Transportation Consortium of South-Central States

Tran-SET

Solving Emerging Transportation Resiliency, Sustainability, and Economic Challenges through the Use of Innovative Materials and Construction Methods: From Research to Implementation

Development, Education, and Implementation of a Low-Cost Audio Sensor-based Autonomous Surveillance System for Smart and Connected Transportation Infrastructure Construction and Maintenance

Project No. 19PPLSU12

Lead University: Louisiana State University

Final Report

August 2020

Disclaimer

Acknowledgements

The research team would like to thank you the Project Review Committee including Mr. Jacob Fusilier, Senior Project Manager, LaDOTD, Dr. John Song, Associate Vice President , AECOM, and Mr. Jason Bittner, Principal Analyst and Practice Area Lead, ARA for their insightful suggestions and feedback.

TECHNICAL DOCUMENTATION PAGE

1. Project No.	**2. Government Accession No.**	**3. Recipient's Catalog No.**
19PPLSU12		

4. Title and Subtitle Development, Education, and Implementation of a Low-Cost Audio Sensor-based Autonomous Surveillance System for Smart and Connected Transportation Infrastructure Construction and Maintenance	**5. Report Date** August 2020 **6. Performing Organization Code**
7. Author(s) PI: Yong-Cheol Lee https://orcid.org/0000-0002-0040-0894 Co-PI: Jin-Woo Choi https://orcid.org/0000-0002-9210-9681	**8. Performing Organization Report No.**
9. Performing Organization Name and Address Transportation Consortium of South-Central States (Tran-SET) University Transportation Center for Region 6 3319 Patrick F. Taylor Hall, Louisiana State University, Baton Rouge, LA 70803	**10. Work Unit No. (TRAIS)** **11. Contract or Grant No.** 69A3551747106
12. Sponsoring Agency Name and Address United States of America Department of Transportation Research and Innovative Technology Administration	**13. Type of Report and Period Covered** Final Research Report Aug. 2019 – Aug. 2020 **14. Sponsoring Agency Code**

15. Supplementary Notes

Report uploaded and accessible at Tran-SET's website (http://transet.lsu.edu/).

16. Abstract

Each DOT has to govern and oversee an enormous number of transportation construction and maintenance projects. However, since a transportation construction project entails several miles of a job site including numerous work tasks and equipment operations, it has been increasingly challenging for each DOT to consistently monitor progress of all projects in each State as well as efficiently evaluate work performance. In particular, with limited human resources and time, DOTs in Region 6 States have managed large-scale transportation construction and maintenance projects by a human inspection and recovered direct and indirect damages of transportation infrastructure systems caused from the recent natural disasters. In this demanding situation, DOT practitioners and project managers have long recognized the importance of automated monitoring and surveillance of transportation construction and maintenance processes that helps consistently track work progress and take immediate remedial action. As one promising supplement for site monitoring and human inspection, this project proposes a new approach for low-cost audio sensor-based autonomous site and safety surveillance of transportation construction and maintenance, which allows for faster, more convenient, and more accurate work zone monitoring. The proposed innovation using the sound-based site and safety monitoring framework possesses several competitive advantages over traditional site management and existing vision-based work monitoring methods, which not only sounds can be easily recognized and instantly analyzed by diverse sound sensors. In addition, this sound-based monitoring approach supports an unlimited range of monitoring angles and illumination levels with lightweight data processing and comparatively quick analytics. To achieve these goals, this study developed a low-cost wearable audio-sensor for automated work zone monitoring and real-time activity log generation. This new intelligent site and safety surveillance system is expected to support real-time monitoring of construction progress, evaluation of task performance, and rapid identification of safety issues in transportation construction and maintenance projects.

17. Key Words	**18. Distribution Statement**
Workzone monitoring, Sound classification, Activity identification, Real-time supervision	No restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report)	**20. Security Classif. (of this page)**	**21. No. of Pages**	**22. Price**
Unclassified	Unclassified	46	

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized.**

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
^oF	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	^oC
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERATURE (exact degrees)		
^oC	Celsius	1.8C+32	Fahrenheit	^oF
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m^2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS, ABBREVIATIONS, AND SYMBOLS

API	Application Programming Interface
ASR	Automatic Sound Recognition
BIM	Building Information Modeling
DNN	Deep Neural Network
DOT	Department of Transportation
DOTD	Department of Transportation and Development
GMM	Gaussian Mixture Models
GPS	Global Positioning System
HMM	Hidden Markov Models
IaaS	Infrastructure as a Service
IoT	Internet of Things
I2C	Inter Integrated Circuit
I2S	Inter-IC Sound
JSON	Java Scrip Object Notation
MCU	Microcontroller Unit
MEMS	Micro-electromechanical Systems
MFCC	Mel Frequency Cepstral Coefficient
PaaS	Platform as a Service
RFID	Radio Frequency Identification
RMS	Root Mean Square
SaaS	Software as a Service
SBC	Single Board Computer
SQL	Structured Query Language
STFT	Short Time Fourier Transform
SVM	Support Vector Machines
USB	Universal Serial Bus
UWB	Ultra-Wide Band
Wi-Fi	Wireless Fidelity
ZCC	Zero Crossing Count

EXECUTIVE SUMMARY

Each DOT has to govern and oversee an enormous number of transportation construction and maintenance projects such as new bridge construction or highway maintenance. However, since a transportation construction project entails several miles of a job site including numerous work tasks and equipment operations, it has been increasingly challenging for each DOT to consistently monitor progress of all projects in each State as well as efficiently evaluate work performance. In particular, with limited human resources and time, DOTs in Region 6 States have managed large-scale transportation construction and maintenance projects by a human inspection and recovered direct and indirect damages of transportation infrastructure systems caused from the recent natural disasters. In this demanding situation, DOT practitioners and project managers have long recognized the importance of automated monitoring and surveillance of transportation construction and maintenance processes that helps consistently track work progress and take immediate remedial action. The lack of an automated and robust monitoring system of construction processes and work performance in the transportation construction industry has resulted in significant cost and schedule overruns in almost 90 percent of transportation construction and maintenance projects, bringing relevant safety issues. Therefore, fulfilling the demand for improved transportation construction processes and establishing an unmanned field and safety monitoring framework is one of high priority needs for Region 6 States. Even though some transportation construction projects adopted an expensive vision-based monitoring system using on-site cameras or drones, these methods involve a tremendous amount of a visual data analysis and are vulnerable to numerous blind spots of any site. They also require computationally heavy-weight processing, which hinders real-time analytics and monitoring and always require a certain level of illumination.

