A Bayesian Network Model for Fire Assessment and Prediction

Mehdi Ben Lazreg(✉), Jaziar Radianti, and Ole-Christoffer Granmo

Centre for Integrated Emergency Management,
University of Agder, Grimstad, Norway
{mehdi.b.lazreg,jaziar.radianti,ole.granmo}@uia.no
http://www.ciem.uia.no

Abstract. Smartphones and other wearable computers with modern sensor technologies are becoming more advanced and widespread. This paper proposes exploiting those devices to help the firefighting operation. It introduces a Bayesian network model that infers the state of the fire and predicts its future development based on smartphone sensor data gathered within the fire area. The model provides a prediction accuracy of 84.79 % and an area under the curve of 0.83. This solution had also been tested in the context of a fire drill and proved to help firefighters assess the fire situation and speed up their work.

Keywords: Bayesian network · Indoor fire · Smartphone sensors

1 Introduction

The international association of fire and rescue services reported approximately a million fires in buildings or domestic houses around the world in 2012 alone. These fires unfortunately left 23.7 thousand victims [9]. Thousands of people around the world are affected directly or indirectly by fire. Such facts have previously motivated numerous works in the field of automated fire detection that tried to find some solution to prevent fires and limit the casualties.

During a fire, people tend to leave the building, however, there are potential rescuers going in and trapped victims inside carrying smartphones. In this paper, we propose a model for fire assessment and prediction based on a Bayesian network and smartphone sensors. The number of smartphone user has been growing considerably and is expected to grow even further. Moreover, these smartphones are more and more equipped with advanced sensor technology. The sensor data is gathered from the smartphone located in the fire zone and fed to the Bayesian network. Bayesian networks are capable of handling uncertainty in data which is a common issue when dealing with fire incidents [14]. In addition, they can be adapted to deal with different fire scenarios. To assess the fire status in a specific room, the Bayesian network uses the sensor data along with the estimated state of the fire in neighbouring rooms. The model follows the fire

P. Pardalos et al. (Eds.): MOD 2015, LNCS 9432, pp. 269–279, 2015.
DOI: 10.1007/978-3-319-27926-8_24

development from its ignition until it reaches a fully developed status in addition to forecasting its development.

The topic of automated fire detection and prediction has been extensively studied in the review by Mahdipour et al. [1]. In their review of the subject they showed that most studies focus on detecting fire and reducing the rate of false fire alarm. Various methods have been investigated, including image and video processing, computer vision and statistical analysis to enhance fire detection. These methods focus only on detecting the fire, whereas our method not only detects but follows the development of the fire. Other researches have focused on detecting and predicting fire development by means of wireless sensor networks in context of outdoor and residential area fires. Bahrepour et al. [3] use wireless sensor network (combination of temperature, ionisation, CO and photo-electric sensors) along with machine learning techniques that includes decision tree neural network and naïve Bayes to detect outdoor and indoor fire. Ma [4] used sensor network (temperature, smoke thickness and CO) and neural network fusion algorithm to compute the probability of a fire generated by coal occurs. Nonetheless, those methods are limited only on detecting the fire and they did not take into consideration the sate of the fire in neighbouring rooms as a factor in the fire's propagation. Matellini et al. [5] used Bayesian networks to model the fire development within dwellings from the point of ignition through to extinguishment. Cheng et al. [6] modelled the building as a direct acyclic graph and used Bayesian networks to model fire dynamics in the building and determine the probability of the fire spread from a room to another. However, these methods do not use sensors as a basis for detecting and predicting the fire, but only use the state of the fire in different rooms of the building to deduce its development. Combining Bayesian network with sensor technology with taking into consideration the state of the fire in neighbouring rooms to assess and predict the fire state can be considered as the main contribution of this paper.

The paper is organised as follows: Sect. 2 provides a brief introduction to Bayesian networks. Section 3 presents the fire assessment and prediction model. In Sect. 4, we evaluate the model based on two criteria: its performance for assessing the fire and its usefulness in case of fire. We finally conclude this work in Sect. 5, and reveal the possible future direction.

