Detection of forest fires using machine learning

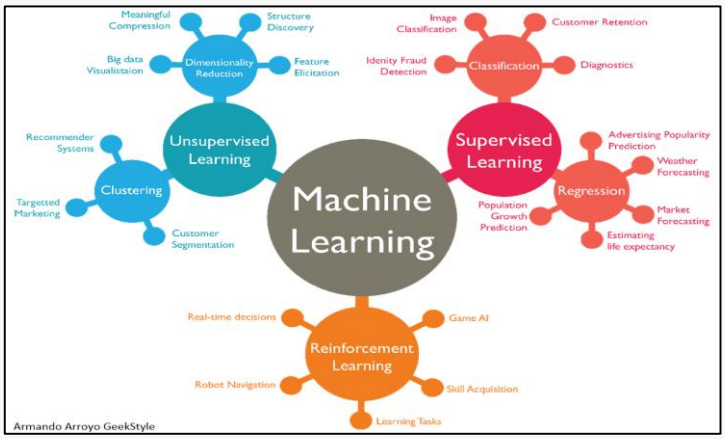
Arpit patrange : 3rd April 2021

ABSTRACT

Wireless Sensor Networks (WSN) has gained attention because it has been useful in warning about disasters. Predicting natural disasters like hailstorm, fire, rainfall etc. by WSN are infrequent and stochastic. This is an important topic of research. Detection of those disasters should be fast and accurate as they'll cause damage and destruction at an outsized scale. In this paper, comparison of varied machine learning techniques like SVM, regression, decision trees, neural networks etc. has been done for prediction of forest fires. The projected approach during this paper presents how regression works best for detection of forest fires with high accuracy by dividing the dataset. Fast detection of forest fires is accomplished during this paper by taking less time as compared to other machine learning techniques.

INTRODUCTION

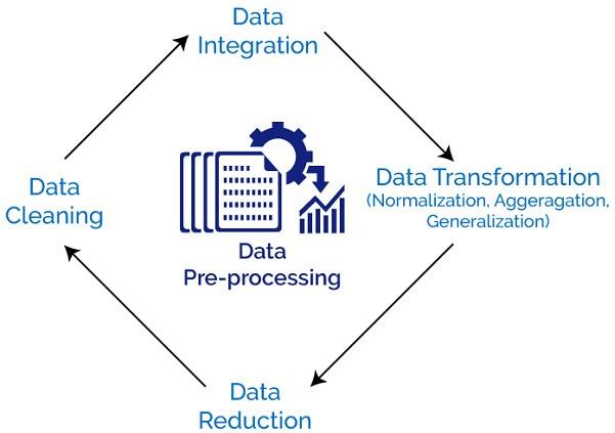
One of the most tremendously occurring disasters in recent times is forest fires. Due to these wildfires, tons of acres of forest area are becoming destroyed. The significant reasons that lead to the occurrence of forest fires are warming due to the increase in the average temperature of the earth, and human negligence. Dynamic Integrated Model of Climate and therefore the Economy (DICE) indicates that the economy will lose about $23 trillion within the next 80 years thanks to the change in climate. In Africa, South America, Southeast Asia, and New Zealand, forest fires occur thanks to human factors like husbandry of animals and agriculture. Nowadays, there are various technologies for fire modelling to predict the spread of fires, like physical models and mathematical models. These models be contingent on data collection during forest fires, simulation, and lab experimentations to specify and predict fire growth in many regions. Recently, simulation tools have been used to predict forest fires, but simulation tools faced some problems like the accuracy of input file and simulation tool execution time. Machine learning may be a sub-branch of AI (AI) to find out computers aspect. Machine learning are often divided into two classes: supervised, unsupervised and reinforcement. In supervised learning, a supervisor is existed to offer insights to the training algorithm on how a choice or an action is bad or good. In supervised learning, the entire the info set is labelled completely. Supervised machine learning algorithms are as rectilinear regression, Support Vector Machine (SVM), Artificial Neural Networks (ANN) and decision trees. In unsupervised algorithm, the info set isn't labelled. This leads that the algorithm must define the labels. The structure of the info set and therefore the relationship between the features are going to be learned by the algorithm. Unsupervised machine learning procedures are as k-means clustering and Self-Organizing Map (SOM). In reinforcement learning, the training algorithm gets punished just in case "> just in case of a wrong action and gets rewarded in case of correct action.