As one promising supplement for site monitoring and human inspection, this project proposes a new approach for low-cost audio sensor-based autonomous site and safety surveillance of transportation construction and maintenance, which allows for faster, more convenient, and more accurate work zone monitoring. The proposed innovation using the sound-based site and safety monitoring framework possesses several competitive advantages over traditional site management and existing vision-based work monitoring methods, which not only sounds can be easily recognized and instantly analyzed by diverse sound sensors. In addition, this sound-based monitoring approach supports an unlimited range of monitoring angles and illumination levels with lightweight data processing and comparatively quick analytics. To achieve these goals, this study involved the following steps. First, the PIs collected sound data from real transportation construction and maintenance projects and web resources including more than 150 data with 14 types of work activities and equipment operations. By using the deep neural network (DNN)-based sound classification technique, this system dynamically enhanced sound classification accuracy. Second, this system includes a developed low-cost wearable audio-sensor for automated work zone monitoring and real-time activity log generation. The light-weight sound data can be readily captured, analyzed, shared, and visualized for real-time site monitoring and activity log generation that have not been achieved by the current methods because of limited human resource for in-person site monitoring and heavy-weight video data processing. This new intelligent site and safety

surveillance system is expected to support real-time monitoring of construction progress, evaluation of task performance, and rapid identification of safety issues in transportation construction and maintenance projects.

Efficient work zone monitoring and management of numerous highway construction sites can be achieved through continuous remote monitoring and real-time activity log generation. To accomplish these goals, the PIs propose audio-based monitoring and activity log generation of highway construction work zones that innovatively transform noise into meaningful history data. Sounds emanating from work zones provide crucial insights regarding work activity types,

1. INTRODUCTION

The ever-growing need to construct more roads, bridges, tunnels, highways, etc. to enhance commute causes the DOT in each state in the U.S. to govern and oversee a large number of transportation construction and maintenance projects each year simultaneously. These projects generally span across several miles and involve numerous tasks and equipment operations. The DOT is endowed with the duty to continuously monitor the progress and constantly evaluate the quality of performance of each of these projects so that timely decisions can be made and the project can be executed as per schedule. As such, there is a significant need for a real-time project management and performance monitoring system that can help the DOT professionals and project managers to consistently monitor multiple projects at various locations simultaneously through cloud-based remote methods. In this project the researchers explored and developed a cloud-based integrated autonomous monitoring and surveillance system that utilizes sound and other data collected from low-cost and light-weight wearable sensors in real-time from the job site for forecasting progress, working conditions, emergencies, performance quality, and safety issues.

The traditional approach for tracking work progress and project status is manual- and human-based site supervision requiring professionals to be present on the site for the job, causing the need to hire a large number of professionals for just supervising multiple projects. With the automated surveillance system, the supervisors will be capable of simultaneously monitoring multiple job sites from a remote location. Furthermore, since job sites stretch over several miles, there is a considerable need for dynamic interaction and data collection methods such as sensors that can be automatically relocated to the new portion of the stretch as the job progresses. If static sensors are used, it may require additional manual human resources for relocating the sensors to new positions as the job progresses. In addition, since construction sites are dynamic in nature, the position of these sensors need to be planned in advance so that they do not come in the way of equipment, machinery, and workers.

Sound-based data collection is largely dependent on the distance of an audio sensor from a point of occurrence of an event due to issues related to audibility and background noise. If static sound sensors are utilized and they are fixed in positions that are further away from the occurrence of the events so as not to come in the way of construction jobs, it can reduce the efficiency of the system for identifying the ongoing activity correctly. As such, in this project we proposed the use of the wearable sensor technology where workers can make the sensor a part of their job attire and carry them around while doing the construction jobs without any hassle, thereby granting the much-needed dynamic nature to the sensors.

The efficiency of sound-based data collection and activity identification can be determined based on the establishment of a sound library including sound samples from each of the machinery and equipment to be used in the project that is used for training the machine learning model for accurate prediction of the activity. Since there is a vast number of construction equipment and machinery that are used in transportation construction and maintenance work and numerous brands that manufacture them, it is imperative to create a sound library of the specific equipment and machinery that is to be used in the project to increase the accuracy of the model. Finally, this project explored the use of a cloud-based platform that is capable of streaming the data collected by the sensors directly into the cloud. The data is then captured, stored, and analyzed on this cloud-based platform in real-time for valuable information that can help supervisors gain appropriate insight into the status and progress of the project for making timely decisions. The cloud-based platform also allows for a visualization platform for supervisors where the insights from the analyzed data is continuously streamed from all active construction sites almost in real-time.

2. OBJECTIVES

As one promising supplement for site monitoring and human inspection, this study proposes a new approach for low-cost audio sensor-based unmanned site and safety surveillance of transportation construction and maintenance, which allows for faster, more convenient, and more accurate work zone monitoring. The proposed innovation using the sound-based site and safety monitoring framework possesses several competitive advantages over traditional site management and existing vision-based work monitoring methods, which not only sounds can be easily recognized and instantly analyzed by diverse sound sensors. In addition, this sound-based monitoring approach supports an unlimited range of monitoring angles and illumination levels with lightweight data processing and comparatively quick analytics. In this proposed study, the PIs aim to investigate the way to improve the accuracy of audio-based event detection and implement the prototypes in transportation construction and maintenance work zones with practitioners in Region 6 States' DOTs.

The capabilities of this system are not only limited to identify the on-going activity at the job site, but also extend to identifying work progress, emergent situations, working conditions, and safety issues. The outcomes of this research involve new scientific knowledge on the implications of sound recognition and cloud-based platform integration for analysis and visualization in the transportation construction sector for real-time monitoring and forecasting. The main objectives of this project to achieve the goal, can be listed as follows:

- Development of a low-cost and light-weight wearable sensor that is capable of capturing various data from the project site in real-time such as sound-data, GPS location, climatic conditions such as temperature, humidity, air pressure, etc.

- Development of a Deep Neural Network (DNN) model based on project schedule for accurate identification of on-going activity at the construction site.

- Integrating a cloud-based platform for real-time streaming, storage, analysis and visualization of results from data analysis.

- Development of a prototype system for real-time monitoring of multiple projects simultaneously through web-based visualization.