2 Bayesian Network

A Bayesian Network (BN) represents a set of random variables and their conditional dependencies using a directed acyclic graph (DAG) [7]. In brief, a BN is composed of [14]:

- Directed acyclic graph: contain a set of nodes and directed edges connecting one node to another in a way that starting from a node A there is no sequence of edges that loops back to node A. In a BN, the nodes may represent an observable quantity, latent variable, unknown parameter or hypotheses. The edges represent the causal relationship between two events represented by two nodes: an edge directed from node A to node B implies that the occurrence

of an event represented by node A has a direct impact on the occurrence of another event represented by node B. In a DAG, family terminology is used to describe the relationship between nodes. Hence the parents of A ($pa(A)$) are a set of nodes that have an edge directed to A. The children of A are a set of nodes that are reached by an edge generated from A.
– A set of probabilities: each node in the DAG is assigned a probability distribution if it is a root node or a conditional probability distribution if it is not. Those probabilities express the likelihood that the event symbolised by a certain node accrues.

Bayesian network is based on the fundamental assumption of causal Markov condition [8]. This assumption specifies that each node in the DAG of the BN is conditionally independent of its non-descendent nodes given its parents. To further explain this assumption, let us consider a $DAG = (V, E)$ where V represents the set of nodes in the DAG and E is the set of edges between those nodes. Let $X \in V$ be a node in this DAG. Let $child(X)$ be the set of all the children of X and $pa(X)$ the set of all parents (direct causes of X). The causal Markov condition can be expressed formally as follows

$$\forall X, Y \in V; Y \notin child(X) \Rightarrow P(X|Y, pa(X)) = P(X|pa(X)). \tag{1}$$

From Eq. (1), it can be concluded that for any BN composed of a set of nodes $\{X_1, X_2, ..., X_n\}$ the joint probability is given by

$$\begin{aligned} P(X_1, X_2, ..., X_n) &= P(X_n|X_1, X_2, ..., X_{n-1})P(X_1, X_2, ..., X_{n-1}) \\ &= P(X_1)P(X_2|X_1)...P(X_n|X_1, X_2, ..., X_{n-1}) \\ &= \prod_{k=1}^{n} P(X_k|pa(X_k)). \end{aligned}$$

3 Fire Assessment and Prediction Model Based on Bayesian Network

3.1 Fire Assessment

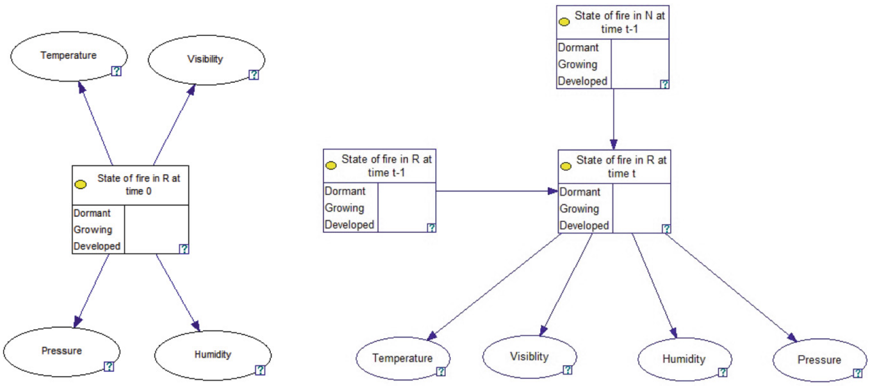
A fire is a dynamic process that evolves through time. Its status at a present time t depends on that at previous time steps [6]. One might then think of using a dynamic Bayesian network (DBN) to capture fire dynamics. However, for each node a DNB keeps other nodes for each time step. Thus, DBN is more complicated and consequently more process consuming [14]. Since we intend to run the application on mobile devices with limited power and battery life, the application of a simple BN is preferable. The Bayesian network presents each room in the building by a node (see Fig. 1(b)). This choice was motivated by the work done by Cheng et al. [6] and Granmo et al. [15] who also used Bayesian network to model indoor hazards.