DATA PREPROCESSING

Data preprocessing is a data mining technique which is used to transform a raw data into a useful and efficient format.

In this project data preprocessing is used to filter the raw data and transform it into a useful data.



LITERATURE REVIEW

This dataset is public available for research. The details are described in [Cortez and Morais, 2007].

P. Cortez and A. Morais. A Data Mining Approach to Predict Forest Fires using Meteorological Data. In J. Neves, M. F. Santos and J. Machado Eds., New Trends in Artificial Intelligence, Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence, December, Guimaraes, Portugal, pp. 512-523, 2007. APPIA, ISBN-13 978-989-95618-0-9.

Available at:
http://www.dsi.uminho.pt/~pcortez/fires.pdf

This is a very difficult regression task. It can be used to test regression methods. Also, it could be used to test outlier detection methods, since it is not clear how many outlier are there. Yet, the number of examples of fires with a large burned area is very small.

1. X - x-axis spatial coordinate within the Montesinho park map: 1 to 9

2. Y - y-axis spatial coordinate within the Montesinho park map: 2 to 9

3. month - month of the year: "jan" to "dec"

4. day - day of the week: "mon" to "sun"

5. FFMC - FFMC index from the FWI system: 18.7 to 96.20

6. DMC - DMC index from the FWI system: 1.1 to 291.3

7. DC - DC index from the FWI system: 7.9 to 860.6

8. ISI - ISI index from the FWI system: 0.0 to 56.10

9. temp - temperature in Celsius degrees: 2.2 to 33.30

10. RH - relative humidity in %: 15.0 to 100

11. wind - wind speed in km/h: 0.40 to 9.40

12. rain - outside rain in mm/m2 : 0.0 to 6.4

13. area - the burned area of the forest (in ha): 0.00 to 1090.84

(This output variable is much skewed towards 0.0, thus it may make sense to model with the logarithm transform).

Burned areas in forest fires were predicted using estimation methods because the Multilayer Perceptron (MLP), SVM, Radial Basis Function (RBF) networks and fuzzy logic. The results indicate that MLP gives more accurate results.

Existing, and Dependable Storage for an Incompletely Trusted Environment (FARSIGHT) simulator was used to predict forest fires spread within the Euro-Mediterranean countries. The outputs of FARSIGHT were obtained by two models, a custom fuel model and a standard fuel model. The experimental results showed that theaccuracy of the custom fuel model was better than the quality fuel model. An intelligent system called geometric semantic genetic programming to predict burned areas.

The results obtained using that intelligent system were better than using standard genetic programming. A novel organization called forecast to foresee the spread of forest fires in the future. The forecast may be a system that mixes AI.

METHODOLOGY

Machine learning models play a serious role within the process of evaluation and prediction. Prediction is usually done by using the available variables within the info set. Through the available variables within the info set, machine learning models can make predictions for the future. In this section, linear regression, polynomial regression, decision tree and random forest regression ensemble learning are presented.

Linear Regression:

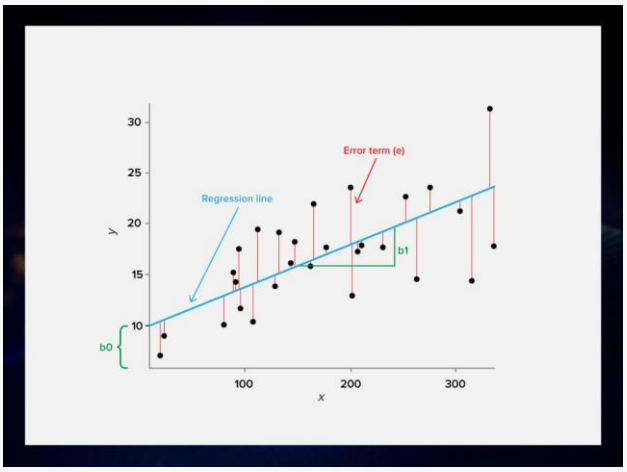
Simple rectilinear regression is beneficial for locating relationship between two continuous variables. One is predictor or experimental variable and other is response or variable. It looks for arithmetic relationship but not deterministic relationship. Relationship between two variables is claimed to be deterministic if one variable are often accurately expressed by the opposite. Statistical relationship isn't accurate in determining relationship between two variables.

