Data Mining Techniques Applied to Wireless Sensor Networks for Early Forest Fire Detection

Massinissa Saoudi
Lab-STICC UMR CNRS 6285
Université de Bretagne
Occidentale, Brest, France
University College Dublin,
Ireland
massinissa.saoudi@univ-brest.fr

Ahcène Bounceur
Lab-STICC UMR CNRS 6285
Université de Bretagne
Occidentale, Brest, France
20, Avenue Victor Le Gorgeu,
29238, Brest, France
ahcene.bounceur@univ-brest.fr

Reinhardt Euler
Lab-STICC UMR CNRS 6285
Université de Bretagne
Occidentale, Brest, France
20, Avenue Victor Le Gorgeu,
29238, Brest, France
reinhardt.euler@univ-brest.fr

Tahar Kechadi
UCD,University College Dublin
Belfield, Dublin 4, Ireland
tahar.kechadi@ucd.ie

ABSTRACT

Nowadays, forest fires are a serious threat to the environment and human life. The monitoring system for forest fires should be able to make a real-time monitoring of the target region and the early detection of fire threats. In this paper, we present a new approach for forest fire detection based on the integration of Data Mining techniques into sensor nodes. The idea is to use a clustered WSN where each sensor node will individually decide on detecting fire using a classifier of Data Mining techniques. When a fire is detected, the corresponding node will send an alert through its cluster-head which will pass through gateways and other cluster-heads until it will reach the sink in order to inform the firefighters. We use the CupCarbon simulator to validate and evaluate our proposed approach. Through extensive simulation experiments, we show that our approach can provide a fast reaction to forest fires while consuming energy efficiently. [1]

Keywords

Fire detection; Wireless sensor networks; Data Mining; Intelligent Decision Making.

1. INTRODUCTION

Forest fires are one of the main causes of environmental degradation nowadays. Over 8388 forest fires have been counted in the Mediterranean region in France between January 2010 and October 2015 [1]. Forest fires can be deadly

---

threats to the environment and to human life. In some of these fires, large areas of forests of more than 21 954.61 hectares have been destroyed [1] and many people or animals have died. Therefore, the monitoring and early detection of forest fires is very important for efficient prevention and protection against the damage caused by them.

Several technologies and systems have been proposed to detect fire, e.g., systems employing charge-coupled device cameras and infrared detectors, satellite systems and images [2, 3] and wireless sensor networks [4, 5, 6].

As a good solution, wireless sensor networks (WSNs) are an emerging technology that can be used for forest fire detection and other applications (home and environmental monitoring [7]). A wireless sensor network is usually composed of a few sinks and a large quantity of small sensor nodes, which are able to sense, process and communicate data [8]. To detect fire, a sensor node can be deployed in the forest and collect data such as temperature or humidity, and deliver these data to the base station (sink node) where they can be processed and analyzed automatically for the detection of fire without requiring manual operations performed by humans. This application produces a big volume of geographically distributed and heterogeneous data. In addition, the design and deployment of sensor networks has raised many challenges related to their large size (up to thousands of sensor nodes), random deployment, lossy communicating environment, limited battery power, limited processing unit, small memory, and high failure rate. Energy consumption is a particularly limiting factor for the life-time of a node in a WSN. Therefore, processing and communication should be minimized and there is a permanent need to balance the power consumption on all nodes, on the basis of their residual energy. It is also necessary to integrate Data Mining techniques into the sensor nodes in order to solve the above limitations, e.g., to reduce the size and to improve the quality of the collected data in an intelligent way.

Data Mining (DM) is a process of extracting hidden patterns from large data sets and a critical component of the knowledge discovery process [9]. This process needs to coordinate predictive analysis and decision support systems in real-time. The detection of forest fire in real-time is our

challenge.

To this aim, we propose a new approach based on Data Mining (DM) techniques and a clustered architecture. Each sensor node will individually decide on fire detection using a classifier of DM technique. When a fire is detected, the corresponding node sends an alert through its cluster-head which will pass through gateways and other cluster-heads until it reaches the sink in order to inform the firefighters.