A fire normally goes through five stages (dormant, growing, developed, decaying and burnt out) [6]. In this research, we only focus on the first three stages for two main reasons. First, this choice simplifies the classification process. Second, it allows us to model the fire at it most dangerous stages (growing and developed). Therefore, the fire in a room R can be dormant, growing or developed. Let S be a set representing the fire state in a room

$$S = \{dormant, growing, developed\}.$$

Smartphones come with a variety of sensors. The most appropriate in case of fire are: temperature, humidity, pressure and light from which the visibility can be deduced. Let O be the set of observed sensor data in R

$$O = \{temperature, humidity, visibility, pressure\}.$$



(a) BN at time 0 (b) BN at time t

Fig. 1. BN for real time fire assessment

At time 0, the fire state in R influences the observed sensor's values recorded in that room. This is modelled by edges going from the node representing the fire status in the BN to the nodes representing each sensor (Fig. 1(a)). Furthermore, to model the dynamic aspect of the fire at a time $t > 0$, we added to the model in Fig. 1(a) a node that represents the status of the fire in a room at previous time step. Moreover, the fire in R depends also on the situation of the fire in neighbouring rooms. Therefore, a node representing the fire state in the neighbouring room at previous time step is added to the BN. Note that the graph in Fig. 1(b) only represented one neighbouring room for simplicity. In reality, R can have multiple neighbours. In that case edges are added between each neighbour and R. If we had used a DNB to model the fire we would end up with 12600 nodes for a 30 min fire simulation for each room in the building instead of the 7 nodes that we have in our model.

Let R_t be the random variable representing the state of the fire in R and N_t that of the neighbouring room at a time t. The BN infers the fire state in R

at time 0 based on the value of temperature, humidity, visibility and pressure collected by the phone sensors placed in R. At a time t, we add the fire state in R at $(t-1)$ and the fire state in the neighbouring room N as a factor in the inference process (Fig. 1). This inference is performed using the joint probability distribution of the random variable in the network expressed as follows

$$P(R_0, O) = P(R_0|O)P(O) \text{ if } t = 0 \tag{2}$$

$$P(R_t, R_{t-1}, N_{t-1}, O) = \frac{P(R_{t-1})P(N_{t-1})P(R_t|R_{t-1}, N_{t-1})P(R_t|O)P(O)}{P(R_t)} \text{ if } t > 0. \tag{3}$$

Finally, at each time step the probability distribution of a node representing the fire state in a room at a time t will be passed to the node that represents its former state as virtual evidence for the next iteration. Unlike normal evidence or soft evidence where the evidence of an observed event is deterministic like in the case of temperature provided by a sensor-, the virtual evidence uses a likelihood ratio to present the confidence toward the observed event. In our case, since the observed state of the fire at a time t is derived from the BN with a certain probability and thus uncertain, it is more appropriate to pass it to the node presenting the previous state as a virtual evidence. The whole process of fire assessment for a single room R is summarised in Algorithm 1.

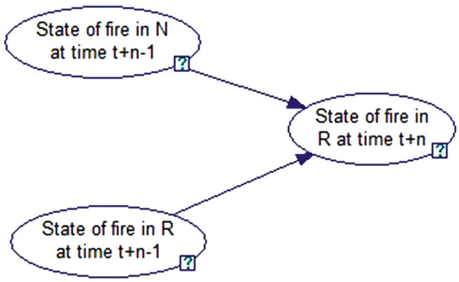


Fig. 2. BN for fire forecasting

3.2 Fire Prediction

In addition, the BN should be able to forecast the state of the fire at a future time $(t+n)$. For this we designed a BN illustrated in Fig. 2. The network in Fig. 2 is similar to the network designed for fire assessment. The only difference is the lack of node representing the sensor data since the sensor data provided by the smartphone sensors is only known at the present time. Thus, at each future time step $(t+n)$ the state of the fire in room R is inferred from the state of the fire in R and its neighbouring rooms in the previous time step $(t+n-1)$. The joint probability distribution would be as follows

$$P(R_{t+n}, R_{t+n-1}, N_{t+n-1}) = P(R_{t+n-1})P(N_{t+n-1})P(R_{t+n}|R_{t+n-1}, N_{t+n-1}). \tag{4}$$

The probability distribution of the fire in R at time $(t+n)$ is then passed to the node representing the probability distribution of the fire in R at time $(t+n-1)$ as virtual