3. LITERATURE REVIEW

The monitoring of work quality and progress of infrastructure construction and maintenance projects that spread over several miles has always been a critical challenge for the project managers in the transportation construction industry. In addition, the identification and prevention of safety issues is an added challenge that have restricted the improvement of construction quality and management. According to the International occupational safety statistics, the construction industry entails one of the highest industry accident rates *(1)*. In addition, the National Census of Fatal Occupational Injuries survey conducted by the U.S. Bureau of Labor Statistics in 2015 shows that there were 4836 workers killed on construction sites due to illness and fatalities, and that 12% of injuries were caused by the exposure to hazardous environment, fire, and explosion *(2).* The primary causes of the construction accidents are the unique nature of the industry including the harsh environment, the mistaken behavior of an unskilled worker, and a lack of safety awareness, poor safety management, and weak enforcement of mandatory safety rules *(1,2)*. Roadway construction projects requires workers to work in more adverse situations amidst harsh weather conditions and speeding traffic making them more vulnerable to safety and health hazards. On one hand there is a need to construct more roads in the US to enhance commute and on the other there is endless maintenance work that needs to be complied for the already existed ones in a timely manner. Thus, the ability to recognize issues related to safety, work quality, and progress is critical for infrastructure construction as it can help to avoid unforeseen accidents *(3)*. It is imperative to properly adopt advanced safety identification technologies and security management systems for consistently monitoring construction work activities and accurately recognizing construction safety implications *(4).* According to Cheung et al., construction safety issues and risky situations can potentially be enhanced by advanced information and communication technologies *(5).*

Traditional approaches for identifying safety issues and unexpected risks related to job quality, and progress are generally manual in nature, depending on human reporting and manual data collection. As such, issues are often recognized late; consequently the necessary follow-up measures are taken late when unexpected situations occur. To overcome the limitations of manual efforts, automated monitoring and identification is a promising approach for robust management on a construction site *(6).* With the rapid development of information technology, researchers have been constantly exploring new and innovative methods for integrating data acquisition techniques such as sensors, vision-based tracking, global positioning system (GPS), ultra-wide band (UWB), and radio frequency identification (RFID) to collect real-time field data in construction *(7-12).* The sound-based method for real-time data collection is a comparatively new approach, which has proven significant benefits and reliability for activity detection and event monitoring. Sound data when compared to image data require less storage space and therefore have less data processing weight for data analysis. In addition, a sound-based method can be equally efficient for capturing data during the night as it does require a certain level of illumination like the visual-based tracking *(13).*

Environmental sound recognition has become popular over the past decade, leading to an extensive amount of research. In an Automatic Sound Recognition (ASR) system, sound recognition using signal processing and machine learning techniques. Various applications have been developed for audio-based surveillance, sound event recognition related to ASR system *(14).* It is an emerging technology that entails several benefits for both indoor and outdoor activities and event detection. In recent times, a number of researches have been conducted to explore the applicability of a sound-based monitoring system especially in public traffic safety and private driving securities *(15),* detection of car crash *(16),* recognition of driver's inattentive behaviors *(17),* speed limit check on highways *(18),* and voice recognition system for prevention of car thefts and accidents *(19)*. In terms of outdoor monitoring, an audio-based method is used to detect impulsive sounds including glass breaks, human screams, and gunshots by employing classification algorithms such as Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM) *(20)*. In health and safety management, sound signals generated by body falls and distress speech expressions are applied to detect emergency incidents for patients with attached equipment at home *(21)*.

Recently, several studies investigated audio-based applications for improving construction processes and management. One study explored the potential application of the speech recognition technology for bridge inspection and showed key factors influencing the results *(22)*. Cheng et al. explored the implementation and the benefits of an audio-based approach for activity identification of heavy equipment using a Support Vector Machine (SVM) *(23)*. Cho et al. employed the frequency domain approach to identify overlapped sounds generated by different activities simultaneously at a construction site *(24)*. In recent years, many research studies have been conducted to explore the accuracy and efficiency of the DNN model for sound-based event identification *(25-28)*. Even though these studies proved the feasibility and the benefits of an audio-based field monitoring and its implications, this approach entails a critical problem in establishing a sound data library covering heterogeneous sound types of equipment and work activities.

The construction job site is dynamic in nature and movement of workers and equipment keep changing making it difficult to pre-plan a specific location for installing the sensor without coming in the way of construction activities or compromising with the accuracy of the data collected. In this context, many researchers have explored innovative technologies such as mobile sensors and equipment mounted smartphones *(29-30)* to grant a dynamic nature to the sensors so that the challenges of a dynamic environment can be met out without compromising accuracy. Furthermore, the need for a cloud-based system for data management and analysis has been long felt in the construction industry for enhancing real-time information update and sharing among the different stakeholders involved in the project. However, developing a fully connected cloud based full-fledged project management system operated in real-time is still in a conceptual phase and yet to see the light of the day. This project is aimed to take a step forward in the above direction by using a cloud-based system for collecting and analyzing field data obtained through sensors in real-time to forecast specific information related to the project, such as progress, on-going activity, working conditions, and emergency situations.

4. METHODOLOGY

The proposed framework was developed using advanced and state-of-the art technologies that included low-cost and light-weight sensors that can be worn by the construction crew as a part of their job attire, deep neural network model for sound classification, cloud-based computing platforms, and web-based visualization. In this research, we used the Microsoft Azure platform as the main cloud-based platform for collection, storage, computing and visualization of field data and Autodesk Forge along with Postman for hosting the project model for visualization. In addition, Raspberry Pi was used for linking the sensor with Microsoft Azure, Figure 1 below shows the research plan and data flow for integration of sensors worn by workers at the construction site with the cloud-based platforms.