The remainder of this paper is organized as follows. In Section 2, related work is presented. Our approach will be discussed in Section 3. Then, Section 4 exhibits and discusses the obtained results. Finally, Section 5 concludes the paper.

2. RELATED WORK

During the last years, various studies have been performed regarding the detection of forest fires using WSNs.

The authors of [3] propose a combination of a Wireless Local Area Network (WLAN) and sensor-node technology for fire detection. The system is comprised of multi-sensor nodes and IP cameras in a wireless mesh network to detect and verify a fire in rural and forest areas of Spain. When a fire is detected by a wireless multi-sensor node, an alert generated by the node is propagated to a central server on which a software application runs for selecting the closest wireless camera(s). Then real-time images from the zone are streamed to the sink. In this study, the sensor nodes are deployed with a large distance between the nodes and the data from sensors and cameras are collected and processed at a base station. However, our proposed system considers a clustered deployment strategy where the distances between neighboring sensor nodes are rather short. In this way, our goal is to detect forest fire more quickly and to send the related information to a base station as rapidly as possible.

The authors of [6] combine a wireless sensor network with an artificial neural network (ANN) for forest fire detection. The system they propose collects data from a region via a WSN by sensors, such as temperature, light, and smoke. All readings are transmitted (after being transformed into information and then into knowledge) to an already trained ANN, at the central processing unit (base station). The ANN running at the base station uses the received information to test whether some part of it belongs to the fire class for fire detection.

In [9], the authors present a comparative analysis of various Data Mining techniques on WSN fire detection data using the WEKA tool. The goal was to see which of them has the best classification accuracy of fuzzy logic generated data and which is the most appropriate for a particular application of fire detection. In our system, we use real sensing data and we simulate under conditions close to the reality.

In [4], the authors present a framework for forest fire detection which includes a clustered network architecture for the deployment of sensor nodes, as well as an interaction protocol of intra and inter-clusters. They develop a simulator to perform simulation tests in order to examine the proposed system protocols and components. In the end, their system manages to provide effective and efficient operation that consumes less energy without disturbing the rapid reaction capability. In this study, the fire detection is done on a cluster-head level. In our approach, we suppose that every node in the WSN contains all the required functions. In this way, communication overhead between neighboring nodes is avoided and each sensor node can detect fire locally by itself. This allows to reduce the energy consumption and to improve the performance of the WSN.

3. PROPOSED APPROACH

In this section, we describe our proposition for fire detection based on WSN and Data Mining techniques. We first introduce some assumptions and primitives and then identify the important design features that a wireless sensor network should possess, as well as the Data Mining techniques used in order to be able to successfully monitor a forest and to detect fires.

- *Early Detection and Accurate Localization:* Early detection and high accuracy of the localization of forest fire are necessary for a rapid intervention of firefighting personnel at the correct place.

- *Energy efficiency:* The deployment of a WSN for fire detection should consume energy very efficiently because the replacement of batteries may be too costly, impractical or even not possible. The energy consumption should also be balanced among nodes in order to maximize the life-time of the WSN.

3.1 Assumptions and primitives

In our proposed approach, we consider a WSN with one base station and hundreds of multi-sensor nodes. More precisely, there are n sensor nodes in the WSN, each denoted by s_i, $1 \leqslant i \leqslant n$, and identified by a unique identifier id_i in order to distinguish them. We assume that any two sensor nodes can directly exchange messages if their Euclidean distance is not greater than their communication range R_c. Hence, the set of neighbor nodes $N(s_i)$ of a given node s_i can be defined as follows:

$N(s_i) = \{s_j : dist(s_i, s_j) <= R_c, \ j = 1, ..., n \ and \ j \neq i\}$.