The core idea is to get a line that most closely fits the info. The best fit line is that the one that total prediction error (all data points) are as small as possible.

A linear regression line has an equation of the form

$$Y = a + bX$$

Where X is the explanatory variable and Y is the dependent variable. The slope of the line is b, and a is the intercept.



Representation of model

```
#Linear_Regression_Model-->>ForestFire

from sklearn.linear_model import LinearRegression
linear_regressor = LinearRegression()
linear_regressor.fit(x_train,y_train)
```

Polynomial Regression:

Polynomial regression may be a superior case of rectilinear regression where we fit a polynomial equation on the info with a curvilinear relationship between the target variable and therefore the independent variables.

In a curvilinear relationship, the worth of the target variable changes during a non-uniform manner with reference to the predictor (s).

$$Y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_n x^n$$

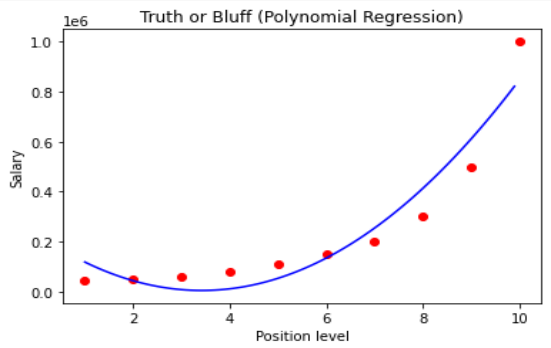
This equation are often wont to represent a linear relationship. But, in polynomial regression, we've a polynomial equation of degree n represented as:

Here: $\theta 0$ is the bias.

$\theta 1, \theta 2, \ldots, \theta n$ are the weights in the equation of the polynomial regression, and n is the degree of the polynomial.

The number of higher-order terms upsurges with the cumulative value of n, and hence the equation becomes more complicated.

Example of polynomial-graph (degree 2):



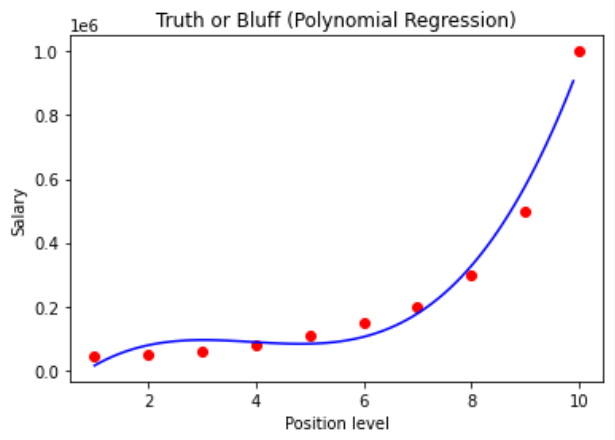
Representation of model for degree 2:

```
#for_degree_2
from sklearn.preprocessing import PolynomialFeatures
polynomial_0 = PolynomialFeatures()
polynomial_x_train_0 = polynomial_0.fit_transform(x_train)
polynomial_x_test_0 = polynomial_0.fit_transform(x_test)

p_linear_regresssor_0 = LinearRegression()
p_linear_regresssor_0.fit(polynomial_x_train_0,y_train)

predict_p_linear_regressor_0 = p_linear_regresssor_0.predict(polynomial_x_test_0)
```

Example of polynomial-graph (degree 3):



Representation of model for degree 3:

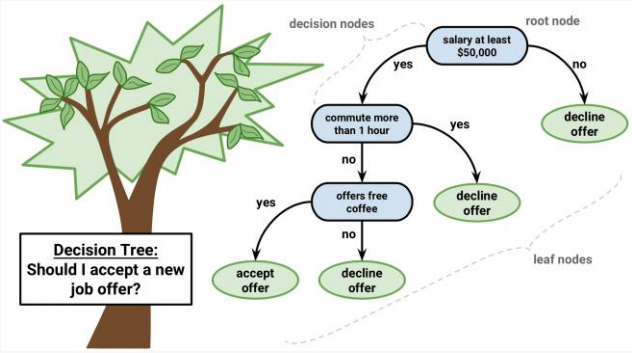
```
#for_degree_3
polynomial_1 = PolynomialFeatures(degree = 3)
polynomial_x_train_1 = polynomial_1.fit_transform(x_train)
polynomial_x_test_1 = polynomial_1.fit_transform(x_test)

p_linear_regresssor_1 = LinearRegression()
p_linear_regresssor_1.fit(polynomial_x_train_1,y_train)

predict_p_linear_regressor_1 = p_linear_regresssor_1.predict(polynomial_x_test_1)
```