Algorithm 1. Algorithm for fire assessment using BN

Loop
  forall the $o_i \in O$ **do**
    o_i = registered sensor data
  end
  forall the $s_i \in S$ **do**
    If (t=0) infer $P(R_0 = s_i|O))$
    Else infer $P(R_t = s_i|O, R_{t-1}, N_{t-1}))$
  end
  virtual evidence(R_{t-1})= R_t
  $t++$
EndLoop

Algorithm 2. Algorithm for fire prediction using BN

while $t \leq T$ **do**
  forall the $s_i \in S$ **do**
    infer $P(R_t = s_i|R_{t-1}, N_{t-1})$
  end
  virtual evidence (R_{t-1})= R_t
  $t++$
end

evidence. This process is done recursively until a final time T in the future is reached ($t + n = T$). The whole process of fire prediction is summarised in Algorithm 2.

As Algorithms 1 and 2 suggest we need to infer $P(R_t|O, R_{t-1}, N_{t-1}))$ from Eqs. 3 and 4. To do that we need to compute $P(R_t, R_{t-1}, N_{t-1}, O))$ and $P(O)$ with are respectively known as the most probable explanation and the probability of evidence problem. These problem are difficult problems known to be NP-complete and PP-complete problem respectively [14]. Therefore, Eqs. 3 and 4 cannot be solved directly to obtain the probability of each state of the fire due in general to high computational complexity. However, different algorithms have been developed to approximate a solution for those equations. We used one of the fastest and most precise of them: the Estimated Posterior Importance Sampling algorithm for Bayesian Networks (EPIS-BN) [12]. It is based on using loopy belief propagation [13] to compute an approximation of the posterior probability over all nodes of the network. The loopy belief propagation is based on approximating the problem of computing $P(R_t|O, R_{t-1}, N_{t-1}))$ by computing $P(R_t|R_{t-1}, N_{t-1}))$ and $P(O|R_t = s_i)$ where $P(R_t|O, R_{t-1}, N_{t-1})) = \alpha P(R_t|R_{t-1}, N_{t-1}))P(O|R_t)$. Then, it uses importance sampling to refine this approximation. Importance sampling allows to approximate a function by another function called importance function. It is used to approximate $P(O|R_t)$.

4 Test Results and Discussion

4.1 Test Settings

We used the third floor of the University of Agder building as the scenario for our model. The floor contains 5 classrooms, 30 offices, 7 group rooms, 4 computer labs, 2 meeting rooms, 12 corridors and 3 stairways used as escape routes from the fire. The building is an interesting case study since it is large enough to be a challenge for firefighters in the event of a fire: based on our meeting with firefighter they stated that they rarely phase a fire spreading in a building of this amplitude. Each room in the building will be represented by a BN as described in Figs. 1 and 2.

The network described in the previous section is trained and tested using data obtained from several fire simulations runs produced using the fire dynamics simulator (FDS) [11]. The FDS permits the imitation of the geometry of a building and its material properties, the definition of fuel that triggers fire, and the placement of devices such as visibility and temperature sensors in the simulated environment in such a way that fire parameter data can be measured and collected. A user needs first to build a 3D space object called mesh to make a fire simulation, which will be used to construct the 3D building geometry being the target of fire simulations. The user can define the fire cause and starting point and thermal properties of the building material. For our BN experiments, we created a model of the third floor of our university building that follows all the real dimensional size and the detailed rooms and furniture.

For completing the model, the user can place devices and sensors such as sprinklers, smoke detectors, heat flux gauges and produce the quantity outputs, for example, temperature, visibility and so on. The type of sensors placed in each room in this 3D university building is in line with our research goals, i.e. to get information about the temperature, humidity, visibility, and pressure. We run this simulation twice with different starting points of the fire. During those simulations, all defined sensors would register all the data produced in this simulation. The output of the fire parameters produced by FDS comes as a table containing the value of temperature, humidity, visibility and pressure in each room at each second for 30 min as well as the corresponding fire state to those values.