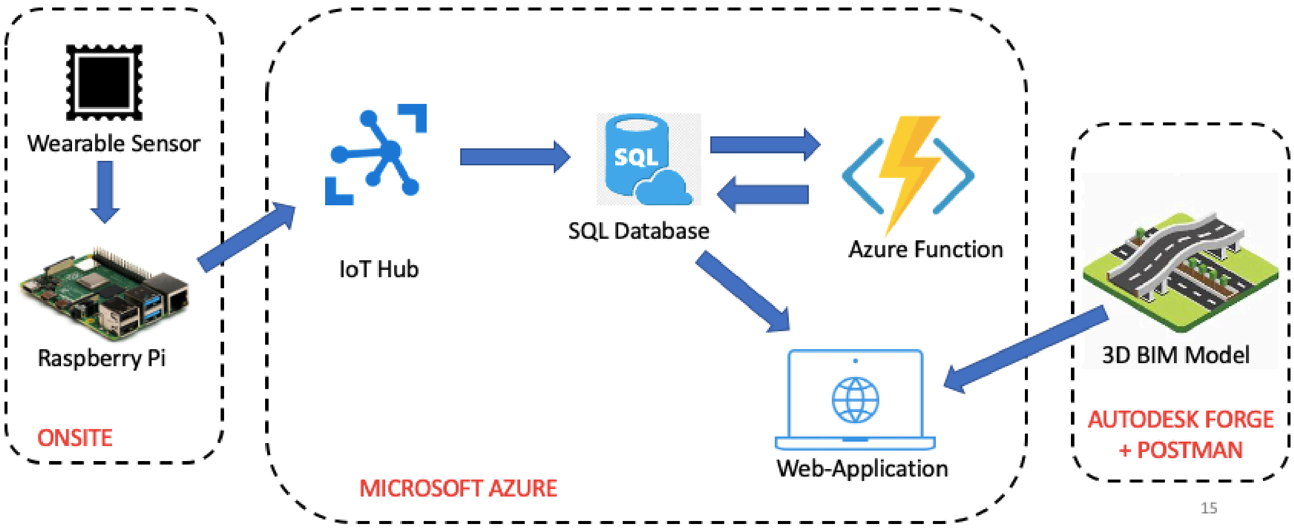


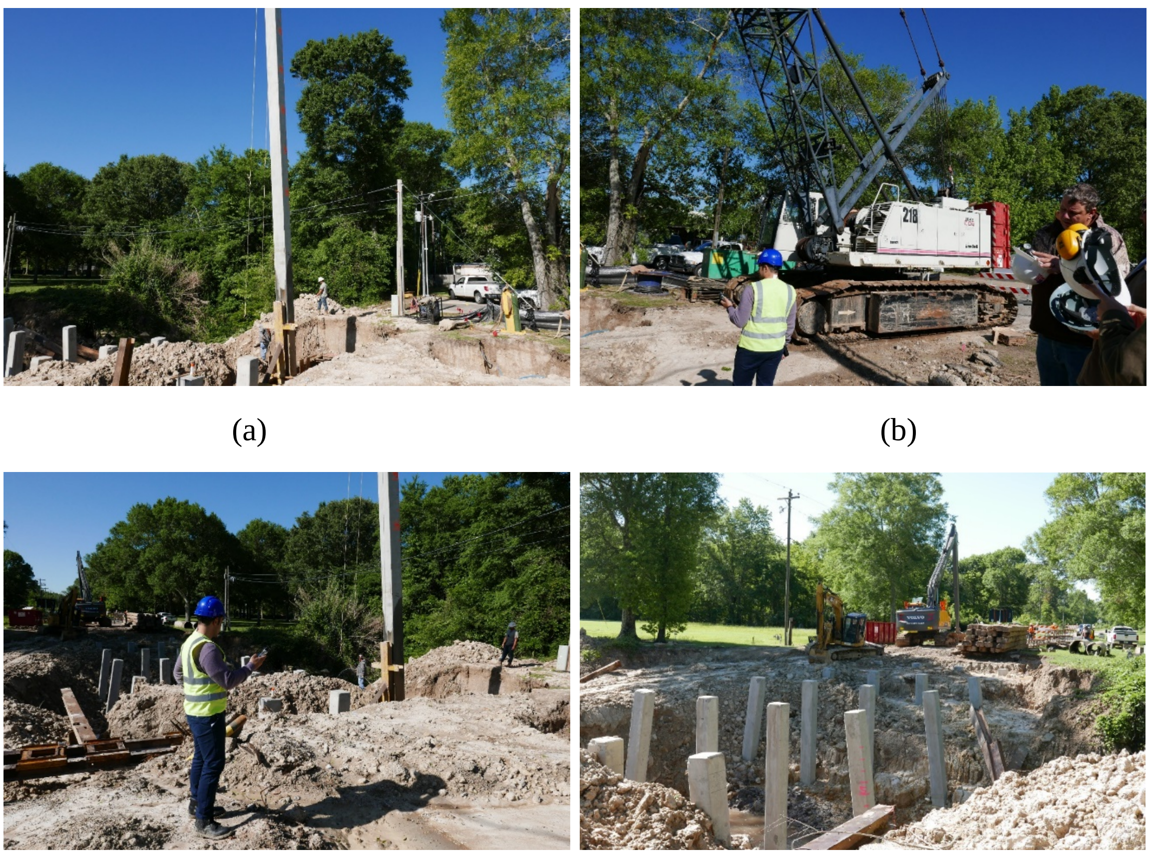
Figure 1. Research plan and data flow.

To achieve the proposed objectives the research was conducted in the following six phases:

- Phase One: Collecting and cleaning sound data
- Phase Two: Developing a DNN-based sound classification algorithm
- Phase Three: Developing a low-cost and light-weight wearable sensor
- Phase Four: Integrating Sensor Data and cloud-based computing platform
- Phase Five: Setting up the cloud-based platform
- Phase Six: Developing a web-based application for visualization

4.1. Phase One: Collecting and cleaning sound data

To adopt a sound-sensor based approach for construction progress and safety monitoring, there were several fundamental and practical challenges involved. One of the major concern was the accuracy and the reliability of sound recognition as it was not practically demanding for the system to establish a versatile sound data library. Heterogeneous types of working sounds are generated from a diverse working environment, materials, work behaviors, and others. Therefore, covering all sound types of construction work activities and equipment operations in a single library is not practically viable. Thus, a sound data archive for transportation construction sound was developed as per the project schedule and equipment and machinery used. This helped to achieve the proposed objective of real-time monitoring and forecasting of work progress as well as pre-notification and rapid identification of unexpected situations at the construction site. The data was collected in collaboration with Dane Lecoq in the Louisiana Department of Transportation and Development (DOTD). Most of the data was collected from two off-system bridge projects and one roadway construction project managed by DOTD in Louisiana. However, some data was also collected from various web resources.



(a) (b)

(c)

(d)

Figure 2. Sound collection at the Tucker Road Bridge.

First, this study established a project-specific sound data library for distinctly covering the sound training data sources planned in a construction project. This library provided reliable sound training data that explicitly represented the different types of sound of the planned activities. All the sound data collected were then split into short segments of 2s each in length. Sound samples containing disturbance or background noises were manually eliminated. The final library consisted of over 2700 sound samples of 2s each distributed under 14 categories of construction equipment and machinery. The list is given in the table below:

Table 1. List of equipment/machinery in the sound library.

Equipment/Machinery	No. of sound samples
Excavator	200
Concrete Mixer	200
Compactor	193
Bulldozer	154
Dumper	200
Dragline	200
Vibrator	200
Crusher	200
Shovel	200
Roller	200
Grader	200
Distributor	200
Paver	200
Scraper	200

4.2. Phase Two: Developing a DNN-based sound classification algorithm

The sampling rate used in this study is 22.05 KHz, which is the default value used by the Librosa package that was used for analyzing the sound files in Python. Table 2 below shows the list of features that were extracted from each of the sound samples in this study. Some of the extracted features of the sound samples are shown in Figure 3.

Table 2. List of features extracted.