Decision Tree Learning:

Decision tree learning is one of the most widely used and practical methods for inductive inference. It is a method for approximating discrete-valued functions, in which the learned function is represented by a decision tree. It is robust to noisy data and capable of learning disjunctive expressions. Decision tree learning algorithms that includes widely used algorithms such as ID3, CART, and C4.5.Learned trees can also be re-represented as sets of if-then rules to improve human readability. Decision tree algorithms transform raw data to rule based decision making trees.



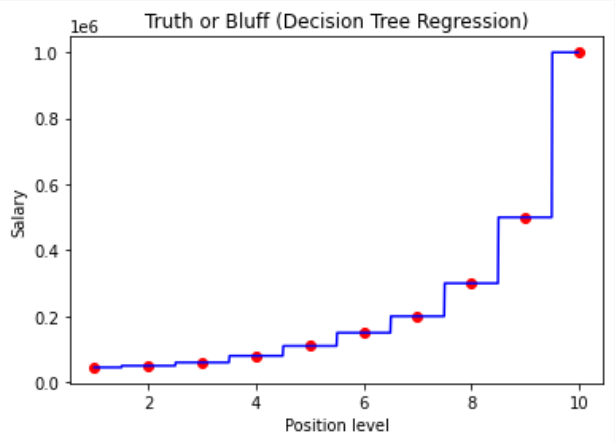
A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values We use standard deviation to calculate the homogeneity of a numerical sample. If the numerical sample is completely homogeneous its standard deviation is zero.

$$SDR(T,X) = S(T) - S(T,X)$$

SDR(Hours , Outlook) = **S**(Hours) – **S**(Hours, Outlook)

= 9.32 – 7.66 = 1.66

Example of decision tree-graph:



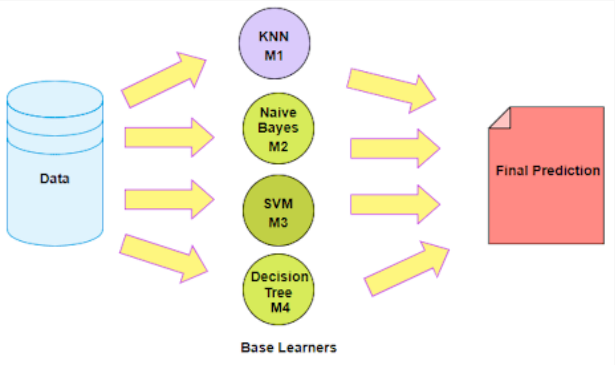
Representation of model for decision tree:

```
#Decision_Tree_Regression-->forestfire

from sklearn.tree import DecisionTreeRegressor
decision_tree_regressor = DecisionTreeRegressor(random_state=0)
decision_tree_regressor.fit(x_train,y_train)
```

Random Forest Regression:

An Ensemble method may be a practice that mixes the predictions from multiple mechanism learning algorithms together to form more accurate predictions than a person model. A model comprised of the many models is named an Ensemble model.



Random forest is a Supervised Learning algorithm which uses ensemble learning method for classification and regression.

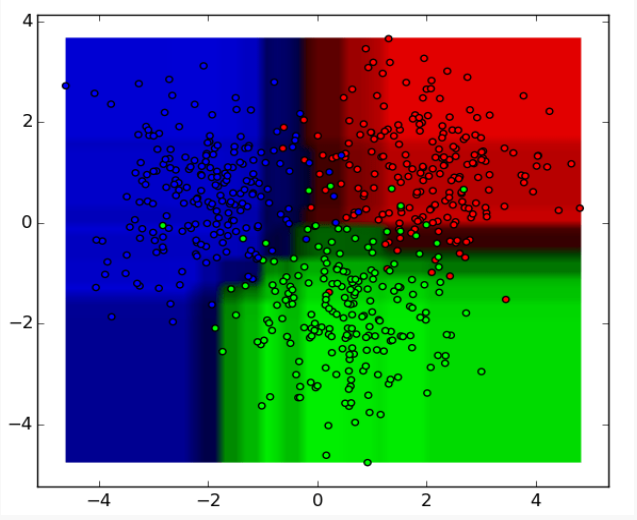
Random forest is a bagging technique and not a boosting technique. The trees in random forests are run in similar. There is no communication between these trees while building the trees.