Fires can differ in size and shape. These constraints can influence the possibility to detect fire. Therefore, it is necessary to find the optimum size of target area coverage by a single node. For simplicity, we assume that a planar area can be covered by a sensor node if their Euclidean distance is not greater than the sensing range R_s. A forest fire f can be seen as a function $f(x_1, x_2, .., x_m)$, where the variables x_i represent m attributes such as temperature, humidity,... each of them being sensed by the sensor unit in the node. We also assume that e_i is the current remaining energy level of the node s_i and r_i^h is the risk level of season h, $1 \leqslant h \leqslant 4$ of the node s_i. Each cluster-head has a fire threshold $FT\{low, medium, high\}$ and a_i is the number of received alerts from each of its member nodes for the specified period.

3.2 Proposed Approach

In this part we describe the proposed architecture for forest fire detection. As shown in Figure 1, a large number of sensor nodes are manually deployed in the forest. These sensor nodes are organized as clusters so that each node has a corresponding cluster-head. Each sensor node can measure environmental temperature, relative humidity, smoke and light. Consequently, the communication overhead between neighboring nodes is avoided and each sensor node can detect fire locally by itself. In order to precisely locate the source of the fire and to reduce the energy consumption

we assume that the base station knows the precise position of the sensor nodes at the beginning together with the corresponding id_i. Each sensor node predicts the fire using a Data Mining technique and sends the alert containing its id_i, to the corresponding cluster-head. The cluster-head calculates the danger rate and sends the id_i and damage rate to the sink via the gateway node. The sink detects the location of fire using the stored coordinate that corresponds to the received id_i for possible actions, such as alerting local residents or personal fire fighting, and stores the alert in the server for the sake of statistical analysis.
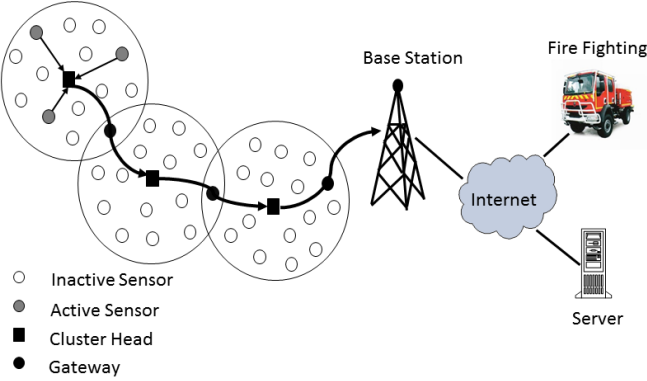

Figure 1: The proposed architecture for forest fire detection.

The proposed approach can be divided into four main phases : a clustered network architecture, route discovery to the sink, fire detection and routing alerts to the sink. Next, we describe the design of each of these phases in more detail.

3.2.1 Clustering

An efficient functioning of a WSN depends on the topology of the network. An architecture based on a clustered topology provides important advantages for forest fire detection. Hence, it is possible to benefit from rapid detection of fire danger, to maximize the life-time of the network, to achieve connectivity and fault-tolerance.

In order to ensure the maximum life-time of a WSN, it is necessary to perform a good energy management in order to cope with depletion of sensor nodes. The objective of connectivity is to guarantee that the most important nodes of the network can communicate with other nodes that are located in their clusters. We also grant particular attention to low computational complexity and high accuracy. These properties are achieved by Data Mining techniques that efficiently detect the fire with as minimal computation as possible.

We have chosen a distributed clustering algorithm [10] which can help us to route data within the WSN and to achieve the above aims. Our algorithm is based only on neighborhood information which is preferable for WSNs as illustrated as follows :

1. Each node s_i broadcasts its information to its neighbors $N(s_i)$.

2. Each node makes the decision according to its local information of the topology to be cluster-head or not.

3. The node selected as a cluster-head broadcasts its status to its neighbors and invites them to join his cluster.

4. If the node receives at least two messages to join two different clusters then it declares itself as a gateway, otherwise it declares itself as a member node.

The selection of cluster-head (CH) is based on weight (the residual energy e_i and a few parameters such as node degree $|N(s_i)|$). The node having the highest weight within this neighborhood is declared as cluster-head. The gateway nodes in the cluster are used to relay data among cluster-heads. The member nodes just treat the fire detection and sent alerts to the corresponding cluster-head.