Symbol	Description
MFCC	Mel Frequency Cepstral Coefficient
STFT	Short Time Fourier Transform
Chroma	Chromagram
Mel	Melspectogram
Contrast	Spectral Contrast
Tonnetz	Tonal Centroid
ZCC	Zero Crossing Count
RMS	Root Mean Square

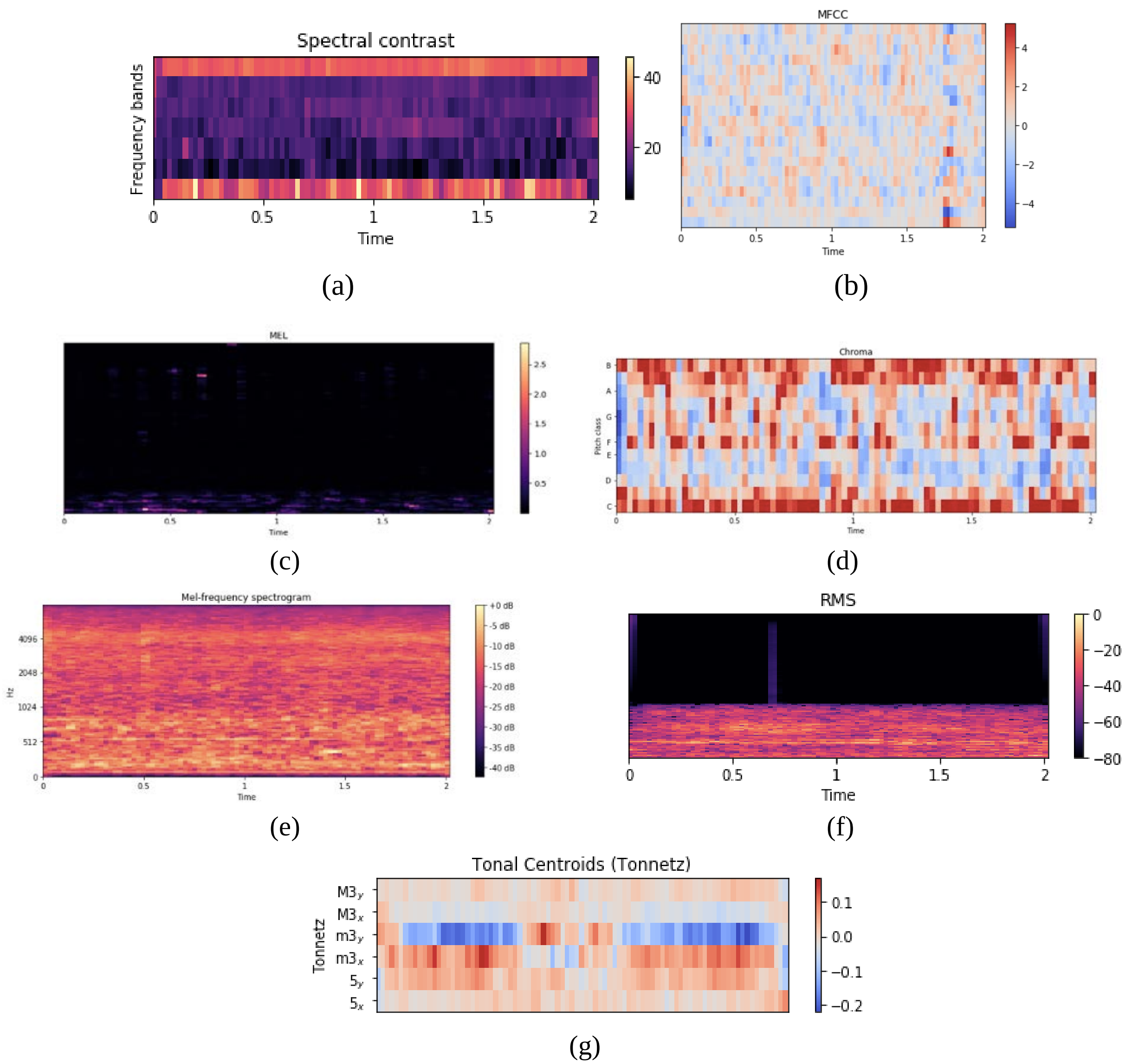


Figure 3. Visual Images of Extracted Features: (a) Spectral Contrast for Loader, (b) MFCC for Excavator, (c) Mel feature of Bulldozer, (d) Chroma for Concrete Mixer, (e) Mel frequency for Concrete Pump, (f) RMS for Vibrator, (g) Tonnetz for Compactor.

In this study, a Deep Neural Network (DNN) model was used for sound classification. This study was also previously conducted using a KNN classifier, but it was found that for a KNN classifier, the accuracy was too low to be reliable. The accuracy, in case of a KNN classifier, can be sufficiently improved using a daily planned schedule. However, the researchers believe that making the sound classification dependent on daily work schedule compromises on the level of automation that is aimed for, as it means that any change in daily schedule would require manual reprogramming of the system before it can be used for accurately monitoring the ongoing activity. Thus, the DNN model was developed in Python using the Keras library which is an open source neural network library. The DNN model is a fully established commonly used classifier in the field of sound, speech and image classification *(31-33)*. It is an artificial neural network with multiple layers between the input and output layers. The DNN relevant studies show a high recognition accuracy that can be adapted for leveraging the sound-based construction activity and operation monitoring. The architecture of the DNN model used for this study consists of an input layer, two hidden layers with 100 units in each, and an output layer. The model implements the Adam algorithm for optimization. Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments. The source code for the model is provided in Appendix A. The deep structure is shown below in Figure 4.

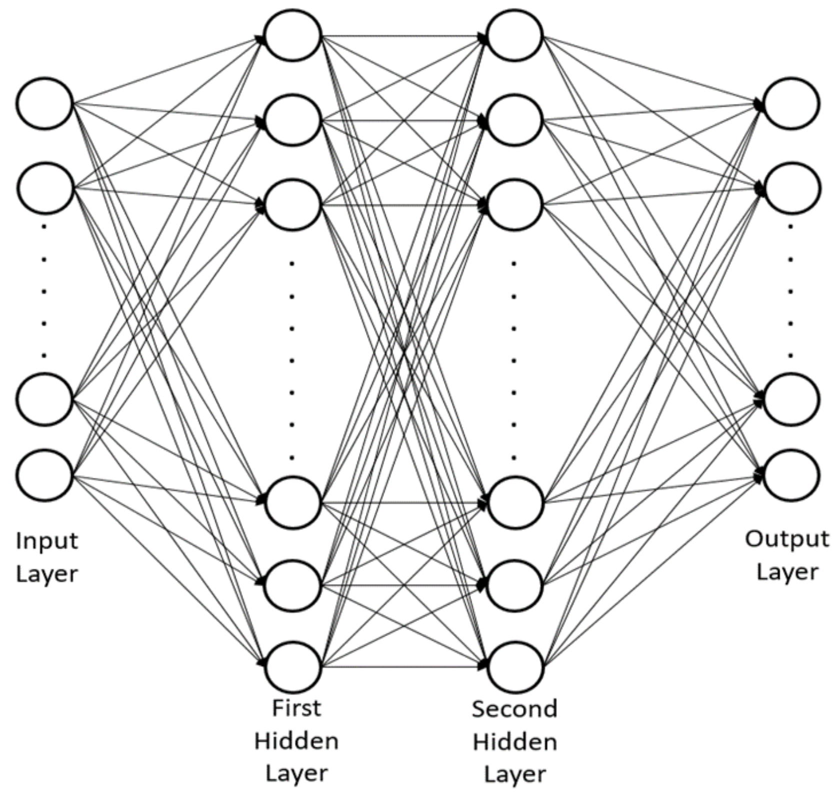


Figure 4. Architecture of the DNN model.

