. 2. AUC = 1

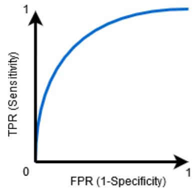

Fig. 3. 0.5<AUC<1

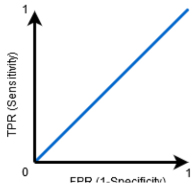

Fig. 4. AUC = 0.5

A confusion matrix is a table of True Positive (TP), False Negative (FN), True Negative (TN), and False Positive (FP). It is often used to visualize the accuracy of a classification model on a set of labeled test data.

3.3 Experimental Results

Let us first analyze the dataset. To visualize the relationship between different sensor parameters, 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. As seen in the plot, the correlations range from -0.85 to 0.36. This implies that the parameters are strongly correlated. It can be observed that some parameters are positively correlated (increase in smoke results in an increase of CO and CO_2 concentrations) while some are negatively correlated (increase in CO_2 concentration (possibly due to fire outbreak) results in a rapid decrease of O_2 concentration). Random Forest Classifier decides the variable importance automatically which can be seen in Fig. 6. The rising pattern of the cumulative importance curve helps us understand that the temperature sensor placed right above the burning object and smoke sensor shares the greater weight and the rest follows as shown.

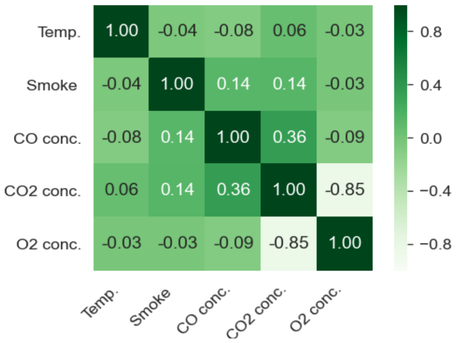

Fig. 5. Correlation Matrix of Sensor Parameters

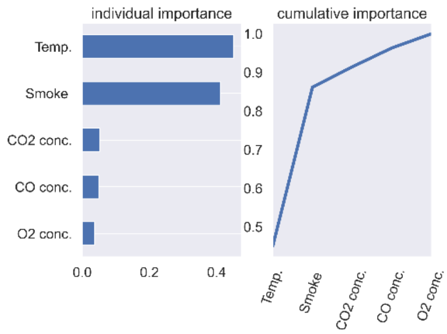

Fig. 6. Feature Importance

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 0.98990 or 98% and a logarithmic loss of 0.34856 was recorded. This indicates that our model has been able to classify the test dataset with very good accuracy. A log loss close to 0 ensures that the uncertainty of the probability spitted by our model is less, hence better the accuracy. An AUC ROC curve was obtained with an AUC score of 0.99892. The curve is shown in Fig. 7. This AUC score near to 1 supports the model score and log loss and further ensures the class prediction accuracy. It is clear from the scores that the model has 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(Actual: true, Predicted: true) = 822, FN(Actual: false, Predicted: false) = 257, TN(Actual: true, Predicated: false) = 5, and FP(Actual: false, Predicted: true) = 6.

