

FOG SUPPORTED WIRELESS SENSOR NETWORKS FOR FOREST FIRE DETECTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Fouad Amira
September 2018

Fog supported Wireless Sensor Networks for Forest Fire Detection

By Fouad Amira

September 2018

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Özgür Ulusoy(Advisor)

İbrahim Körpeoğlu

Ahmet Coşar

Approved for the Graduate School of Engineering and Science:

Ezhan Kardeşan
Director of the Graduate School

ABSTRACT

FOG SUPPORTED WIRELESS SENSOR NETWORKS FOR FOREST FIRE DETECTION

Fouad Amira

M.S. in Computer Engineering

Advisor: Özgür Ulusoy

September 2018

Fog computing is a new paradigm that aims to extend the concept of cloud computing to the edge of the network, providing the end users network with extra storage and processing power. One big contribution of Fog computing is in the context of Wireless Sensor Networks (WSNs). WSNs consist of cheap, battery powered and simple processing devices that make it fall short in handling relatively complex processes. Therefore, applying fog computing to WSNs will fill the gap between the cloud and the network and by that, it will enable computationally extensive operations which were earlier possible only at the cloud side. In our work, we exploit the processing power provided by the Fog to minimize the power consumption of WSNs for forest fire detection through the use of data mining techniques. The Fog layer uses the data generated by the network such as temperature, humidity, rain, etc., to train a model that predicts the probability of forest fires. Next, the Fog layer uses this model to predict the mode of operation of the network based on the current condition of the environment. While a high predicted probability of forest fire results in an increased activity of the WSN, a low fire probability results in a reduced network activity. As a result, our proposed model optimizes the energy consumption within the WSN and improves the detection time of forest fires.

Keywords: Wireless Sensor Networks, forest fire detection, forest fire prediction, Machine Learning, Fog computing.

ÖZET

ORMAN YANGINLARN ALGILAMAK İÇİN SIS DESTEKLI KABLOSUZ SENSÖR AĞLAR

Fouad Amira

Bilgisayar Mühendisliği, Yüksek Lisans

Tez Danışmanı: Özgür Ulusoy

Eylül 2018

Sis bilişim, bulut bilişim kavramını ağıın sınırlarına kadar genişletmeyi amaçlayan, son kullanıcı ağıına ekstra depolama ve işleme gücü sağlayan yeni bir kavramdır. Sis bilişimin büyük bir katkısı Kablosuz Sensör Ağları (KSA'lar) kapsamındadır. KSA, nispeten karmaşık süreçlerin işlenmesinde kısa kalmasını sağlayan ucuz, akülü ve basit işleme cihazlarından oluşur. Bu nedenle, KSA'lara sis bilgisinin uygulanması bulut ile ağ arasındaki boşluğu dolduracak ve bu sayede daha önce sadece bulut tarafında mümkün olan, hesaplama açısından kapsamlı operasyonların yapılmasına olanak sağlayacaktır. Bizim çalışmamızda, veri madenciliği teknikleri kullanılarak orman yangınlarının tespiti için KSA'ların güç tüketimini en aza indirmek amacıyla sis tarafından sağlanan işlem gücünü kullanıyoruz. Sis tabakası, orman yangınlarının olasılığını tahmin eden bir model üretmek için sıcaklık, nem, yağmur vb. gibi ağ tarafından üretilen verileri kullanır. Ardından, ortamın mevcut durumuna bağlı olarak ağıın çalışma şeklini tahmin etmek için sis tabakası bu modeli kullanır. Orman yangınının yüksek olasılıkla tahmin edilmesi, KSA'nın arttırılmış bir aktivitesi ile sonuçlanırken, düşük yangın olasılığı, azaltılmış bir ağ aktivitesi ile sonuçlanmaktadır. Sonuç olarak, önerilen modelimiz KSA içindeki enerji tüketimini optimize eder ve orman yangınlarının tespit süresini azaltır.

Anahtar sözcükler: Kablosuz Sensör Ağları, orman yangını tespiti, orman yangını tahmini, Makine Öğrenmesi, Sis Bilişim.

Acknowledgement

First and foremost, I would like to thank my supervisor Prof. Dr. Özgür Ulusoy for his wise guidance, patience and his suggestions regarding this work. It has been an honour for me to work with him.

I would like also to thank the jury members: Prof. Dr. İbrahim Körpeoğlu and Prof. Dr. Ahmet Coşar for their invaluable feedback during the review of my thesis.

Finally, I would like to thank my family and friends for their help and support.

Contents

1	Introduction	1
2	BACKGROUND	4
2.1	Wireless Sensor Networks for environment monitoring	4
2.2	Fog Computing with Wireless Sensor Networks	6
2.3	Data Mining for Environmental Monitoring	7
3	RELATED WORK	10
3.1	Fog Computing and WSNs	10
3.2	WSN for environment monitoring	11
3.3	Data Mining for forest fire prediction	13
4	PROPOSED APPROACH	15
4.1	Hierarchy	16
4.2	Topology and Architecture	16

4.3	Communication protocols	18
4.4	Data Mining	20
4.4.1	Support vector machine (SVM)	21
4.4.2	Artificial neural network (ANN)	22
5	EXPERIMENTAL RESULTS	24
5.1	Forest fire prediction	24
5.1.1	Dataset	24
5.1.2	Preprocessing	25
5.1.3	Support Vector Machine	26
5.1.4	MLP	27
5.2	WSN Simulation	27
5.2.1	Effect on energy efficiency	29
5.2.2	Effect on fire detection time	30
5.3	Comparison with a similar approach	31
6	CONCLUSIONS	35

List of Figures

4.1	The three layers in our proposed model	15
4.2	Network Architecture	18
4.3	Example of SVM creating a separating line between classes in 2D	21
4.4	Multilayer perceptron (MLP)	22
4.5	overview of the proposed model	23
5.1	Histogram of the fire size in our dataset	25
5.2	A screen capture of CupCarbon Simulator	28
5.3	Energy drop in our proposed model compared to the base model .	30
5.4	Average remaining energy in the approach proposed by [Aslan et al., 2012]	32
5.5	Average remaining energy in our proposed approach	33

List of Tables

5.1	Results of applying SVM.	26
5.2	Results of applying MLP.	27
5.3	Simulation Parameters.	29
5.4	Average detection time of our model and the base model.	31
5.5	Average detection time of our model and the model proposed by Aslan et al.	34

Chapter 1

Introduction

Recent advances in hardware development combined with low cost hardware production and the advances in the field of wireless communication have made it easier to deploy Wireless Sensor Networks (WSNs) in a vast breed of new applications, such as military surveillance, smart city applications, health monitoring and environment monitoring ([Kim et al., 2007], [Chinrungrueng et al., 2007], [Yu et al., 2005]). A WSN is a collection of small and low cost sensing units which are usually powered by batteries. These sensing units have the capability of sensing, processing and communicating via a wireless medium that makes WSNs a practical solution to provide a constant and reliable monitoring for harsh environments. However, with the available technologies, WSNs are battery powered, meaning that, the lifetime of the application running on top of the network is limited by the lifetime of the batteries powering the sensing nodes. As a result, the energy must be consumed in the network in the most efficient and effective way to maximize the lifetime of the application.

[Gia et al., 2015]

In our work, we implemented a framework that aims to increase the efficiency of power consumption in WSNs used for event detection. Environment monitoring applications usually focus on detecting a specific event at a certain time which is usually a small portion of the whole monitoring time. However, in order

to guarantee a fast detection time of the events, the application needs to run at a constant rate, which will cause the network to loose energy while trying to detect an event at times when the event is highly improbable. Our proposed approach makes use of data mining techniques and data analytics within the network to help maximize the activity of the network when an event has a high probability of occurring and in turn, minimizing the network activity at other times. Since WSNs consist of small processing units, data analytics inside the network is not a practical solution or is even impossible. One way of overcoming this limitation is to integrate Fog computing with WSNs.

Fog computing, proposed by [Bonomi et al., 2012], aims to extend the cloud computing paradigm to the edge of the network. It provides processing power, storage and services to the devices at the edge of the network eliminating the need to access the cloud resources and thus reducing the delay caused by the limited bandwidth. Integrating Fog computing with WSNs adds more intelligence and resources to the network enabling it to perform more complex tasks which was not feasible otherwise. This also provides a faster response time for WSN applications that are critical in their nature, such as environmental monitoring.