It is necessary to re-select a new cluster-head among nodes in order not to overload a few nodes with respect to others. There are several studies for cluster-head rotation in [10]. The best way is to use the remaining battery for triggering the clustering algorithm at local regions. When the battery of the cluster-head is below a specified threshold then it broadcasts a message to its neighbors to select a new cluster-head among them.

3.2.2 Routing

There are several routing algorithms in the literature [11]. In our work, we adapt an algorithm based on the cluster network to maximize the life-time, to provide a best performance of the network and allow to route the alert from the node to the sink as rapidly as possible.

After applying the clustering algorithm, each node is declared as a cluster-head or as a gateway including a routing table. At the beginning, the routing table is empty. When the sink propagates the discovery route message which contains its id_i, then the concerned gateways will receive the message and save the identifier of the sink in their routing table. Each gateway node of the sink forwards the discovery route message which contains their identifiers to the next cluster-heads except the sink. When the cluster-heads receive the discovery route message, they save the gateway identifier in the routing table in chronological order. In the same way, each cluster-head forwards the discovery message to the next gateway with the exception of the previous. As soon as all cluster-heads and gateways have received a discovery message they are ready to route the message to the sink. With this technique, the cluster-heads and gateways can use multiple paths to route messages to the sink in the network. The multi-path communications are aimed to improve the reliability, fault-tolerance and performance of the network. For that, the first recorded node is established as the active communication routing while the other nodes are stored for future need, e.g., when the current active node is broken or fails. On the other hand, it is possible to use the other nodes to route data.

3.2.3 Fire Detection

Our work is based on the measurement of real data from sensors (temperature, humidity, light and smoke) and a prediction of fire using classification techniques of Data Mining at the member node level, discarding normal values and transmitting only abnormal values to the cluster-head. This process reduces the number of exchange messages, removes redundancy, improves the system speed and decreases the potential network traffic, extends network life-time and makes early fire detection possible. Also observe, that the rate of

sensing data varies according to year seasons: The sensing rate is high in summer, average in spring and autumn and low in winter. In order to reduce sensing energy consumption, we use an intelligent method which is based on the risk level of the node r_i^h. The node computes its r_i^h for each season h according to the number of fires detected in the season of the previous year. According to [1], in summer, between June 21, 2015 and August 21, 2015, there have been 956 forest fires with 743 fires between 7 am and 9 pm and 213 fires between 9 pm and 7 am. In this case, the rate for sensing data p_i for the next summer is computed as follows :

$$p_i = r_i^h/t \qquad (1)$$

where t is the number of hours in a day when the fire is detected. In our example, 743 fires are detected in the 92 days of summer in the period of 14 hours in one day. The rate for the sensing of node i in this period is : $p_i = (743/92)/14 \simeq 0.58$ fire/h or 1 fire in 1 hour and 45 minutes. Therefore, the sensor node declared as member turns periodically on and off its radio and its multiple sensors according to p_i. On the other hand, the node can sense in the determinate period of 20 minutes every $1.25h$.

In this paper, we use Data Mining to process the sensing data in the sensor node taking into account the limited computing and storage capabilities. We are interested in techniques for finding and describing structural patterns in data as a tool for helping to explain the data and make prediction from them. Classification is one of the popular Data Mining techniques that consists in predicting correctly the probability of a new instance to belong to the predefined class using the set of attributes describing this instance. There are many classification/predictive methods and in this paper, we will focus on the Naïve Bayes Classifier method.

Naïve Bayes Classifier.
This method uses Bayesian statistics and Bayes theorem to find the probability of each instance to belong to a specific class. The training data contain attributes x_i and are split into two classes C_k (Fire, Non), $1 \leqslant k \leqslant 2$. The learning of Gaussian naïve Bayes algorithm relies on the computation of the mean μ_k and the variance σ_k^2 of each attribute x_i in each class C_k. To find the probability of a new sensing instance $I(x_1, x_2, ..., x_m)$ to belong to a specific class C_k, the following formula 2 is applied :

$$P(C_k|x_1, x_2, ..., x_m) = \frac{P(C_k) \prod_{i=1}^{m} P(x_i/C_k)}{evidence} \qquad (2)$$

where

$$evidence = \prod_{i=1}^{m} P(x_i/Fire) + \prod_{i=1}^{m} P(x_i/Non) \qquad (3)$$

and

$$P(x_i/C_k) = \frac{1}{\sqrt{2\pi\sigma_{C_k}^2}} e^{-\frac{(x_i - \mu_{C_k})^2}{2\sigma_{C_k}^2}} \qquad (4)$$