As we have seen in Sect. 2 the BN is composed of a DAG (described in Figs. 1 and 2) and for each node a probability distribution representing the likelihood that an event represented by that node accuses. We trained one BN based on the table produced by the FDS simulations for all the rooms (we ended up with 128 fire examples). This allows to learn those probability distributions. This includes $P(R_t|R_{t-1}, N_{t-1})$, $P(temperature|R_t)$, $P(pressure|R_t)$.... Once learned those probabilities are used in the inference process to solve Eqs. 3 and 4. The building structure is then loaded into the app. It consist of a table with the room its location and neighbours. Copies of the trained BN nodes are then created for each room based on this table.

Further, We simulate two another set of fire scenarios to test the BN. The lines containing the sensor data are retrieved consecutively from the table produce by FDS and fed as evidence to the Bayesian network. The results of this test are the probabilities of each fire state in each room as a function of time. Thus, the BN prediction varies from room to room and from time step to time step.

We have implemented the BN using JSMILE, a Java interface of SMILE (Structural Modelling, Inference, and Learning Engine) [10]. It allows the creation, editing and use of Bayesian network for probabilistic reasoning and decision making under uncertainty.

4.2 Performance Testing

First, we present the results of a test on a specific scenario from the scenarios we used to test our BN. The results for two representative rooms ($R1$ and $R2$) are presented in Figs. 3 and 4. These Figures show the probability of a dormant, growing and developed fire in the two rooms as a function of time as well as the actual state of the fire (black line). Room $R1$ is the neighbouring room toe the fire starting point whereas room $R2$ is located on the opposite side of the building and thus it is the furthest room to the fire starting point. For $R1$ (Fig. 3), the predicted probabilities match the actual state of the fire. The delay of detecting the growing phase of the fire is 3 s. For Room $R2$ (Fig. 4), the BN is not sure about its fire state predictions, especially during the growing phase of the fire. This can be due to a conflict between the sensors' data obtained from the simulation and the fire state in neighbouring rooms: the neighbouring rooms experience a developed fire that should propagate to $R2$ however the sensors' data suggest that the fire is dormant in the mentioned room. In spite of not distinguishing between the growing and developed state of the fire in room R2, the BN was able to at least detect that there is fire in the room (regardless of its state) with a delay of 30 s. In the remaining rooms, the results vary from room to room but they similar to the results presented for room $R1$ and $R2$ with delays to detect the growing and developed state of the fire varying from 3 to 67 s.

Overall the test set, to calculate the overall accuracy of our model, we first take the most probable state as the predicted state. Then, for each room we compute the percentage of the correct classifications. Finally, we average the result over all the room in the building, The overall accuracy of the Bayesian network is then 84.79 %. The model also has an overall area under the curve (AUC) of 0.83. The AUC allows to test the ability of the BN to predict each state of the fire. The AUC can be viewed as the probability that the model gives a higher probability to the fire state that is actually correct. An AUC of 0.83 means that, given an instance from the test set, the model has an 83 % chance of giving a higher probability to the correct state of the fire for that instance. The AUC is a useful metric even if the state of the fire are imbalanced (in our case the growing state is less frequent the two other states). It was extended to evaluate multi class classification problem by Hand et al. [2] who proposed to use the formula in Eq. 5 where c in the number of classes and $A(i,j)$ is the AUC of the binary classification of two classes i and j out of the c classes:

$$AUC = \frac{2}{c(c-1)} \sum_{i<j} A(i,j). \tag{5}$$

4.3 Game Scenario Testing

As mentioned in the previous section, we implemented the BN model in an Android app, and tested this fire development and prediction app in a serious game. We used a fire scenario simulated by FDS to get the sensor data and feed them to the BN. In this game, a group of players (9 persons), acting as firefighters, conducted a search and rescue operation, while another group (13 persons) played victims trapped in the rooms during the indoor fire hazard, with one person acted as the MCU (Medical Care Unit). The game took place at the University of Agder (UiA) and two game sessions of 30 min each had been planned, i.e. one without and one with app support. We focused