The added layer of processing power and storage to WSNs makes it possible for the applications to perform data analytics to where the data is generated within the sensing network. Data mining can be used for extracting valuable information from a large amount of data generated by WSNs ([Mahmood et al., 2012]). The generated data hides information that describes the behavior of the environment that is being monitored using WSNs. This extracted knowledge can then be used to better understand the nature of the environment and hence building a framework that is environment-aware. The framework adapts its activity level based on the predictions made by applying data analytics to the data previously generated by the same application or other applications.

In order to test our hypothesis, we applied our platform to the problem of

forest fire detection using WSNs. Lately, WSNs have been proposed as an alternative for the old methods of forest fire detection, such as, camera-based systems ([Mao et al., 2017], [Wang et al., 2009]), infrared detectors and satellite images ([Cuomo et al., 2001], [Giglio et al., 2003]). WSNs have proven to be more accurate than camera and infrared-based systems and they provide a quicker response time compared to satellite based techniques. However, they fall short when it comes to the lifetime of the application. To overcome this shortcoming our framework integrates WSNs with Fog computing to make use of the capabilities added by the Fog computing in analyzing the data generated by the WSNs, such as, temperature, humidity, wind speed, etc. Following the data analysis, our framework employs a predictive model to assess the likelihood of forest fire occurrence based on the readings from the sensing nodes at a certain time period.

The remainder of this paper is organized as follows. Chapter 2 discusses the background of the concepts and technologies we used in our study. Chapter 3 presents the related work on integrating Fog computing with WSNs and the work on using data mining techniques for forest fire prediction. Chapter 4 discusses our proposed architecture for increasing the efficiency of WSNs for forest fire detection. Chapter 5 discusses our experiments and simulation results. Chapter 6 concludes our thesis and provides a discussion on our future work.

Chapter 2

BACKGROUND

In this chapter we discuss the concepts and technologies we used in our proposed solution. We start by presenting the definition and main characteristics of Wireless Sensor Networks and its applications in the field of environment monitoring. Then we present the concept of Fog Computing and the enhancement that Fog can add to WSNs. Finally, we provide a discussion about data mining techniques and how to use them to enhance efficiency of WSN used for environment monitoring.

2.1 Wireless Sensor Networks for environment monitoring

The recent rapid development of electronics and wireless communication made it possible to start the concept of Wireless Sensor Networks (WSN). A collection of small battery powered nodes with limited processing capabilities can cooperate and coordinate collectively to form a network of connected sensors. The resulting network can sense the environment and deliver the results to a centralized processing point (Sink) by using various routing protocols that use multi-hopping to

deliver messages between two different points within the network.

The deployment scheme of the nodes has a significant impact on the efficiency and lifetime of the application, sensing nodes can either be deployed in a regular fashion or in a random fashion. In regular deployment, sensing nodes are placed in a regular pattern with almost equal distances between the sensing nodes. Regular deployment achieves a good balance in power consumption amongst the sensing nodes which extends the lifetime of the network while maintaining full coverage. However, regular deployment is not always feasible due to the requirements of the applications such as the ones that require deploying the network in a remote environment like mountains and forests. On the other side, Random deployment is suitable for applications where regular deployment is not possible. However, sensing nodes have different distances between each other which results in a non equal power consumption since far nodes will consume more energy while trying to deliver a message while closer node will consume less. This difference in power consumption will result in losing parts of the network in the early stages of the network's lifetime.

Another factor that affects the performance of the network is the topology of the network. The network can follow either a flat or clustered hierarchical topology. In the flat topology, sensor nodes are deployed in the space with equal hardware resources and equal responsibilities. In hierarchical topologies, the network is divided into clusters. Within each cluster, a special designated node will work as a Cluster-Head (CH) that controls the other regular nodes in the cluster. The literature contains a wide range of communication and routing protocols for both of the topologies.

The ability of the network to self organize its operations using various routing protocols and algorithms makes it suitable for environment monitoring applications, especially the remote inaccessible locations, such as harsh natural locations

and inaccessible disaster locations where human intervention or traditional monitoring methods are not applicable.

2.2 Fog Computing with Wireless Sensor Networks

Cloud Computing has witnessed a ubiquitous spread in the recent years due to the low cost of storage and the availability of high bandwidth internet connections. Cloud computing has become an efficient replacement to owning and managing a private data center, especially with the 'pay-as-you-go' model offered by most of the cloud service providers. However, the emergence of Internet of things (IoT) requires location awareness and low latency which makes cloud computing fall short in satisfying the needs for the new emerging technologies.

Fog Computing proposed by [Bonomi et al., 2012] comes as the natural extension to overcome the shortcomings of the cloud computing paradigm. Fog Computing forms a virtual platform that offers computing power, storage and networking services at the edge of the network and closer to the end devices. Bringing this new layer of resources closer to the edge of the network provides user's applications with location awareness which can be vital for applications such as Connected Vehicles. The Fog also plays a big role in time critical applications that require an immediate response from the application such as healthcare IoT [Gia et al., 2015], online gaming and video streaming applications

One big contribution of Fog Computing is in the field of WSNs. WSNs consist of motes made of very cheap hardware with small processing power and energy sources which makes them incapable of performing complex data analytics within the network. Combining Fog computing with WSN adds a layer of computational

power and storage closer to the edge of the network (WSN). The new added layer allows the WSN to perform more complex calculations data processing within the WSN, which takes the WSN beyond the phase of only sensing the environment to the phase of analyzing the sensed data and makes use of these analytics to add more functionality to the application running on top of the WSN.

In their work [Naranjo et al., 2017] made use of the Fog computing layer to design a protocol that dynamically opts cluster heads in the WSN based on the remaining energy level. The protocol makes use of the added processing power to calculate the Cluster Heads choice that achieves the best balance in energy between the sensing nodes. Similarly, [Yaseen et al., 2016] leveraged the capabilities of Fog computing to protect wireless sensor network against selective forward attacks. The authors propose that the Fog layer will run a global monitoring application to trace malicious sensors and secure the WSN.

2.3 Data Mining for Environmental Monitoring

Data mining is the field of study that aims to find patterns and statistical dependencies in large dataset to provide insights and useful knowledge about the underlying system from which the data were collected. Machine learning as being one major technique for data mining focuses on developing algorithms that can learn from a collection of historical data in order to provide predictions in the future. The majority of Machine Learning algorithms can be classified into two main categories:

- Supervised learning: refers to a type of machine learning algorithms that requires the training dataset to be labeled with the true class of each record so it can be later used by the algorithm to adjust its learning process based on the label of the data. supervised machine learning algorithm include but not limited to: Artificial Neural Networks (ANN), Support Vector Machines (SVM) and Logistic Regression.

- Unsupervised learning: refers to the type of machine learning algorithms tried to find relations within the training dataset without the need for true labels to be provided during the learning process. unsupervised machine learning algorithms include: K-mean clustering and Latent Dirichlet allocation (LDA).

In our study, we use supervised learning algorithms to predict the probability of forest fires. The algorithms are trained using a data set that combines a set of features that represent the state of the environment under study such as: temperature, humidity, wind speed, time of the year, etc. The dataset is then combined with a label for each record of these features indicating whether that record resulted in a fire incident or not during that day of the year.

With the help of Fog computing, WSN can go beyond their traditional functionality of only sensing the environment and delivering the readings to a sink. The added computational power and storage allow the WSN to apply data analytics to the readings of the sensors to better understand the behaviour of the environment. It also makes it possible to aggregate the readings from the sensors and reduce the amount of data that needs to be sent out to the cloud reducing bandwidth and energy consumption.

[Saoudi et al., 2016] implemented a fire detection WSN that uses data mining techniques to train a machine learning model to predict the probability of forest fires based on the readings of the sensors in the WSN. The model is trained in an offline mode and later stored in the sensing nodes to discard the sensor readings that has a low probability of forest fire. By doing so, their proposition reduces the amount of data transmission between the WSN and the sink and hence reduces the traffic within the network and reduces energy consumption.

Our work makes use of the functionality provided by the Fog computing layer to increase the efficiency of WSNs used for forest fire detection. We utilize the added computational power and storage to provide WSN with a machine learning

model that is dynamically trained on the data collected from the sensing nodes and then adjust the operating frequency of the WSN, minimizing energy loss and enhancing the efficiency of the WSN.



Chapter 3

RELATED WORK

In this chapter, we first present the work that has been done on integrating Fog computing with WSNs. We then discuss the research results on using WSNs for forest fire detection. Finally, we present the studies that focused on using data mining techniques for forest fire prediction.