Therefore, the fire is detected if the probability of the $Fire$ class $P(Fire|x_1, x_2, ..., x_m)$ is greater than the probability of the $NonFire$ class $P(Non|x_1, x_2, ..., x_m)$. In case, where these two probabilities are equal, we cannot distinguish the presence or absence of fire. The solution is to launch another classifier in order to detect the fire.

In our work, the fire detection can be categorized into two phases, offline and online as shown in Figure 2. The offline process produces predefined patterns (the model) from the forest environment for the two cases that fire is present or not, using (Naïve Bayes) classification technique and learning from historical data. The model obtained from learning needs to be known before the detection. This phase cannot operate online because wireless sensor nodes have resource limitations: energy, memory and computation. The model as the output from this process is stored in member sensor nodes. The second phase (online process) consists in finding the correspondence between the predefined model from previous processes and sensor reading instances. This process provides a fast detection and reduces the response time. The output from this process is a possibility to detect fire or not.
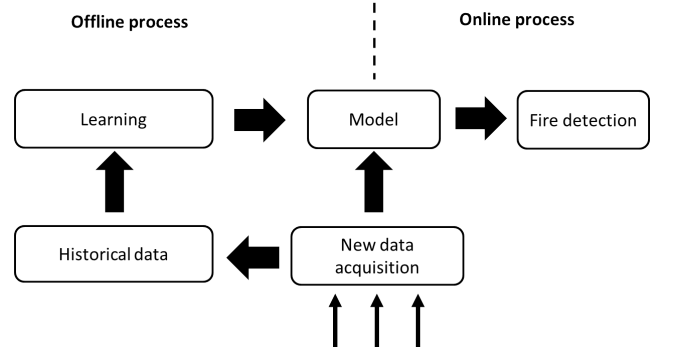

Figure 2: The phase of fire detection.

When the fire is detected by a member node, an alert message is sent to the respective cluster-head. This will reduce the processing cost of all data by the cluster-head and also reduce the communication between the member nodes and their cluster-head. Therefore, the energy consumption is reduced. In addition, the cluster-heads can apply smart scheduling and adaptive transmissions to reduce the overhead on most sensor nodes near the sink. When cluster-heads receive an alert from their members, they compute the number of the received alerts a_i from each of their members and they use a fire threshold FT to determine the current risk level of fire. Then a cluster-head will send an alert to the sink via its gateway using its routing table.

The risk level of fire is determined by comparing a_i with FT. The cluster-head sends an alert to the sink containing the identifier of the corresponding node together with the risk level of fire (low, medium, high). Note, that a_i is re-initialized whenever the member nodes are in sleeping mode. This way, the energy consumption is further reduced.

4. SIMULATION AND RESULTS

To evaluate our proposed approach, we have implemented and performed extensive simulation experiments. In this section we first describe our simulator and then present our experimental results and discussions.

4.1 CupCarbon Simulator

CupCarbon [12] is a Smart City and Internet of Things Wireless Sensor Network (SCI-WSN) simulator. Its objective is to design, visualize, debug and validate distributed algorithms for monitoring, environmental data collection, etc. and to create environmental scenarios such as fires, gas, mobiles, and more generally within educational and scientific projects.

CupCarbon offers two simulation environments. The first simulation environment is a multi-agent environment [13], which enables the design of mobility scenarios and the generation of events such as fires and gas as well as the simulation of mobiles such as vehicles and flying objects [14]. The second simulation environment represents a discrete event simulation of wireless sensor networks which takes into account the scenario designed on the basis of the first environment.