It operates by constructing a mess of decision trees at training time and outputting the category that's the mode of the classes or mean prediction of the individual trees.

A random forest is a meta-estimator which aggregates many decision trees, with some helpful modifications:

The number of topographies that can be split on at each node is limited to some percentage of the total (which is known as the hyper parameter). This ensures that the ensemble model doesn't rely too heavily on a person feature, and makes use of all potentially predictive features.

Each tree draws a random example from the first data set when generating its splits, adding an extra element of randomness that forestalls over fitting.



Representation of model for Random Forest Regression:

```
#Random_Forest_Regression-->forestfire
from sklearn.ensemble import RandomForestRegressor
random_forest_regressor = RandomForestRegressor(n_estimators=10,random_state=0
random_forest_regressor.fit(x_train,y_train)
```

ERROR DETECTION

In most of the time we use two major types of error to find the accuracy of the models named as,

Mean squared error:

Means Squared Error (MSE), which is simply the average value of the SSE cost that we minimized to fit the linear regression model. The MSE is useful to compare different regression models or for tuning the parameters via greed search and cross-validation, as its normalizes the SSE by the sample size.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

R2 Score:

Coefficient of determination (R2), which can be understood as standardized version of the MSE, for better interpretability of the model's performance. Or in other words, R2 is the fraction of response variance that is captured by the model. The R2 value defined as

$$R^2 = 1 - \frac{SSE}{SST}$$

Here, SSE is the sum of squared errors and SST is the total sum of squares:

$$SST = \sum_{i=1}^{n}\left(y^{(i)} - \mu_y\right)^2$$

In other words, SST is simply the variance of the response.

R2 is indeed just a rescaled version of MSE:

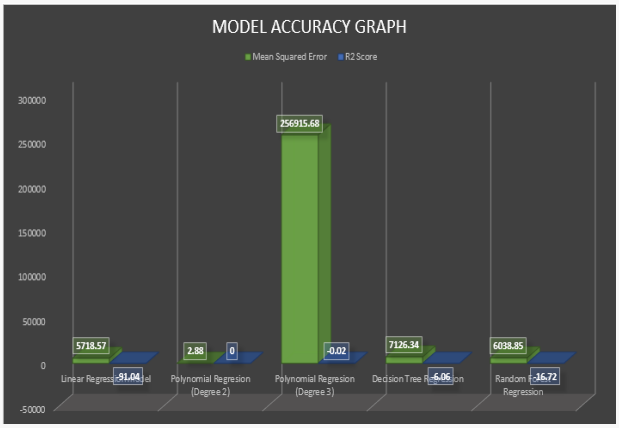
$$R^2 = 1 - \frac{SSE}{SST}$$

$$1 - \frac{\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2}{\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \mu_y\right)^2}$$

$$1 - \frac{MSE}{Var(y)}$$

For the training dataset, the R2 is bounded between 0 and 1, but it can become negative for the test set. If R2 =1, the model fits the data perfectly with a corresponding MSE = 0.

CONCLUSION



From the above predicted graph and the explanation that mentioned in the methodology It is clearly concluded that the 2nd graph which named as polynomial regression having degree 2 is giving the best prediction from the other machine learning models which are implemented on the dataset. For more clarification of the result please refer the following images.

Error calculation for linear regression

```
PREDICTIONS ON THE BASIS OF LINEAR REGRESSION MODEL

MEAN_SQUARE_ERROR
5718.576327781552

MEAN_ABSOLUTE_ERROR
22.280706456811853

R2_SCORE
-91.40251945988307
```

Error calculation for polynomial regression (degree 2) – best prediction

```
PREDICTIONS ON THE BASIS OF POLYNOMIAL REGRESSION MODEL(best
model)

FOR_DEGREE_2(best prediction)

MEAN_SQUARE_ERROR
2.886658591849513e+23

MEAN_ABSOLUTE_ERROR
81385316684.63167

R2_SCORE
-0.006510810649086096
```

Error calculation for polynomial regression (degree 3)

```
FOR_DEGREE_3

MEAN_SQUARE_ERROR
256915.68537154605

MEAN_ABSOLUTE_ERROR
213.37739666462963

R2_SCORE
-0.0232699103665166
```