3.1 Fog Computing and WSNs

[Bonomi et al., 2012] suggest that Fog computing is the solution for overcoming the unidirectional flow of data in WSNs. They state that using Fog computing would allow bidirectional flow of data to enable the network to go beyond the regular sensing of the environment, but also to feed the network with data coming from the top of the hierarchy. [Naranjo et al., 2017] present a routing algorithm that aims to increase the lifetime and enhance the energy efficiency in Fog-supported WSNs. [Brzoza-Woch et al., 2016] make use of the intelligence layer added by Fog computing to WSNs to design a system for use in smart levee monitoring. The authors proposed a layered architecture similar to the architecture we used in our work, and the aim of their application is to analyze the data

generated by the big number of sensors monitoring the environment to detect any potential flood risk. The application is independent of the cloud and hence, the system is able to operate in an independent manner and report the important events. The paper also presents a general architecture for the Fog devices hardware regarding its versatility and ability to receive data from multiple sensing devices. [Lee et al., 2016] make use of Fog computing to provide a smart gateway for Wireless Sensors and Actuator Network (WSAN). The Fog layer lies between the internet and the WSN satisfying the sensing data aggregation, flexible query management between the web and a sensor network, and efficient access by the Internet. [Daniluk, 2015] proposed adding a Fog computing layer on top of WSNs to specifically adjust the sleep state of each sensor node (shallow sleep, deep sleep, awake). His work does not include any specific data analysis strategies, unlike our work which makes use of data mining techniques.[Yaseen et al., 2016] leveraged the Fog Computing infrastructure to secure mobile WSN and detect network intrusions. The authors proposed a heirarichal network architecture similar to the model that we are proposing in our network with the cloud layer being the top layer, followed by the Fog layer, and at the bottom comes the sensors layer. The proposed system makes use of the computational power, storage and networking services offered by the Fog layer to detect selective forwarding in WSN. [Majd et al., 2016] followed a similar approach by utilizing the Fog Computing concept to find an optimal placement of smart access points in WSN. The added virtual computing power added by the Fog layer is utilized to predict the behavior of the network based on information collected within the network.

3.2 WSN for environment monitoring

Lately, WSNs have gained an increasing popularity in the field of environment monitoring. [Cerotti et al., 2014] proposed a comprehensive framework for the use of wireless sensor networks for forest fire detection, the authors suggest that an analytical solution for forest fire detection is feasible regardless of the size and complexity of the model. [Silvani et al., 2015] performed experiments on WSN for fire detection under both lab scale indoor and real outdoor conditions and

they concluded that WSN can be used for fire detection with satisfactory results. [Bhattacharjee et al., 2012] utilized WSNs to detect the initiation of fire in coal mines and then detect the spreading direction in order to contain and stop the fire from spreading. The authors validated their approach using simulation. The main concern of this work is to minimize the delay in fire detection with no focus on energy consumption and the WSN lifetime. [Aslan et al., 2012] proposed a general framework for the use of WSNs in forest fire detection. In their design, the authors focused on four main components of a WSNs: the deployment scheme of the sensing nodes, the logical topology and architecture of the network, intra-cluster communication scheme, and inter-cluster communication scheme. Their work aims to achieve the best detection time for forest fire event, and at the same time, they provided environmental awareness in the network in order to optimize energy consumption. However, the proposed framework does not make use of the generated data for understanding the behavior of the network and it does not provide data analytics within the network. [Yu et al., 2005] proposed a WSN paradigm to achieve real time detection for forest fires. Readings of humidity, temperature and smoke are used as an input for an ANN placed at the cluster heads of the network. The ANN at each cluster head calculates the fire danger given each set of inputs and then decides to either send the aggregated readings of the sensor inputs to the sink or it discards the readings based on a comparison with a predefined threshold. The proposed paradigm reduces the communication overhead and hence, reducing the overall energy consumption. The authors validated their work using simulation and compared their proposed solution against a base model that does not make use of in-network data processing. The results show significant reduction in communication overhead while preserving a good accuracy. [Saoudi et al., 2016] implemented a fire detection WSN which is similar to the work proposed by [Aslan et al., 2012]. The authors provided a hierarchal network architecture that uses data mining techniques in order to limit the number of data transmissions between the sensing nodes and the sink. Their system trains a machine learning model in an off line mood and then, it stores the trained model inside each sensing node before the physical deployment of nodes. This makes it relatively impractical to change the parameters of the trained model after deployment. [Soliman et al., 2010] proposed a fire detection approach that

makes use of a neural network deployed in the base station of a wireless sensor network. Their model is trained using a dataset created by the authors by spreading sensors in an area that is similar to a forest area. The results of Soliman et al.'s work show a very good classification between the fire and non-fire classes. However, the authors did not take energy consumption into account in the scope of their work.

3.3 Data Mining for forest fire prediction

Data mining techniques have been recently used to replace the old methods (Fire Weather Index by [Van Wagner et al., 1987]) for forest fire prediction. [Cortez and Morais, 2007] used meteorological data in order to predict the size of the burnt area caused by forest fires. The authors examined real-world data generated from a forest in the northern part of Portugal. They used different data mining techniques such as Support Vector Machines and Artificial Neural Networks to create a regression model that tries to predict the fire size. The input features included weather condition readings and different components of the Canadian Fire Weather Index (FWI). The regression model was generalized well for forest fires of small sizes. However, the accuracy for fires of large sizes was low. In their work [Satir et al., 2016] used Artificial Neural Networks in order to predict the forest fire probability for different spatial locations. The authors used real-world data from the Mediterranean region. They achieved an accuracy of 0.83, and also found out that the forest elevation, tree cover and temperature are strongly correlated with the problem of forest fire prediction. [Stojanova et al., 2006] used different data mining techniques such as decision trees and logistic regression to predict forest fires in Slovenia. The authors used a combination of geospatial and meteorological data in order to train a binary classifier that predicts the probability of forest fires in a certain day of the year. [Leuenberger et al., 2013] used Random Forests [Breiman, 2001] with a high dimensional feature space in order to predict forest fires in Switzerland. [Sakr et al., 2011] proposed an efficient forest fire prediction model that utilizes the use of only two weather parameters that are

easy to measure in order to reduce the cost of the system and its maintainability. The authors only used cumulative precipitation and relative humidity as the only features to train two different machine learning models, namely Support Vector Machines and Artificial Neural Networks. The experiments showed the ability of the aforementioned machine learning models to predict forest fire probability with a limited number of input.



Chapter 4

PROPOSED APPROACH

In this chapter we propose a model that leverages the capabilities of Fog computing to enhance the performance of WSN for environmental monitoring by enabling data analysis inside the network. The added memory and processing power enable us to analyze the behavior of the WSN based on the generated data within and outside of the WSN. This knowledge is then used to predict the future behavior of the network and hence to consume the energy stored in the batteries in a more efficient and effective manner. Figure 4.1 shows the layers of our proposed model.

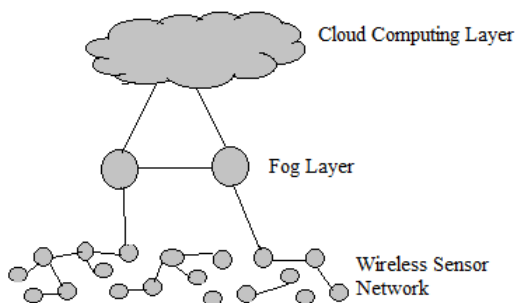


Figure 4.1: The three layers in our proposed model

4.1 Hierarchy

The upper layer of our model includes the cloud computing layer which cooperates with the lower layer (Fog layer) to collect and store historical data generated by the network. In addition to that, cloud computing layer can collect some other data about the environment that cannot be collected by the sensors but it is available through web services and APIs such as, weather forecast, precipitation, etc. After collecting the necessary data, the cloud computing layer builds a predictive model on the collected data in order to predict the probability of an event occurring at any specific time interval (probability of forest fire in our application). The next layer in the hierarchy is the Fog layer, which is mainly used to store the predictive model trained on the cloud computing layer. The Fog layer can also train the predictive model, but since the data size can be huge, this job is usually forwarded to the cloud computing layer. The Fog layer receives the data generated by the sensors in the network and then feeds it to the trained model, in addition to some other data acquired from other services about the environment. Then, the fog layer tries to predict the probability of an event occurring. According to the predicted probability, the Fog node will communicate with the WSN and provide the WSN with the necessary information that can be used for adjusting the activity level of the network, such as, sleep scheduling and working frequency of the sensing nodes.

The idea of integrating Fog computing with WSNs should be applicable for different fields and applications. However, in the work we are proposing, we focus on prolonging the lifetime of WSNs used for forest fire detection.