Networks can be designed and prototyped by an ergonomic and easy to use interface using the OpenStreetMap (OSM) framework to deploy sensors directly on the map. It includes a script called SenScript [12] which allows to program and to configure each sensor node individually.

CupCarbon offers the possibility to simulate algorithms and scenarios in several steps. The energy consumption can be calculated and displayed as a function of the simulated time. This allows to clarify the structure, feasibility and realistic implementation of a network before its real deployment.

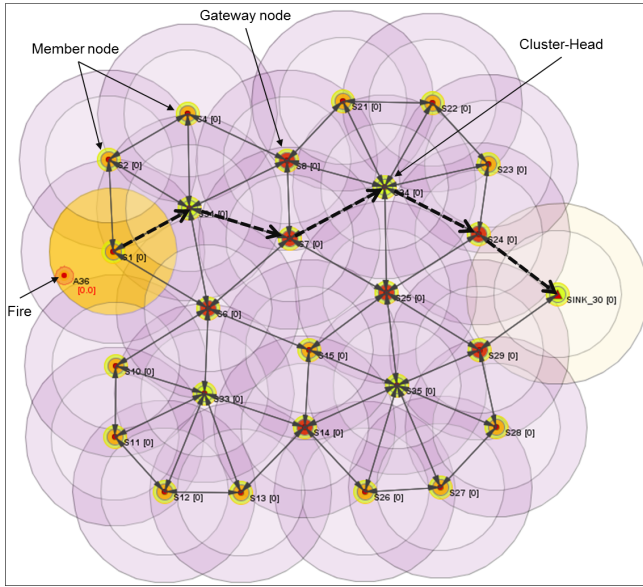Figure 3 shows an example of detecting fire with our approach using the CupCarbon simulator.

Figure 3: CupCarbon Simulator.

When the member node s_1 detects fire, it sends the alert to its cluster-head s_{32} , which itself sends the alert to the gateway s_7, which corresponds to the first node recorded in its routing table. In the same way the gateway s_7 forwards the alert to the next cluster-head s_{34}, from s_{34} the alert is sent to s_{24} and finally, the gateway s_{24} forwards the alert to the sink.

4.2 Results

In order to evaluate the performance of our approach for forest fire detection, the nodes are deployed in the plane representing a forest. The maximum communication range R_c of each node is set to be $100m$. Each sensor node is equipped with battery and multi-sensor devices, which are used to collect data such as temperature, humidity, light and smoke i.e., TMP36, 808H5V5, GL5537 IDR and MQ-135, respectively. The MAC protocol used in the simulation is 802.15.4 which is implemented in the CupCarbon simulator.

To estimate the energy consumption of the proposed approach, we compute the energy consumption in transmission/reception, sensing and computation and we don't take into account the energy consumption in mode of standby, idle and sleep. First, to estimate the transmission/reception energy consumption, we use the energy model of the TelosB sensor node. Its energy consumption is estimated as $59, 2\mu J$ to transmit one byte and as $28, 6\mu J$ to receive one byte [15]. We have used the Super Alkaline AALR6 battery which is a portable energy source with a capacity of 9580 Joules. Second, to estimate sensing energy consumption, we use the following Table [16].

Table 1: Sensing energy of sensors

Type of sensor	Energy consumption (μJ)
Temperature	270
Humidity	72
Light	0.123
Smoke	225

Finally, to estimate the computational energy, we use the energy model of the TelosB sensor node. The energy consumed in computing 1 time clock is $1.2\,nJ$ on the TelosB at $4\,mhz$ [17].