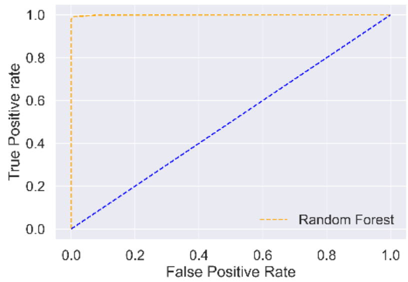

Fig. 7. AUC-ROC Curve

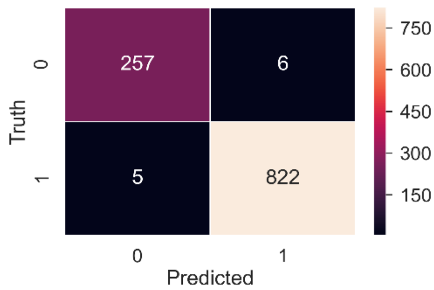

Fig. 8. Confusion Matrix

3.4 Computational complexity consideration

Random forests work by building a multitude of decision trees. Let us calculate the time complexity for building a complete decision tree that is not pruned. If n is the number of records and v is the number of variables/attributes, we have 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 of a learning algorithm-based smoke alarm system, with system architecture, discussion of the algorithm deployed, and visualization of the results obtained has been presented. The central purpose of the paper was to find a solution for the false alarm of the traditional alarm system which has been addressed. The results indicate that the Random Forest Classifier-based fire outbreak detection has been able to provide a solution to the problems associated with existing fire outbreak detection systems.

5 REFERENCES

[1] R.R Nair.: Fire Safety in India - An Overview. Safety and Health Information Bureau, Vashi, Navi Mumbai, (2012).

[2] Majid B., Nirvana M., and Paul H.: Use of AI Techniques for Residential Fire Detection in Wireless Sensor Networks. IEEE Journal of Quantum Electronics., vol. 475, pp. 311-321, (2009).

[3] Michael J. Karter.: False Alarm Activity in the US 2012. National Fire Protection Association Fire Experience Survey, pp. 1-8, (2013).

[4] Jun H.P., Seunggi L., Seongjin Y., Hanjin K, and Won-Tae K.: Dependable Fire Detection System with Multifunctional Artificial Intelligence Framework. Sensors., vol. 19, article no. 9: 2025, (2019).

[5] M. Bagheri., and M. Hafeeda.: Efficient K-Coverage Algorithms for Wireless Sensor Networks and Their Applications to Early Detection of Forest Fires, Technical Report, Computing Science. 2007, Simon Fraser University, (2007).

[6] Qin W., Jiashuo C., Chuang Z., Ji Huang., Zhuo L., Shin-Ming C., Jun C., and Guanghui P.: Intelligent Smoke Alarm System with Wireless Sensor Network Using ZigBee. Wireless Communications and Mobile Computing., vol. 2018, pp. 1-11, (2018).

[7] Uduak U., Edward U., and Nyoho E.: Support Vector Machine-Based Fire Outbreak Detection System. International Journal on Soft Computing, Artificial Intelligence and Applications., vol. 08, pp. 1-18, (2019).

[8] S.R. Turns.: An Introduction to Combustion. McGraw-Hill, New York, NY, USA. vol.287, (1996).

[9] Qingjie Z., Jiaolong X., Liang X., and Haifeng G.: Deep Convolutional Neural Networks for Forest Fire Detection. Proceedings of the 2016 International Forum on Management, Education and Information Technology Application. Atlantis Press., pp. 568-575, (2016).

[10] Mark Thuillard.: Application of Fuzzy Wavelets and Wavelets in Soft Computing Illustrated with the Example of Fire Detectors. Wavelet Applications VII., vol. 4056, pp. 351-361, (2000).

[11] Daniel T.G., Michelle J.P., Richard J.R., and Craig L.B.: Advanced fire detection using multi-signature alarm algorithms. Fire Safety Journal., vol. 37, pp. 381-394, (2002).

[12] Richard D. P., Jason D. A., Richard W. B., and Paul A. R.: Home Smoke Alarm Project, Manufactured Home Tests at Building and Fire Research Laboratory. National Institute of Standards and Technology. NIST Report of Test FR 4016, (2005).

[13] David H. W. and William G. M.: No Free Lunch Theorems for Optimization., IEEE Transactions On Evolutionary Computation., vol. 1, no. 1, pp. 67-82, (1997).

[14] Leo Breiman.: Bagging predictors. Machine Learning., vol. 24, no. 2, pp. 123–140, (1996).

[15] Jehad A., Rehanullah K., Nasir A., and Imran M.: Random Forests and Decision Trees. International Journal of Computer Science Issues (IJCSI)., vol. 9, issue. 5, no. 3, pp. 272-278, (2012).

[16] Jin-Shyan L., Yu-Wei S., and Chung-Chou S.: A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society., pp. 46–51, (2007).