4.2 Topology and Architecture

The right choice of the WSN topology has a significant impact on the power management of the network, data aggregation in the network, localization accuracy

of the burnt area, and the detection time of forest fires. Two possible choices for WSN topologies are flat and hierarchical. In a flat configuration, sensor nodes are distributed over the target area with similar capabilities and they use multi hopping using a routing protocol to deliver the messages from the source of the network to the sink.

In a hierarchical topology as shown in Figure 4.2, sensor nodes are grouped into clusters where some of the nodes are designed as cluster heads, perhaps with extra processing power, extra memory and GPS devices. The added capabilities to cluster heads allow them to provide control over the other nodes in the cluster. [Aslan et al., 2012] performed several tests on the choice of the topology of a WSN used for forest fire detection. They found out that a hierarchical topology has benefits in terms of enhancing the control over the sensing nodes especially with harsh weather environments. It provides faster detection time for forest fires, and increases the efficiency of power consumption which is the main goal of our work in order to provide the network with a longer lifetime.

As it has been discussed in the work of [Aslan et al., 2012], regular deployment scheme of sensor nodes is preferred when low-energy consumption is required. However, it is not always feasible to deploy the sensor nodes in a regular pattern especially with forests with vast areas and harsh environments. This happens because in a regular deployment scheme, sensor nodes have similar distances between each other and hence they would transmit messages to approximately similar distances, and the energy consumption will be similar in all the sensor nodes. However, in random deployment scheme, some nodes will be close to cluster heads resulting in low transmission energy consumption but there will always be distant nodes consuming more energy and this consumption will cause the average energy level of the network to drop faster than the regular deployment scheme.

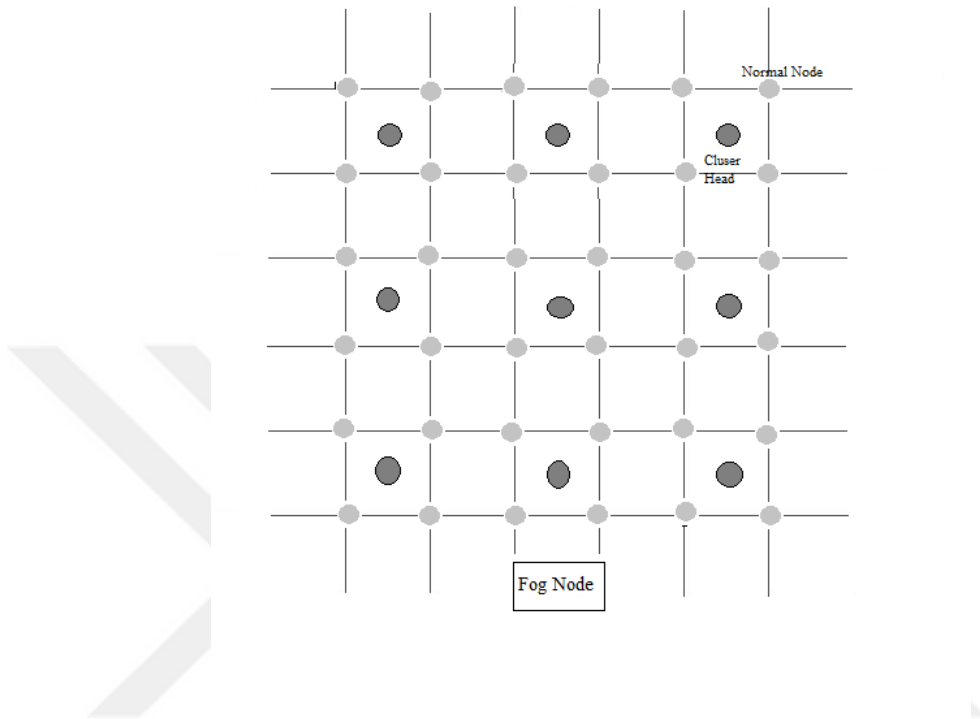


Figure 4.2: Network Architecture

4.3 Communication protocols

Here we illustrate the communication protocols used in our WSN. Our communication model takes into account the probability of fire outburst in the forest and accordingly, it adjusts the communication behavior in order to efficiently use the energy stored in the sensor nodes. The formation of clusters and the selection of cluster heads is out of the scope of this work. However, several works in the literature suggest different techniques for the formation of clusters and also recovering from the death of cluster heads as discussed by [Gholami and Brennan, 2016].

Assuming that the clusters have been formed and the cluster heads have been selected, the cluster heads will send a message to their connected nodes controlling their sending sequence so that they can manage the communication through the shared medium. Cluster heads also send information about the working frequency of the sensing nodes in the bottom of the hierarchy. They obtain these information from the fog node that has already trained the predictive model as

discussed previously. The fog node first receives the weather data collected from the sensing nodes deployed in the forest. It also collects the necessary data about the environment that could not be obtained from the sensing nodes. After preparing the features for that instance of time, the fog node will feed the data to a predictive model. Then, depending on the probability of fire outburst at that moment of time and given the environment inputs, the fog node will send the appropriate working frequency and other necessary information to the cluster heads. In the next step, the cluster heads will distribute their messages to the sensing nodes.

During the fire free periods, the sensing nodes will be following the sending frequency assigned to them by their cluster heads. Some of the nodes inside the cluster can be put into sleeping mood in order to save energy in case there is a very low probability of fire outburst. The working frequency that was initially sent by the fog node will be followed as long as there is no fire thread. Once a sensor node detects a fire thread, it interrupts the working frequency schedule and sends an alarm as soon as possible to its cluster head, and it keeps sending the alarm messages until the cluster head confirms receiving the fire alarm message. The cluster head in turn sends the necessary information to the fog node in order to take the proper actions. The fog node will assign more time slots for the nodes attached to the cluster head to gather more information about the progress of the fire and perhaps predicting its direction in case the fog node keeps a map of the deployment locations of the sensors. This will be very helpful with directing the firefighters to the fighting locations.

During a fire progress, there is a high risk that the sensing nodes and cluster heads will die because of the generated heat. [Aslan et al., 2012] addressed this issue in their work as follows: if the damaged node is a normal sensing node, then the cluster head will be able to detect it and take the necessary action. But, if the damaged node is a cluster head, then there are two scenarios. The first of these scenarios is that the cluster head recognizes the danger of the thread, and in this case it can assign the responsibility of cluster head to any other node in the cluster that has a lower risk of getting damaged. The second scenario is that

the cluster head dies suddenly, and to overcome this, the cluster head sends a replication of its data to the nodes in the cluster as soon as there is a fire thread. Then, if the cluster head dies suddenly, the cluster nodes will detect the event by realizing the absence of ACK messages sent by the cluster head. The first node detecting the absence of a cluster head will exchange messages with the other members of the cluster, and they can together decide on the new cluster head based on the risk criteria discussed earlier.

4.4 Data Mining

The data generated by the WSN carries hidden patterns about the behavior of the network. Finding these patterns will be very useful in the case that power management is a major concern for sustaining the lifetime of the network, especially in the applications that employ WSNs for event detection. Event detection applications operate in a continuous fashion in order to monitor the environment, while the probability of an event occurring at certain time periods is very minimal. Hence we need to minimize the activity of the WSN at those times.

In our work, we analyze real life forest fire data. The dataset contains features that can be obtained from the WSN itself such as relative humidity, temperature and wind speed. It also contains data that can be acquired from external sources about the area under study such as precipitation, cloud cover, etc.

We integrate data mining techniques in our forest fire detection architecture in order to discover the hidden patterns within the data generated by the sensor nodes, which is now feasible due to the potentials of Fog computing. Fog computing provides us with a relatively big processing power that makes it possible to perform data analysis within the network. We are primarily interested in creating a model that can use historical data to predict the probability of forest fire.

We use the data generated by the network at any given moment of time to decide whether we have a fire danger or not. This brings our problem into a

classification problem. Classification is one of the most popular data mining techniques that aim to predict the probability of a new set of features to belong to a predefined class.

In our work we used support vector machines and Artificial Neural Network to train a classifier that can generalize future instances of the data generated by the network.

4.4.1 Support vector machine (SVM)

SVM is a discriminative classifier from the supervised learning family. SVM transfers the input data into a high dimensional feature space by using nonlinear mapping, and then it finds the best linear hyperplane that maximizes the separation between the different classes as shown in Figure 4.3.

$$y = w_0 + \sum \phi(x) \quad (1)$$

In Formula (1) $\phi(x)$ is a nonlinear transformation of the input data vector X . With the proper choice of the kernel function and using cross validation to tune the parameters of the classifier (C, λ) [Cherkassky and Ma, 2004]. SVM generates a hyperplane separating the two classes of interest (fire, no fire).