The details of general simulation parameters are depicted in Table 2 :

Table 2: Simulation parameters

No	Parameters	Values
1	Transmitter range	100 m
2	Number of nodes	100
3	Environment size	500 m. x 500 m.
4	MAC protocol type	802.15.4
5	Card type	Arduino UNO
6	Temperature sensor	TMP36
7	Humidity sensor	808H5V5
8	Light sensor	GL5537 IDR
9	Smoke sensor	MQ-135
10	Energy model	TelosB
11	Battery type	Super Alkaline AALR6
12	Battery capacity	9580 J
13	Energy transmission	59,2 μJ/byte
14	Energy reception	28,6 μJ/byte
15	Energy processing	1.2 nJ/1 time clock
16	Data mining technique	Naïve Bayes

Figure 4 shows the comparison between the energy consumption with our proposition, which respects the environment conditions, and simple sensing. The energy consump-

tion with simple sensing remains at similar levels throughout the year because the rate of sensing is fixed to one threshold throughout the year, but with our proposition, the energy consumption changes depending on the season because our approach adapts the rate of sensing according to the history of the number of fires detected in each season of the last year.
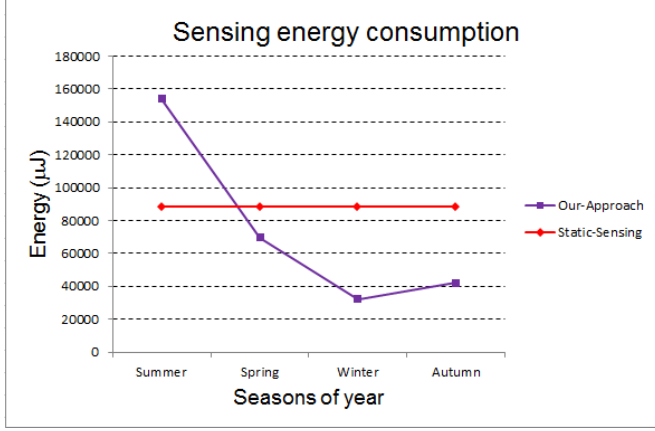

Figure 4: Sensing energy consumption (μJ)

Figure 5 shows the time taken to detect fire and to route an alert from the corresponding node to the sink, in terms of the number of clusters. We performed experiments with up to 10 clusters. We observe, that as the number of clusters in the network increases, our approach provides an improvement in performance.

Figure 5: Average time (milli-sec)

In this simulation, we performed experiments with 100 instances $I(temperature, humidity, smoke, light)$ of data and among which 18 instances represent fire. After the simulation the Naïve Bayes classifies 19 instances as fire, among which the same 18 instances representing fire (true positives) at the beginning and 1 instance is classified as fire but it does not represent fire (false positives). Table 3 shows the results obtained using the Naïve Bayes technique applied to detect a fire in terms of : precision, energy consumption and

response time. The precision P of Naïve Bayes is measured by the following formula :

$$P = \frac{TP}{TP + FP} \qquad (5)$$

where TP and FP are the numbers of true positives and false positives, respectively.

Table 3: Naïve Bayes results

Factors	Results
Energy (nJ)	222,4
Time (S)	0,000972
Precision	0.947

As shown in Table 3, the Naïve Bayes (NB) classifier provides higher classification accuracy, it detects fire rapidly in 0.000972 seconds and it consumes less energy in the computational task in order to detect fire. This energy takes the values of 2,22e-7 joule which is 2,32e-09% from the total battery capacity.

According to the result obtained in Table 3, we noticed that our approach achieves 94.7% of precision and detects 100% of fire but there is one false alert. It is better to have a false alert than not to detect a fire while there is really one. In case of fire detection application, for precautionary measures, we can tolerate the sending of a false alert. In other applications, it is up to the users to choose the criteria of selecting the good classifiers for this application in terms of energy consumption, precision and time of response.

5. CONCLUSION

In this paper, we have proposed a new approach by using wireless sensor networks for forest fire monitoring and detection. Our approach takes into account all characteristics of a WSN that regards low energy capacity, computing limitation, low memory capacity of sensor nodes, and environmental conditions which can affect fire detection and performance of a WSN. Our work is based on measuring and combining real data from different sensors (temperature, humidity, light and smoke) and using the Naïve Bayes (NB) classifier applied to data for fire detection. A node detects fire locally by itself, then it discards normal values and transmits only abnormal values to the sink for fire localization and to inform the firefighters. Applying Data Mining techniques reduces the size of data, deletes redundancy, improves the WSN speed and decreases the network traffic to extend the life-time of the network to guarantee short time of decision and fire detection as early as possible. Our future work will be based on studying and selecting the best classifiers after comparison of various Data Mining techniques applied to detect fire in terms of precision, response time and energy. Furthermore, we intend to find the best algorithm of clustering in order to guarantee an efficient distribution of the sensor nodes to avoid big clusters with a large number of sensor nodes. We also want to secure the message exchange among nodes in order to have a properly protected network.