Error calculation for Decision Tree Regression

```
PREDICTIONS ON THE BASIS OF DECISION TREE REGRESSION MODEL

MEAN_SQUARE_ERROR
7126.348447715383

MEAN_ABSOLUTE_ERROR
26.77375

R2_SCORE
-6.065900014549461
```

Error calculation for Random Forest Regression

```
PREDICTIONS ON THE BASIS OF RANDOM FOREST REGRESSION MODEL

MEAN_SQUARE_ERROR
6038.859069627307

MEAN_ABSOLUTE_ERROR
24.689098076923077

R2_SCORE
-16.729156886331758
```

CODE : https://github.com/Arpit-Patrange/forest_fires_prediction_using_regression.git

REFRENCES

1. Sahin, Y.G., Animals as mobile biological sensors for forest fire detection. Sensors, 2007. 7 (12): p. 3084–3099.

2. Luis, J.A., J.A.G. Galán, and J.A. Espigado, Low power wireless smoke alarm system in home fires. Sensors, 2015. 15(8): p. 20717–20729. 1116 V. Chowdary and M.K. Gupta

3. Bousack, H., et al., Towards Improved Airborne Fire Detection Systems Using Beetle Inspired Infrared Detection and Fire Searching Strategies. Micromachines, 2015. 6(6): p. 718–746.

4. Sardouk, A., et al., Crisis management using MAS-based wireless sensor networks. Computer Networks, 2013. 57(1): p. 29–45.

5. Clemente, A.D.S.B., J.R. Martínez-de Dios, and A.O. Baturone, A wsn-based tool for urban and industrial fire-fighting. Sensors, 2012. 12(11): p. 15009–15035.

6. Ulucinar, A.R., I. Korpeoglu, and A.E. Cetin, A Wi-Fi cluster based wireless sensor network application and deployment for wildfire detection. International Journal of Distributed Sensor Networks, 2014.

7. Molina-Pico, A., et al., Forest Monitoring and Wildland Early Fire Detection by a Hierarchical Wireless Sensor Network. Journal of Sensors, 2016. 2016.

8. Hartung, C., et al. FireWxNet: A multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments. in Proceedings of the 4th international conference on Mobile systems, applications and services. 2006. ACM.

9. Yan, X., et al., Real-Time Identification of Smoldering and Flaming Combustion Phases in Forest Using a Wireless Sensor Network-Based Multi-Sensor System and Artificial Neural Network. Sensors, 2016. 16(8): p. 1228.

10. Premal, C.E. and S. Vinsley. Image processing based forest fire detection using YCbCr colour model. in Circuit, Power and Computing Technologies (ICCPCT), 2014 International Conference on. 2014. IEEE.

11. Angayarkkani, K. and N. Radhakrishnan. An effective technique to detect forest fire region through ANFIS with spatial data. in Electronics Computer Technology (ICECT), 2011 3rd International Conference on. 2011. IEEE.

12. Poobalan, K. and S.-C. Liew. Fire detection algorithm using image processing techniques. in Proceedings of the 3rd International Conference on Artificial Intelligence and Computer Science (AICS2015). 2015.

13. Lloret, J., et al., A wireless sensor network deployment for rural and forest fire detection and verification. Sensors, 2009. 9(11): p. 8722–8747.

14. Millan-Garcia, L., et al., An early fire detection algorithm using IP cameras. Sensors, 2012. 12(5): p. 5670–5686.

15. Tsiourlis, G., S. Andreadakis, and P. Konstantinidis, SITHON: a wireless network of in situ optical cameras applied to the early detection-notification-monitoring of forest fires. Sensors, 2009. 9(6): p. 4465–4482.

16. Yuan, C., Y. Zhang, and Z. Liu, A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. Canadian journal of forest research, 2015. 45(7): p. 783–792.

17. Cruz, H., et al., Efficient forest fire detection index for application in unmanned aerial systems (UASs). Sensors, 2016. 16(6): p. 893. 18. Min, B.-C., et al. Using directional antennas as sensors to assist fire-fighting robots in large scale fires. in Sensors Applications Symposium (SAS), 2014 IEEE. 2014. IEEE. 19. Sahin, Y.G. and T. Ince, Early forest fire detection using radio-acoustic sounding system. Sensors, 2009. 9(3): p. 1485–1498.