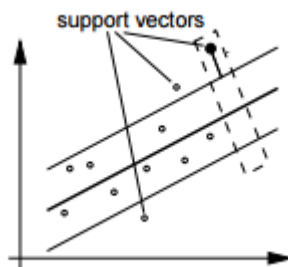


Figure 4.3: Example of SVM creating a separating line between classes in 2D

4.4.2 Artificial neural network (ANN)

ANN is a mathematical model that tries to mimic the learning process done by the human brain in the way it learns from previous examples. Then, it uses this knowledge acquired during the learning phase in order to generalize to new examples which haven't been introduced to the system before.

In our work, we used the multi-layer perceptron neural network (MLP). An MLP creates a directed graph of fully connected layers called hidden layers as shown in Figure 4.4. Each hidden layer consists of multiple processing units called neurons. The neurons receive signals from the previous layer and then use an activation function to generate an output to the layers that follow. MLP uses a supervised learning technique called *back propagation* which uses the training data instances to change the weights of the network. This adjustment is performed to minimize the error between the true output and the output predicted by the ANN. Since the output of our network is binary (fire, no fire) we used the *sigmoid function* as the activation function. When the ANN finishes the learning phase, new instances of the data can be fed to the model to predict their class. Note here that we define a threshold at the output of the sigmoid function to perform the classification task.

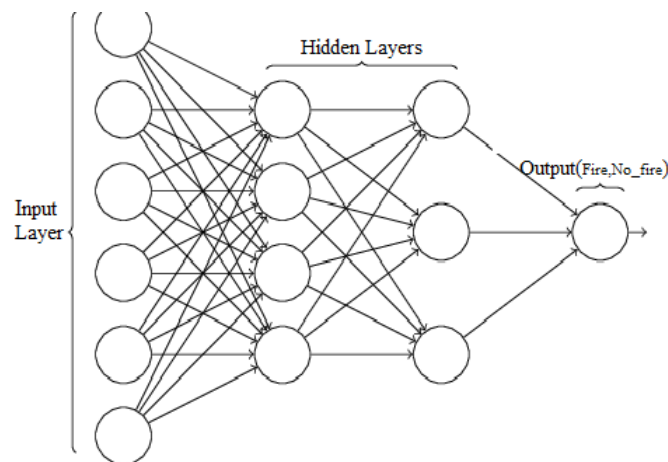


Figure 4.4: Multilayer perceptron (MLP)

$$F(i) = 1/1 + \exp(-x) \quad (2)$$

After training the predictive models on the cloud, the model computed in that stage is stored in the Fog node. Then, the system can feed the data generated by the network to the predictive model to predict the Probability of the event. Depending on the application, the Fog node can prepare the appropriate messages to be sent to the cluster heads. These messages contain information about the working frequencies of the sensor nodes at the lowest level of the hierarchy. A small probability of forest fire will make it possible for the sensor nodes to sense the environment with a relatively long time interval between any two consecutive readings, in the mean while the sensor node can go to a sleeping mode saving the energy of that sensor node.

Increasing the time interval of sensing the environment will result in an increased discovery time in case a fire starts in the forest. However, by using our predictive model we lower the sensing frequency of sensing the environment only if the probability of forest fire occurrence is small. The increased detection time for the events happening in the low probability period is the trade off we are paying for saving the energy of the sensors and extending the lifetime of the environment. Figure 4.5 provides an overview for the proposed model.

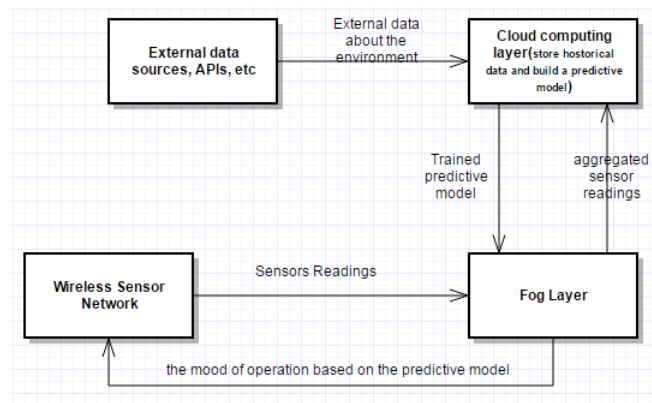


Figure 4.5: overview of the proposed model

Chapter 5

EXPERIMENTAL RESULTS

In this chapter, we describe the experiments we performed to evaluate our proposed model. We first propose our work in building a predictive model for predicting the probability of forest fire occurrence. Later on, we use the trained model to validate our proposed architecture by performing extensive simulation experiments on energy consumption in WSNs.

5.1 Forest fire prediction

5.1.1 Dataset

In order to build our predictive model for forest fire prediction, we obtained historical forest fire data from the official site of the United States Department of Agriculture Forest Service [Short, 2017]. The dataset contains incidents of fire combined with a time stamp, time of fire occurrence, reasons for fire outburst and fire size. In the next step of building our data set, we obtained the meteorological data for the area under study from the University of California for Agriculture

and Natural Resources. This dataset includes features about temperature, humidity, wind speed, etc. The two datasets were then aggregated based on the time stamps of each data entry. The resulting dataset contains meteorological data for each day of the year, with the size of fire on the days where a forest fire occurred based on the first dataset. We had different sizes of fires in our dataset, but the majority of fires are small as shown in Figure 5.1. Note that we used $\log(\text{size} + 1)$ in order to reduce the skewness in the observations

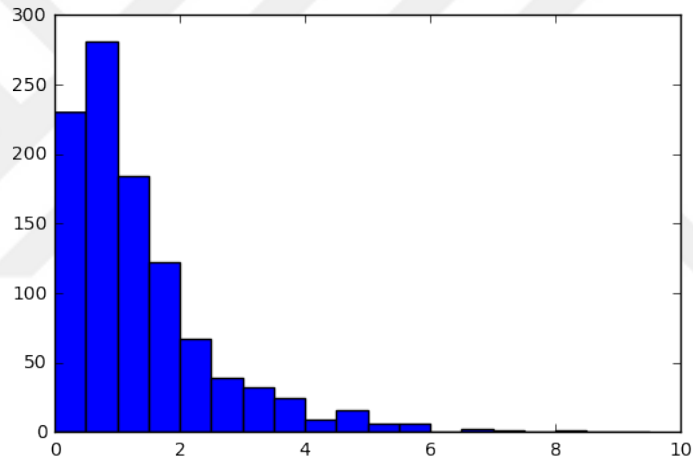


Figure 5.1: Histogram of the fire size in our dataset

5.1.2 Preprocessing

After merging the two dataset together, we performed several preprocessing steps in order to prepare the dataset to be used by the machine learning algorithms.

- The predictive models we used in our work might behave unpredictably if the feature has different ranges for the value and follows different distributions. In order to make the features more consistent, we prepared the features to have zero mean and one unit variance by subtracting the mean

value and then scaling them by dividing the features by their standard deviation.

- Our dataset consists of a collection of numerical and categorical features. However, the predictive models we used in our experiments work only with numerical values. As a result, we had to convert the categorical data we have in our dataset to their numerical representation such as the day of the week and the day of the year. We applied a famous method called One Hot Encoding. One Hot Encoding first transfers categorical features into numerical values by just assigning a numerical value for each category of that specific feature. Then, it maps each value to a vector that is filled with zeros except for the position indexed by the value that has been previously assigned to the categorical value

5.1.3 Support Vector Machine

We started building our predictive model by using support vector machine (SVM). We used 10 fold cross validation to choose the parameters for our SVM classifier. After obtaining the best model using cross validation, we fed the test dataset to the model, and the results are as shown in Table 5.1.

Table 5.1: Results of applying SVM.

Class	precision	recall	F1 score
no-fire	0.86	0.80	0.83
fire	0.74	0.82	0.78
avg/total	0.81	0.81	0.81

5.1.4 MLP

Using MLP, we obtained similar results to the results generated by SVM as shown in Table 5.2. We obtained the best accuracy in our validation set by implementing an MLP with one hidden layer which contains 80 neurons and each neuron implements tanh as the activation function.

Table 5.2: Results of applying MLP.