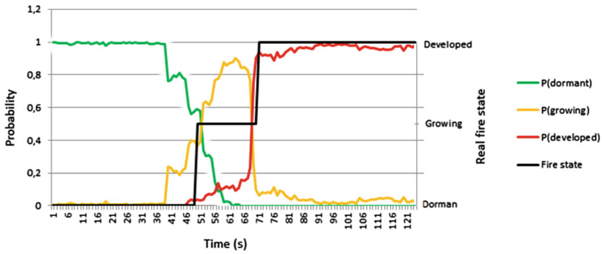


Fig. 3. Fire state probabilities in room R1

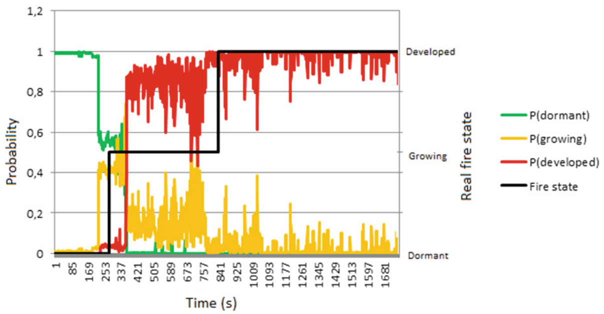


Fig. 4. Fire state probabilities in room R2

on a hypothetical situation where the fire had grown, and several victims were trapped inside.

We hypothesised that the rescue operation with app support (2^{nd} session) would be faster than without app support (1^{st} session). The game goal was to search for victims trapped in the 3rd floor, and rescuing them by moving them all the way to the MCU located by the main entrance of the UiA building. All victims that were saved had to be reported to the CM, who monitored the overall progress of the rescue operation performed by all three rescuer groups. No exact script was given to them as how to act, communicate and interact, except that they were informed on the outline of the roles, tasks, scenarios, prior to the game.

In the session without app, the players should check the room one by one and reported to MCU if the room was clearnobody inside. The communication mode was walkie-talkie software on the smartphone. In that session, each burning room would be marked over time by a fire marker, based on a predefined fire spread. In the scenario with the app, the fire information was available on the smartphone and users could observe the fire spread from room to room by the mean of a heat map. The BN-based fire assessment and prediction app served as a decision support and a basis for rescuers to act while saving the victims. The deployment of the app was conducted in two ways: by sending the app directly to the players to download in advance, and by preparing ten devices with the app installed. The app usage was explained in the briefing, and repeated before the 2^{nd} session was started. In fact, familiarising the players with the app was crucial to the success/failure of the game goal.

The quantitative data collected from this game is about the number of victims saved in the first and second sessions. The game testing shows that the rescue process

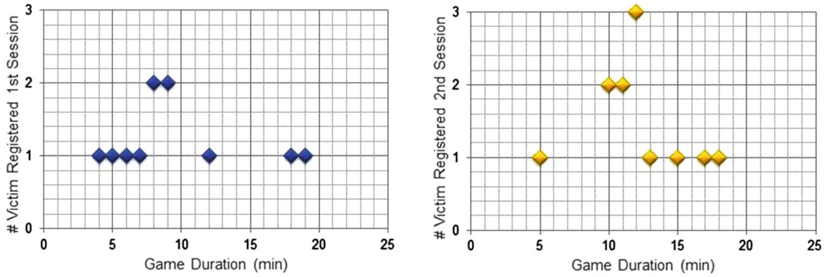


Fig. 5. Number of saved victims as function of time

was faster with the app. Figure 5 shows the time and the number of victims being saved without the fire assessment/prediction app (left) and with the app (right). The horizontal axis indicates the duration of the game in minutes while the vertical axis shows the number of victims saved, as registered by the MCU. The rescue process took 15 min in the first experiment, and 13 min in the second experiment. This time was counted from the moment the first victim was found. In the second session, 11 victims were saved in the last 8 min. On the contrary, the saved victims were spread over a longer time during the first round. There was a longer delay before the rescuers could find the first victim in the second experiment (Fig. 5, right). The reason for this was that the players needed some adjustment to use the app, and some of them experienced technical issues at the beginning of the app use. Further, most of the players relied more on the real-time fire assessment than on the prediction feature while performing their rescue task. This is due to the fact that the rescuers (as they reported in the briefing) had to deal with the real-time assessment of fire situation while trying to save the victim at the same time, and thus could not spend time on "additional" task such as checking where fire would develop in the future.