The accuracy of sound classification of the trained DNN model was evaluated with the help of confusion matrices and F score. The entire library containing over 2700 sound samples of 2s each in length were randomly divided into training and testing sets. 70% of these files were used for training while 30% were used for testing. The model was trained for 50 epochs. The results of the F-score and confusion matrices are shown in Figure 5 and Figure 6 respectively. The classification accuracy is 99%.

$$\text{Precision} = \frac{True\ Positives}{Trus\ Positives + False\ Positives} \quad [1]$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad [2]$$

$$\text{F score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad [3]$$

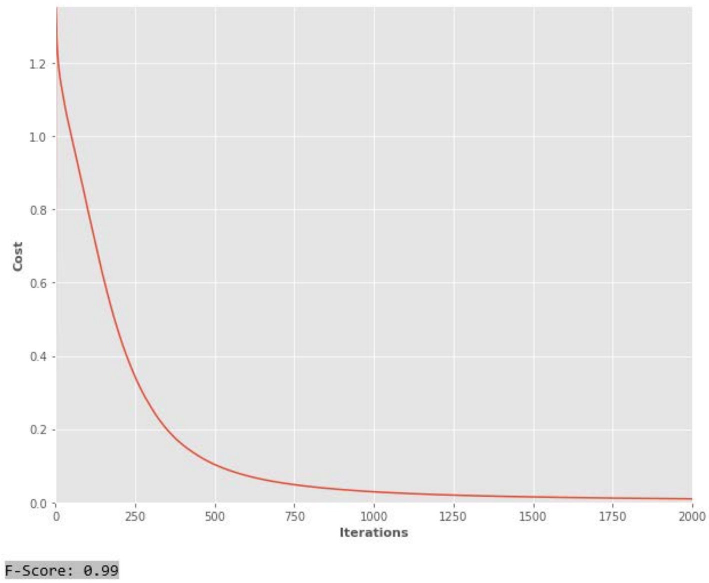


Figure 5. Accuracy evaluation of the DNN model using F-score.

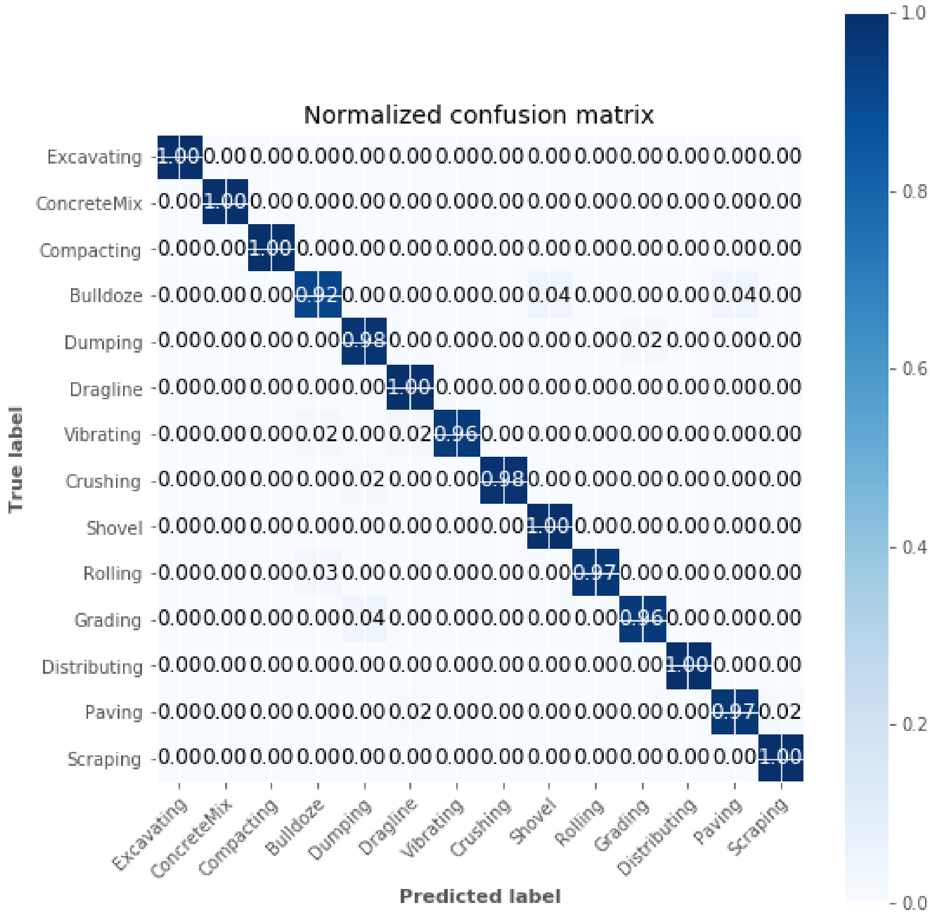


Figure 6. Classification accuracy of the DNN model using confusion matrix.

As is evident from Figure 5 and Figure 6 above, the DNN model has a very high accuracy level and therefore is capable of correctly predicting on-going activities at the construction site.

4.3. Phase Three: Developing a low-cost and light-weight wearable sensor

In recent times, several researches in the field of construction has been exploring the capacity of a wearable sensor technology for capturing on-site data in real-time. In a study of noise hazard prediction framework, Wei et al., integrated a wearable sensor into BIM data [34]. One of the main advantages of a wearable sensor over the traditional ones is that a wearable one provides a dynamic character to the sensor device. A construction site by nature is dynamic, as it keeps changing as the work progresses. As such using a traditional sensor that is static in nature and is fixed at a particular location can be disadvantageous as it would need to be relocated to a suitable location every time the construction job progresses beyond the capturing range of the sensor device. Also, there is high chances of the sensors coming in the way of movement of workers, equipment, and machinery and getting damaged if its location is not properly planned, causing unwanted hassle.

For this study, the main purpose of the wearable sensor device is to provide an on-site solution for capturing and categorizing sound parameters in a construction site. In addition, other measurements were also included such as temperature, geolocation, humidity, and pressure. This was accomplished in a low power, small footprint, and completely wireless sensor hub that constantly uploads the required data to an online server for machine learning and further processing.

A low power microcontroller unit (MCU) or single board computer (SBC) interfacing with an environmental sensor and a sound sensor array was used for monitoring the aforementioned parameters. With a Wi-Fi-enabled system, this device will constantly stream data and periodically send audio files to an online cloud database, with period and sampling rate adjustable by the user. On-board processing includes calculating:

- raw noise
- average noise
- differential noise pressure
- temperature
- altitude
- relative pressure

Geolocation reporting and calculating was achieved using an additional GPS antenna or using a geolocation service with the included Wi-Fi card. An initial prototype has been developed with close to full functionality as described in the primary objective. A second iteration will address the current challenges and complete all project requirements. Our current prototype has been fully developed and has the capabilities to stream all data and locally process noise for additional parameters. The device has been packaged in a (88 x 55 x 30 mm^3) custom case, along with two extruding microphones and an environmental sensor. It also has space for a rechargeable li-ion battery and a micro-USB connection for power or reprogramming as shown in Figure 7.