Class	precision	recall	F1 score
no-fire	0.71	0.71	0.71
fire	0.86	0.86	0.86
avg/total	0.81	0.81	0.81

While validating our results, we aimed to maximize the recall value for the fire class, but this resulted in an increase in the false positive values. This result can be accepted in our application since we can tolerate false alarms while operating our WSN. The effect of that would be an increased activity of the WSN in a time interval with a low probability of fire occurrence, hence wasting more energy. However, missing a fire event is a more critical issue since our system will assume the absence of fire based on the false negative readings. As a result, it will slow down the activity of the WSN resulting in an increased detection time. This is more critical than the first case, since the aim of our application is reduce power consumption, but at the same time, maintaining the efficiency of the network in terms of detection time. As a result, we minimized the false negative rate of fire incidents by liberating our classifier to accept more false positive predictions.

5.2 WSN Simulation

The second phase in our experimental set-up was to evaluate our proposed approach by using simulation. In this section, we present our simulation experiments and show the effect of integrating data mining techniques on extending the life time of Fog-supported WSNs.

CupCarbon: CupCarbon ([Mehdi et al., 2014]) is a smart city and IoT WSN simulator. Although our application is not in the area of smart city applications, the open source environment made it possible to adapt the simulator and perform the simulation for forest fire WSNs. CupCarbon provides simulation for radio propagation in the medium, offers the possibility to simulate several events such as fire and gas propagation, and keeps track of the energy consumption during the lifetime of the sensor network. The energy calculations take into account the different components of the network, such as sensing energy, sleep mood energy, packet transmission/reception energy, etc. Figure 5.2 shows a sample of the simulator.

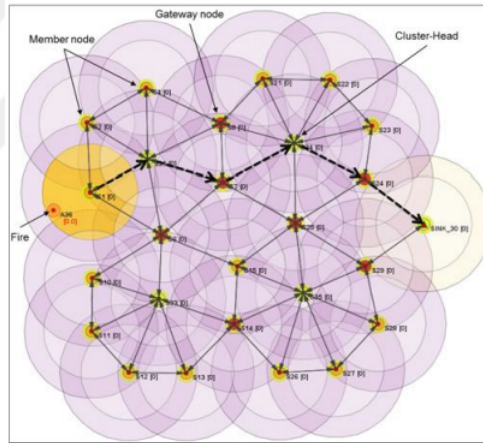


Figure 5.2: A screen capture of CupCarbon Simulator

We implemented our proposed approach using CupCarbon to study the effect of integrating data mining techniques and Fog computing on extending the lifetime of WSNs. We performed several experiments with different number of nodes deployed on the space. As we mentioned earlier, we followed a hierarchal architecture for our deployment with the fog computing node on top of the hierarchy followed by a logical layer of cluster heads, and at the bottom of the hierarchy comes the layer of sensing nodes. Table 5.3 lists the collection of simulation parameters we used in our experiments. We mainly focused on studying the effect of the proposed approach on energy consumption and the effect on fire detection time.

Table 5.3: Simulation Parameters.

Parameter	Value
Radio Radius	100 m
Sensing Energy	100J
Transmission Energy	59.2J/byte
reception Energy	28.6J/byte
Listening Energy	1.0 J/ sec
Sleeping Energy	0.1 J /sec
MAC Protocol	802.15.4

5.2.1 Effect on energy efficiency

Our sensor network tries to adapt its activity level based on the input from the predictive model. Sensing nodes sense the environment and send their readings to their cluster heads. Cluster heads in turn aggregate the data and use multi hopping to deliver the data to the Fog node. The Fog node has already received a trained predictive model that predicts the probability of forest fires based on the data collected from the sensor nodes. Accordingly, the Fog node commands the cluster heads to increase the activity level of the network in case there is a thread of forest fire or it commands it to reduce its activity otherwise. This results in an efficient model for preserving the sensors energy and extending the lifetime of the network. As a consequence, we performed extensive simulations on this proposed approach and then compared it with the base model that works at a constant rate and it doesn't take into account the predictions for forest fire probability.

We used one year worth of data taken from the test data set we used earlier to develop our predictive model. This data set contains a time stamp and forest fire probability at that specific time. We also adjusted the source code of the CubCarbon simulator to create a Fog node that stores the prediction list and then it feeds it to the network dynamically based on the run time of the simulation. Figure 5.3 shows the average energy drop using our model and also the energy drop using the base model.

Note that our approach consumes less energy compared to the base model.

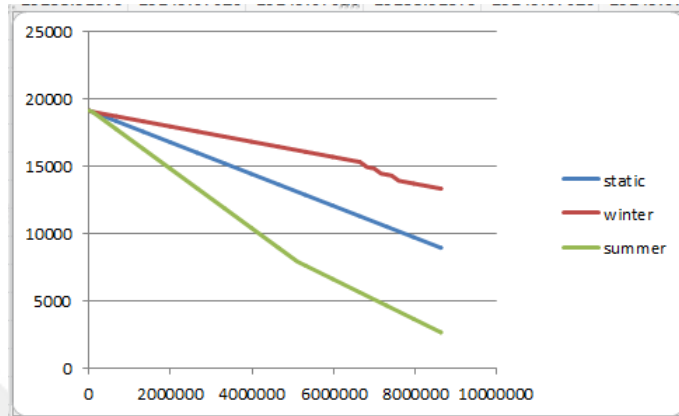


Figure 5.3: Energy drop in our proposed model compared to the base model

However, during the summer times, our predictor generates a higher number of false positive values. As a result, our model consumes more energy during the summer time. The increased activity for our model during summer time provides the forest fire application with a faster response compared to the base model, which can be seen as a trade off for consuming more energy during the summer time.

5.2.2 Effect on fire detection time

Our method tries to reduce the activity of WSNs by increasing the time interval between each time a sensor attempts to sense the environment and sends its data to the sink. The lower the probability of forest fire the lower the sensing frequency. As a result, the sensing nodes can then be scheduled for sleep in order to save energy. On one hand, this technique minimizes the power consumption of the network; but on the other hand, it increases the detection time in case our predictor fails to successfully predict the probability of fire. In this section, we present the evaluation results for the effect of our new approach on the detection time. Table 5.4 shows the average detection time for correctly predicted events, misclassified events (false negative) and fire events occurring in the base model.

Note that for correctly predicted events, our approach detects fires faster than the base model. However, the detection time for the misclassified events can take more time compared to the base algorithm. This is because, our models assumes a low probability of fire occurrence, and as a result it tries to save energy by reducing the activity rate of the network.

Table 5.4: Average detection time of our model and the base model.

Event type	Time (sec)
Correctly predicted events	23.5
Misclassified events (false negative)	61
Events in Base model	44

5.3 Comparison with a similar approach

In this section, we compare our approach with other approaches that try to minimize power consumption in WSN by controlling the activity level of the network depending on the season of the year and the expected danger level of that season on the probability of having a forest fire. [Aslan et al., 2012] proposed a general framework for the use of WSNs in forest fire detection. In their design, they provided environmental awareness in the WSN in order to optimize energy consumption. Their proposed scheme keeps the activity level of sensor nodes low in months when the risk of fire is quite low, like the months of winter season and high on summer season.

On the other side, our approach adjusts its activity level based on the outcome of a trained machine learning model that is regularly trained on a real life data. The environmental conditions are fed into the model to calculate the fire probability of that specific day. In order to compare the two approaches, we used the CupCarbon simulator ([Mehdi et al., 2014]) to build a WSN for each model. The two networks have exactly same type of sensors, number of sensors and distances between sensor nodes. The only difference is in the communication protocol used by the WSN. As we mentioned earlier, the work proposed by [Aslan et al., 2012] adapts its activity level (frequency of sensing the environment and message exchange between nodes and the gateway) based on the month of the year. Figure

5.4 shows the average remaining energy in the sensor nodes throughout a year of simulation. The results show a slow rate of power consumption in the early months of the year and then the consumption rate becomes higher, and finally it adjusts to a slower rate by the end of the year.