Hence, we learnt from the game that being able to see how the fire develops over time was useful in a fire situation to decide where the safest place to escape is, but there was a barrier in practice regarding the usage of the prediction feature. The interview with real firefighters who were present during the game indicated that placing the app with the firefighters' leader, who normally does not go inside the building, can relieve the firefighter from that extra task. The leader can then inform the team members about the future fire situation while the firefighters can concentrate on finding the victims. This could be a better design for future testing the usefulness of fire prediction feature.

5 Conclusion

This paper proposes a model that uses smartphone sensors along with Bayesian network to assess fire situation and predict its development. The Bayesian network infers the probability of each state of the fire based on the sensor data collected from smartphones in the fire area and the state of the fire in the previous time step. It provides an overview of the fire situation along with forecasting its development. The test of the model performance shows that the computed probabilities match the actual state of the fire in 84.79 % of the cases and an area under the curve of 0.83. This solution also helps facilitate and speed up the work of firefighters in order the save more lives as revealed from field experience. The future directions of this work would be to use smoke and temperature sensors in the building alongside the smartphones as well as figuring out the optimal number of sensors needed inside the building to still achieve an acceptable prediction accuracy of the fire. We also plan to develop our model to include fire development from one floor to another.

References

1. Elham, M., Chitra, D.: Automatic fire detection based on soft computing techniques: review from 2000 to 2010. Artif. Intell. Rev. **42**(4), 895–934 (2014)
2. David, H., Till, R.: A simple generalisation of the area under the ROC curve for multiple class classification problems. Mach Learn. **45**(2), 171–186 (2001)
3. Bahrepour, M., van der Zwaag, B.J., Meratnia, N., Havinga, P.: Fire data analysis and feature reduction using computational intelligence methods. In: Phillips-Wren, G., Jain, L.C., Nakamatsu, K., Howlett, R.J. (eds.) IDT 2010. SIST, vol. 4, pp. 289–298. Springer, Heidelberg (2010)
4. Ma, X.-M.: Application of data fusion theory in coal gas fire prediction system. In: International Conference on Intelligent Computation Technology and Automation (ICICTA) (2008)
5. Matellini, D.B., Wall, A.D., Jenkinson, I.D., Wang, J., Pritchard, R.: A bayesian network model for fire development and occupant response within dwellings. In: IEEE Conference on Prognostics and System Health Management (PHM) (2012)
6. Cheng, H., Hadjisophocleous, G.V.: The modelling of fire spread in buildings by bayesian network. Fire Saf. J. **44**(6), 901–908 (2009)
7. Stephenson, T.A.: An Introduction to Bayesian Network Theory and Usage. IDIAP researsh institue Martigny, Switzerland (2000)
8. Hausman, D.H., Woodward, J.: Independence Invariance and the Causal Markov Condition. Oxfor University Press, Oxford (1999)
9. Brushlinsky, N.N., Ahrens, M., Skolov, S.V., Wagner, P.: World fire statistics. In: International Association of Fire and Rescue Service (2014)
10. Druzdzel, M.J.: SMILE: structural modeling, inference, and learning engine and GeNIe: a development environment for graphical decision-theoretic models. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence (1999)
11. Kevin, M., Howard, B., Ronald, R.: Fire dynamics simulator technical reference guide. National Institute of Standards and Technology (2007)
12. Yuan, C., Druzdzel, M.J.: An importance sampling algorithm based on evidence pre-propagation. In: The Conference on Uncertainty in Artificial Intelligence (2003)
13. Murphy, K., Weiss, Y., Jordan, M.: Loopy belief propagation for approximate inference: an empirical study. In: Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (1999)
14. Van Harmelen, F., Lifschitz, V., Porter, B.: Handbook of Knowledge Representation, 1st edn. Elsevier, San Diego (2008)
15. Granmo, O.-C., Radianti, J., Goodwin, M., Dugdale, J., Sarshar, P., Glimsdal, S., Gonzalez, J.J.: A spatio-temporal probabilistic model of hazard and crowd dynamics in disasters for evacuation planning. In: Ali, M., Bosse, T., Hindriks, K.V., Hoogendoorn, M., Jonker, C.M., Treur, J. (eds.) IEA/AIE 2013. LNCS, vol. 7906, pp. 63–72. Springer, Heidelberg (2013)