References

[1] PROMETHEE. Database on mediterranean forest fires in france, http://www.promethee.com, 2015.

[2] Carl Hartung, Richard Han, Carl Seielstad, and Saxon Holbrook. 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*, pages 28–41. ACM, 2006.

[3] Jaime Lloret, Miguel Garcia, Diana Bri, and Sandra Sendra. A wireless sensor network deployment for rural and forest fire detection and verification. *Sensors*, 9(11):8722–8747, 2009.

[4] Yunus Emre Aslan, Ibrahim Korpeoglu, and Özgür Ulusoy. A framework for use of wireless sensor networks in forest fire detection and monitoring. *Computers, Environment and Urban Systems*, 36(6):614–625, 2012.

[5] Mohamed Hefeeda and Majid Bagheri. Forest fire modeling and early detection using wireless sensor networks. *Ad Hoc & Sensor Wireless Networks*, 7(3-4):169–224, 2009.

[6] Hamdy Soliman, Komal Sudan, and Ashish Mishra. A smart forest-fire early detection sensory system: Another approach of utilizing wireless sensor and neural networks. In *Sensors, 2010 IEEE*, pages 1900–1904. IEEE, 2010.

[7] Azhar Mahmood, Ke Shi, Shaheen Khatoon, and Mi Xiao. Data mining techniques for wireless sensor networks: A survey. *International Journal of Distributed Sensor Networks*, 2013, 2013.

[8] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[9] Mirjana Maksimović and Vladimir Vujović. Comparative analysis of data mining techniques applied to wireless sensor network data for fire detection. *JITA-Journal of Information Technology and Applications (Banja Luka)-APEIRON*, 6(2), 2013.

[10] Ossama Younis, Marwan Krunz, and Srinivasan Ramasubramanian. Node clustering in wireless sensor networks: recent developments and deployment challenges. *Network, IEEE*, 20(3):20–25, 2006.

[11] Shio Kumar Singh, MP Singh, DK Singh, et al. Routing protocols in wireless sensor networks–a survey. *International Journal of Computer Science & Engineering Survey (IJCSES) Vol*, 1:63–83, 2010.

[12] ANR PROJECT PERSETPEUR. Cupcarbon simulator, http://www.cupcarbon.com, 2015.

[13] Kamal Mehdi, Massinissa Lounis, Ahcène Bounceur, and Tahar Kechadi. Cupcarbon: A multi-agent and discrete event wireless sensor network design and simulation tool. *In IEEE 7th International Conference on Simulation Tools and Techniques (SIMUTools'14), Lisbon, Portugal*, March 17-19 2014.

[14] M. Lounis, K. Mehdi, and A. Bounceur. A cupcarbon tool for simulating destructive insect movements. *1st IEEE International Conference on Information and Communication Technologies for Disaster Management (ICT-DM'14), Algiers, Algeria*, March 24-25 2014.

[15] Arvinderpal S Wander, Nils Gura, Hans Eberle, Vipul Gupta, and Sheueling Chang Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, pages 324–328. IEEE, 2005.

[16] Mohammad Abdur Razzaque and Simon Dobson. Energy-efficient sensing in wireless sensor networks using compressed sensing. *Sensors*, 14(2):2822–2859, 2014.

[17] Giacomo De Meulenaer, François Gosset, François-Xavier Standaert, and Olivier Pereira. On the energy cost of communication and cryptography in wireless sensor networks. In *Networking and Communications, 2008. WIMOB'08. IEEE International Conference on Wireless and Mobile Computing,*, pages 580–585. IEEE, 2008.