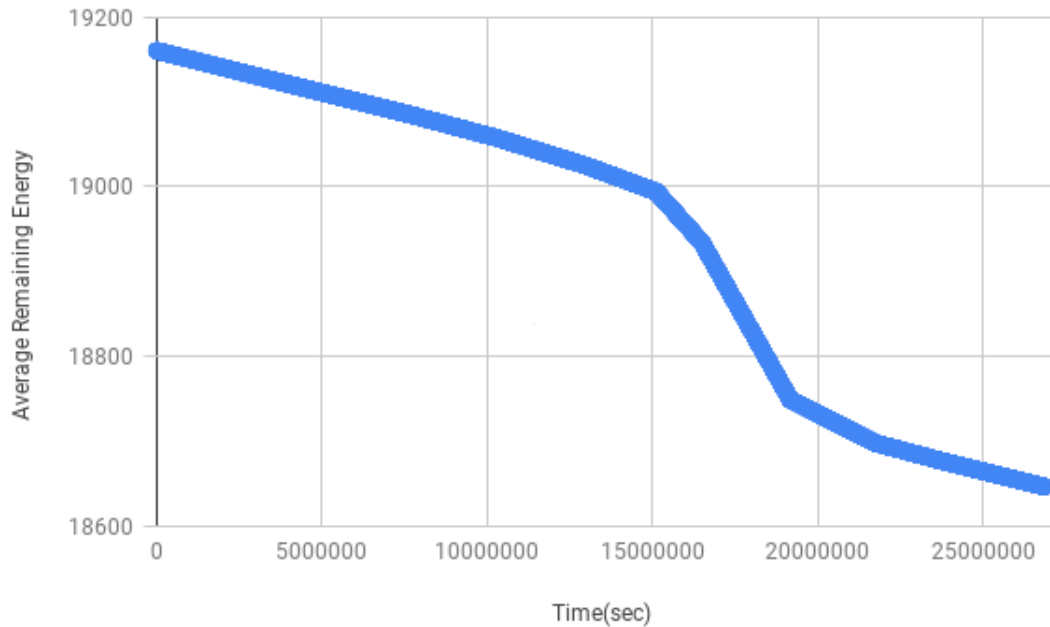


Figure 5.4: Average remaining energy in the approach proposed by [Aslan et al., 2012]

Figure 5.5 shows the average remaining energy in the sensor nodes after running the simulation with our proposed approach which takes daily readings from the Fog node about the predicted fire probability of that specific day and accordingly it adjusts the activity level.

The two approaches are showing similar behavior in energy consumption throughout the year. They start with small energy consumption at the beginning of the year, after that there is an increased activity of both of the approaches during the summer time. Finally, the consumption rate slows down again by the end of the year. The experiments show a larger amount of energy consumed

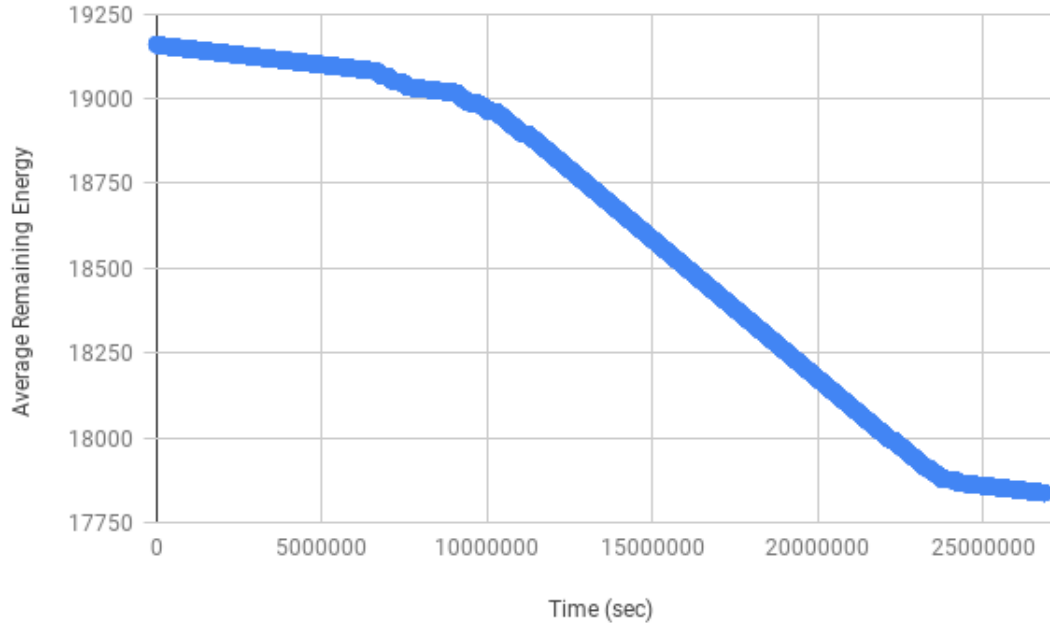


Figure 5.5: Average remaining energy in our proposed approach

by our approach compared to the one proposed by [Aslan et al., 2012]. Our approach calculates the risk factor of a certain day using real life dataset and it might generate readings with high risk periods even in winter times as it can be seen from the irregularity of the graph in Figure 5.5 in the early months of the year. Whereas, the other approach assumes that the risk factor will be small in winter and high in summer as a general rule hard coded in the sensing nodes and it does not dynamically change according the environmental conditions of the forest. Several real life datasets suggest that fire incidents occur during unexpected periods (such as winter and Autumn). That in turn proves that our approach is more sensitive and accurate in detecting forest fires during all seasons since the machine learning model is trained on real life datasets and it calculates the fire risk dynamically everyday.

As we mentioned earlier, our model does not assign a fixed risk value of forest fires at a specific time, instead, it dynamically predicts forest fire probability using the machine learning model that is trained on real life datasets. The dataset we

used to train our model contained instances of forest fires during unexpected seasons such as the winter times. On the other hand, the model proposed in [Aslan et al., 2012] assumes that probability of forest fires during winter seasons is small and then it assigns a small risk value for the whole month. To observe the difference in detection times between the two models, we measured the time that our model took to report fire incidents that was predicted during winter times and compared it against the time reported by the other model for the same time of the year. Our model showed a smaller detection time since it was able to detect the fire and as a result, it increases the activity level of the network leading to a faster fire detection. Whereas, the other model was operating at the same fixed value assigned to that month and hence, it was operating at a slow frequency, leading to a larger detection time. The two models showed similar behavior for fire detection during the summer time since the model of [Aslan et al., 2012] operates the WSN at a higher frequency, similar to our model. Table 5.5 shows the results of our experiment on fire detection time.

Table 5.5: Average detection time of our model and the model proposed by Aslan et al.

	Our Model	Model proposed by Aslan et al.
Avg. detection time in summer Seasons	23(sec)	22(sec)
Avg. detection time in winter Seasons	31(sec)	61(sec)

Chapter 6

CONCLUSIONS

Integrating Fog computing with WSNs adds a new layer of intelligence to the network. This layer provides the WSN with more processing power, storage and location awareness. In our work, we tried to leverage the capabilities of the added layer in order to enable data analysis inside the network. Analyzing the data about the environment where the WSN is deployed provides us with an insight on the behavior of the environment. As a result, we can use this knowledge to optimize the efficiency of the network and limit the waste in energy. To validate our hypothesis, we applied our model to the application of forest fire detection using Fog-supported WSNs. We built a predictive model on historical forest data using Support Vector Machines (SVM) and Multilayer Perception (MLP) neural network. Then we used the prediction in a simulation set-up to measure the loss in energy and the speed of detecting fires.

In our proposition, the Fog node maintains a machine learning model that has been trained on several features about the environment such as temperature, relative humidity, wind speed, precipitation, time of the year, etc. The model tries to predict the probability of forest fire given the aforementioned features at any certain point in time. In order to train our model, we collected a real life

dataset from one of the forests which contains information about the meteorological features of the environment in addition to some other information about the time and the size of fires at that specific area. After training our machine learning model on the collected dataset, we obtained good results demonstrating the relationship between the features of the environment and the probability of fire occurrence. However, we believe that the accuracy of our predictive model could reach more accurate levels by obtaining a dataset with a higher number of observations and more features about the environment which would also enhance the efficiency of the WSN employing the predictive model.

The next step in our experimental setup was to perform extensive simulations to validate our hypothesis. We modified an open source software to build a Fog-supported WSN that employs a machine learning model that coordinates the activity level of the WSN based on the predicted danger level by the ML model.

We compared our model against a base model that operates at a fixed rate with no adaptation to the surrounding environment. We designed the experiment to run our model with a frequency that is lower than the base frequency for low fire probability and a frequency that is higher than the static model when the probability of forest fire is high. Our model consumed less energy in the winter months since the ML model predicted a low probability of forest fires for the majority of the days. The model predicted a larger number of high risk days during the summer months, since we operated the WSN at a very high frequency during the days with high probability of forest fire. Our model consumed more energy than the base model during those times. However, the model improved the detection time of forest fires compared to the base model which is an important factor in containing the fire and limiting its spread.