Figure 7. Pictures of the developed prototype device with 3D-printed housing. The size is 88 x 55 x 30 mm^3.

A NodeMCU ESP32-S MCU is being utilized to capture and process all aforementioned data. Furthermore, it is connected to our established Azure Account and constantly streams all required data as periodic messages. For testing purposes, it also prints all measurements to a serial port for on-site troubleshooting. The device can be reprogrammable with Arduino IDE through a USB connection. Figure 8 shows the individual components of the device.

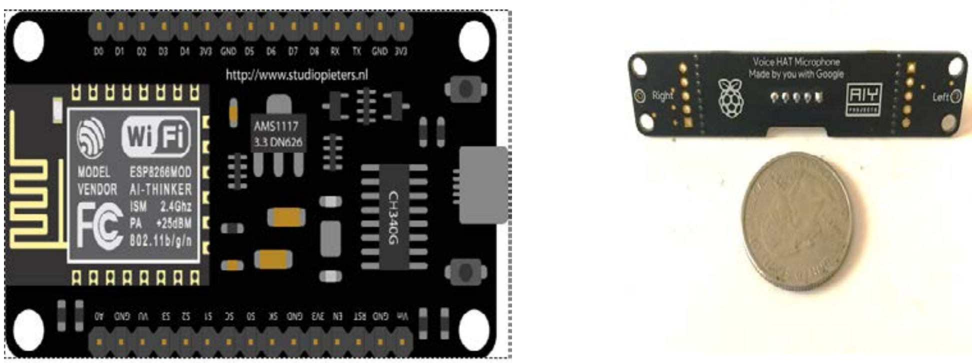


Figure 8. Microcontroller unit with an integrated on-board Wi-Fi transmitter.

Dual MEMS microphones (labelled right and left) are installed that communicate to the MCU through I2S protocol, as the microphones send a digital signal to the board, instead of an analog noise signal. This will further save time and computer resources, as no analog to digital conversion is required prior to sending data. It is important to have dual microphones in order to calculate additional parameters from noise, including average and differential noise pressure. A simple environmental sensor has been installed at the bottom of the case, based on a BME280 temperature, temperature, and humidity sensor. It communicates with the MCU with I2C protocol, sending updated data at a high frequency, as required. Figure 9 shows the environmental sensor module.

Figure 9. Environmental Sensor Module.

4.4. Phase Four: Integrating sensor data and cloud-based computing platform

In this study, all sensor devices were programmed to relay data almost in real-time. The wearable device is programmed to send a new set of telemetry data at an interval of 2 minutes. Considering that construction work takes place for at least 8 hours in each shift, the wearable device would be sending a total of 240 datasets in each shift. Thus, considering the entire time period required for completing the project, the total amount of data collected for each project would be huge and difficult to handle manually on a local computer. In addition, this system's goal is to enable project managers to remotely monitor and handle multiple projects simultaneously, thereby increasing the amount of data coming in and requiring storage to many folds. Therefore, one of the major challenge in this study was to handle the immense amount of data that was being captured by the sensors. Due to this reason, the researchers decided to incorporate a cloud-based computing platform for storing, handling and analyzing the obtained data.

The Microsoft Azure cloud computing platform was chosen for this study as the platform was sufficiently user friendly, suitably compatible with the wearable sensor that was developed in this study, and provided all the back-end resources and services that were required for establishing the system. The researchers did not have to explicitly code the system but were able to utilize the services that were provided by the Azure platform. The researchers mostly wrote source codes for configuring and connecting the different services required to meet the goals and objectives of the monitoring system.

The wearable sensor is capable of capturing all data and sending to Azure IOT Hub through a standard JSON formatted message. Furthermore, the Azure server has been programmed to import the message and translate it to a processing data array to be stored in its SQL database for further analysis. A sample of sent data and received SQL ready array can be seen in Figure 10 which shows how headers and labels are translated into a table format dataset.

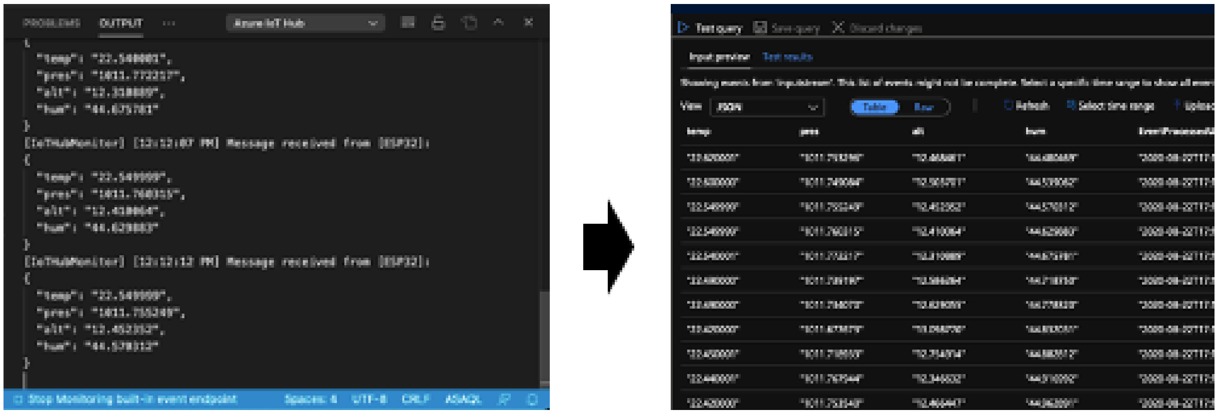


Figure 10. Sample data transmission to data hub.

All of this information is being calculated in real time and streamed wirelessly to be accessed from any site. We system has been set it to a 5 second period for testing, but the time between messages can be reduced at the user's discretion.

4.5. Phase Five: Setting up the cloud-based platform

In this study we incorporated Microsoft Azure as the main cloud-based platform for data collection, storage, computation, and visualization. It is a cloud computing service that allows for building, testing, deploying, and managing applications and services through Microsoft-managed data centers. It provides software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS), and supports many different programming languages, tools and frameworks including third party software and systems.

The data from the sensor was streamed directly into the Azure IoT hub using Raspberry Pi. Raspberry Pi is one of the IoT devices that are compatible with the Azure platform in a way that data can be directly streamed into the Platform as messages once the connection with the device has been established. The sensor was configured to send telemetry data at an interval of 5s. Figure 11 shows the interface of the Azure IoT Hub with the sensor device connected to the platform.