Besides comparing our model against a base model, we also compared it to a similar model from the literature which aims to increase the efficiency of the WSN by making the WSN aware of surrounding environment [Aslan et al., 2012]. The model we used for comparison makes the WSN adapt to the environment by assigning a risk value for each month and the activity of the WSN depends on this value to control the activity level of the WSN. A small risk value is assigned

for the months which are expected to have a low probability of forest fires and a high risk value is used for the summer months. The results of our simulation showed a similar pattern of energy consumed by both models. However, the total energy consumed by our model has shown to be higher than the other one. After investigating the results, we found out that our model returns a higher number of risk predictions even during the non-summer seasons, and by looking at the real life dataset, we found out that fire outbursts occur even in non-summer seasons which gives our WSN the ability to detect those events and operate at a high frequency while the other model will not be able to react fast enough to detect those events since it assigned a small risk values for this time of the year.

Our future work will focus on enhancing the accuracy of the predictive model by training the model on a dataset which contains a higher number of observations and more features about the environment. We will also investigate and simulate the process of retraining the machine learning model on the Fog nodes within the network and by that the network will be able to update the predictive model to capture the changes of the environment on the long run.

Bibliography

- [Aslan et al., 2012] Aslan, Y. E., Korpeoglu, I., and Ulusoy, Ö. (2012). A framework for use of wireless sensor networks in forest fire detection and monitoring. *Computers, Environment and Urban Systems*, 36(6):614–625.
- [Bhattacharjee et al., 2012] Bhattacharjee, S., Roy, P., Ghosh, S., Misra, S., and Obaidat, M. S. (2012). Wireless sensor network-based fire detection, alarming, monitoring and prevention system for bord-and-pillar coal mines. *Journal of Systems and Software*, 85(3):571–581.
- [Bonomi et al., 2012] Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM.
- [Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- [Brzoza-Woch et al., 2016] Brzoza-Woch, R., Konieczny, M., Nawrocki, P., Szydło, T., and Zielinski, K. (2016). Embedded systems in the application of fog computing?levee monitoring use case. In *Industrial Embedded Systems (SIES), 2016 11th IEEE Symposium on*, pages 1–6. IEEE.
- [Cerotti et al., 2014] Cerotti, D., Gribaudo, M., and Bobbio, A. (2014). Markovian agents models for wireless sensor networks deployed in environmental protection. *Reliability Engineering & System Safety*, 130:149–158.
- [Cherkassky and Ma, 2004] Cherkassky, V. and Ma, Y. (2004). Practical selection of svm parameters and noise estimation for svm regression. *Neural networks*, 17(1):113–126.

- [Chinrungrueng et al., 2007] Chinrungrueng, J., Sunantachaikul, U., and Triamlumlerd, S. (2007). Smart parking: An application of optical wireless sensor network. In *Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on*, pages 66–66. IEEE.
- [Cortez and Morais, 2007] Cortez, P. and Morais, A. (2007). A data mining approach to predict forest fires using meteorological data. In *New Trends in Artificial Intelligence, Proc. of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence, APPIA, 2007*, pages 512–523.
- [Cuomo et al., 2001] Cuomo, V., Lasaponara, R., and Tramutoli, V. (2001). Evaluation of a new satellite-based method for forest fire detection. *International Journal of Remote Sensing*, 22(9):1799–1826.
- [Daniluk, 2015] Daniluk, K. (2015). Smart decision fog computing layer in energy-efficient multi-hop temperature monitoring system using wireless sensor network. In *FedCSIS Position Papers*, pages 167–172.
- [Gholami and Brennan, 2016] Gholami, M. and Brennan, R. W. (2016). A comparison of alternative distributed dynamic cluster formation techniques for industrial wireless sensor networks. *Sensors*, 16(1):65.
- [Gia et al., 2015] Gia, T. N., Jiang, M., Rahmani, A.-M., Westerlund, T., Liljeberg, P., and Tenhunen, H. (2015). Fog computing in healthcare internet of things: A case study on ecg feature extraction. In *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, pages 356–363. IEEE.
- [Giglio et al., 2003] Giglio, L., Descloitres, J., Justice, C. O., and Kaufman, Y. J. (2003). An enhanced contextual fire detection algorithm for modis. *Remote sensing of environment*, 87(2):273–282.

- [Kim et al., 2007] Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., and Turon, M. (2007). Health monitoring of civil infrastructures using wireless sensor networks. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 254–263. ACM.
- [Lee et al., 2016] Lee, W., Nam, K., Roh, H.-G., and Kim, S.-H. (2016). A gateway based fog computing architecture for wireless sensors and actuator networks. In *Advanced Communication Technology (ICACT), 2016 18th International Conference on*, pages 210–213. IEEE.
- [Leuenberger et al., 2013] Leuenberger, M., Kanevski, M., and Vega Orozco, C. D. (2013). Forest fires in a random forest. In *EGU General Assembly Conference Abstracts*, volume 15.
- [Mahmood et al., 2012] Mahmood, A., Shi, K., and Khatoon, S. (2012). Mining data generated by sensor networks: a survey. *Information Technology Journal*, 11(11):1534–1543.
- [Majd et al., 2016] Majd, A., Sahebi, G., Daneshtalab, M., Plosila, J., and Tenhunen, H. (2016). Placement of smart mobile access points in wireless sensor networks and cyber-physical systems using fog computing. In *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld), 2016 Intl IEEE Conferences*, pages 680–689. IEEE.
- [Mao et al., 2017] Mao, W., Wang, W., Dou, Z., and Li, Y. (2017). Fire recognition based on multi-channel convolutional neural network. *Fire Technology*, pages 1–24.
- [Mehdi et al., 2014] Mehdi, K., Lounis, M., Bounceur, A., and Kechadi, T. (2014). Cupcarbon: A multi-agent and discrete event wireless sensor network design and simulation tool. In *Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques*, pages 126–131. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

- [Naranjo et al., 2017] Naranjo, P. G. V., Shojafar, M., Mostafaei, H., Pooranian, Z., and Baccarelli, E. (2017). P-sep: a prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks. *The Journal of Supercomputing*, 73(2):733–755.
- [Sakr et al., 2011] Sakr, G. E., Elhajj, I. H., and Mitri, G. (2011). Efficient forest fire occurrence prediction for developing countries using two weather parameters. *Engineering Applications of Artificial Intelligence*, 24(5):888–894.
- [Saoudi et al., 2016] Saoudi, M., Bounceur, A., Euler, R., Kechadi, T., et al. (2016). Intelligent data mining techniques for emergency detection in wireless sensor networks. In *2nd IEEE International Conference on Cloud and Big Data Computing (CBDCom 2016), Toulouse, France, 18-21 July*. IEEE.
- [Satir et al., 2016] Satir, O., Berberoglu, S., and Donmez, C. (2016). Mapping regional forest fire probability using artificial neural network model in a mediterranean forest ecosystem. *Geomatics, Natural Hazards and Risk*, 7(5):1645–1658.
- [Short, 2017] Short, K. C. (2017). Spatial wildfire occurrence data for the united states, 1992-2015. 4th Edition. Fort Collins, CO: Forest Service Research Data Archive.
- [Silvani et al., 2015] Silvani, X., Morandini, F., Innocenti, E., and Peres, S. (2015). Evaluation of a wireless sensor network with low cost and low energy consumption for fire detection and monitoring. *Fire Technology*, 51(4):971–993.
- [Soliman et al., 2010] Soliman, H., Sudan, K., and Mishra, A. (2010). 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.
- [Stojanova et al., 2006] Stojanova, D., Panov, P., Kobler, A., Džeroski, S., and Taškova, K. (2006). Learning to predict forest fires with different data mining techniques. In *Conference on Data Mining and Data Warehouses (SiKDD 2006), Ljubljana, Slovenia*, pages 255–258.

- [Van Wagner et al., 1987] Van Wagner, C., Forest, P., et al. (1987). Development and structure of the canadian forest fireweather index system. *Canadian Forest Service, Forest Technical Report 35, Ottawa, Canada.*
- [Wang et al., 2009] Wang, S.-J., Jeng, D.-L., and Tsai, M.-T. (2009). Early fire detection method in video for vessels. *Journal of Systems and Software*, 82(4):656–667.
- [Yaseen et al., 2016] Yaseen, Q., AlBalas, F., Jararweh, Y., and Al-Ayyoub, M. (2016). A fog computing based system for selective forwarding detection in mobile wireless sensor networks. In *Foundations and Applications of Self* Systems, IEEE International Workshops on*, pages 256–262. IEEE.
- [Yu et al., 2005] Yu, L., Wang, N., and Meng, X. (2005). Real-time forest fire detection with wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, volume 2, pages 1214–1217. IEEE.