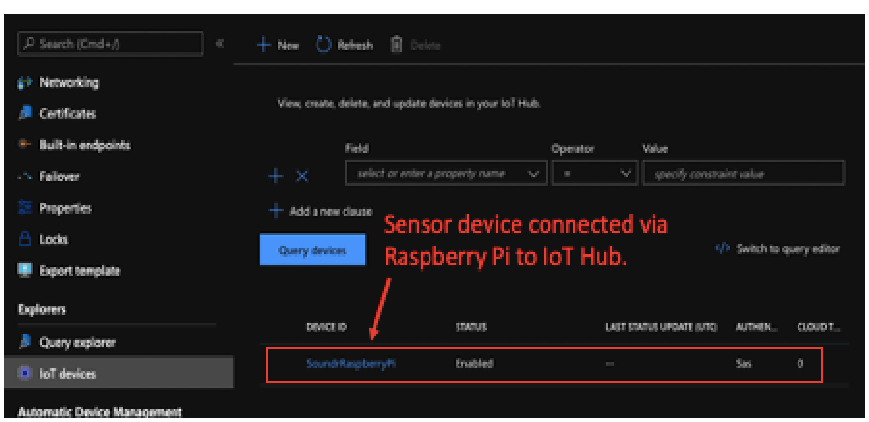


Figure 11. Sensor device connected via Raspberry Pi to the Azure IoT hub.

The Azure IoT Hub collects telemetry data in the form of messages and routes them to the Azure SQL Database using the IoT Stream Analytics Job functionality in Azure. The Azure Stream Analytics job consists of an input, query, and output. The data is ingested from the telemetry messages received from the Azure IoT Hub, queried by the IoT Stream Analytics job and stored into the SQL Database. Figure 12 shows the IoT Stream Analytics interface with the input, output and query, and Figure 13 shows a sample of the sensor data stored in the SQL database.

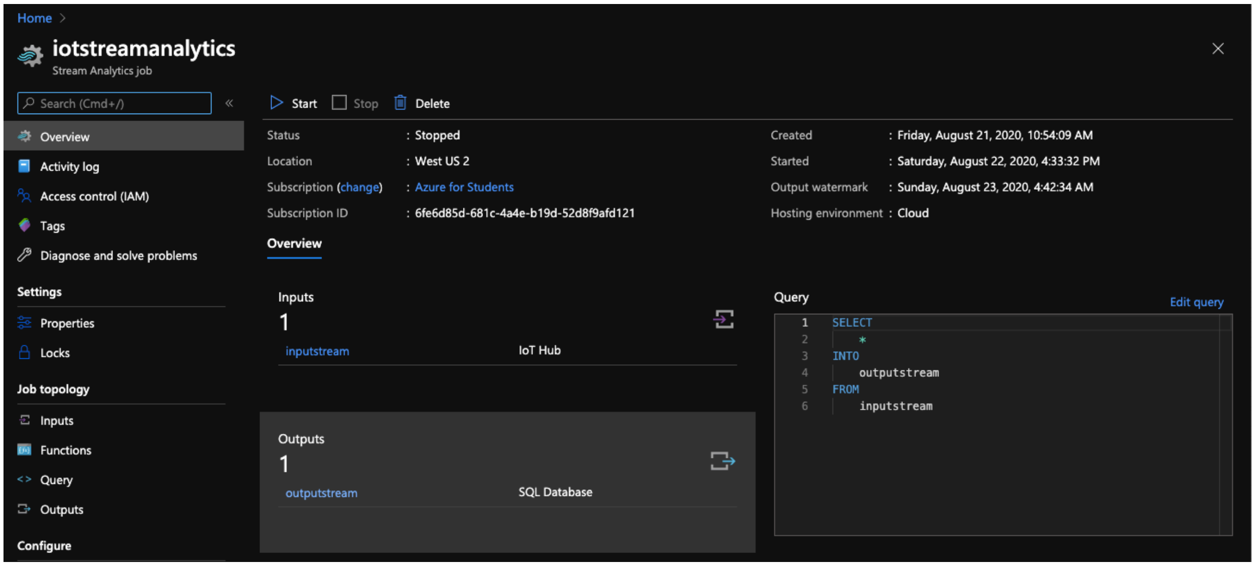
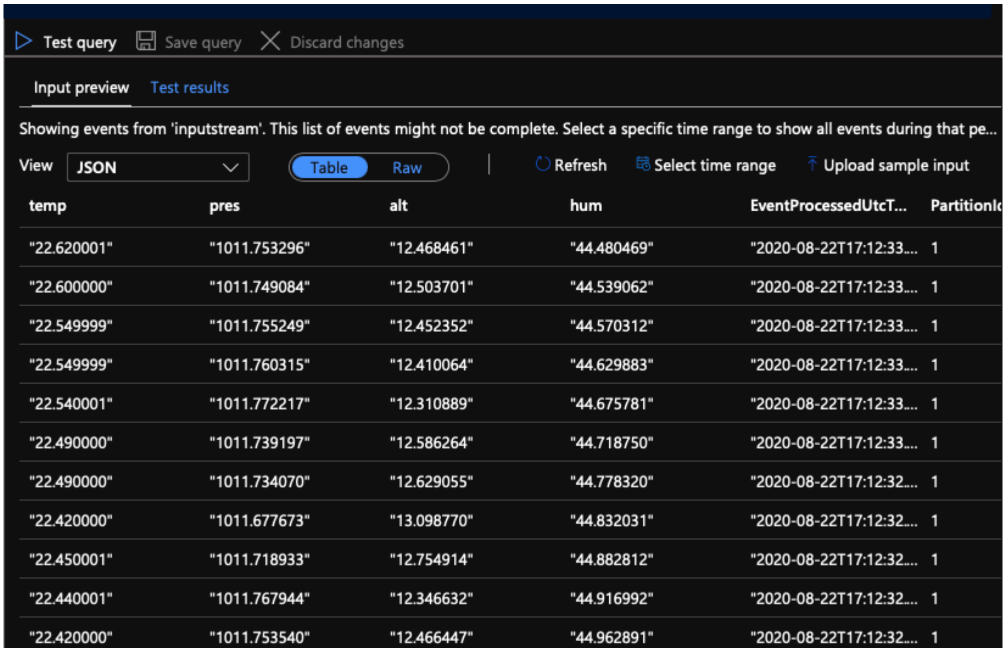


Figure 12. IoT Stream Analytics interface showing input, output, and query.



temp	pres	alt	hum	EventProcessedUtcT...	PartitionI
"22.620001"	"1011.753296"	"12.468461"	"44.480469"	"2020-08-22T17:12:33....	1
"22.600000"	"1011.749084"	"12.503701"	"44.539062"	"2020-08-22T17:12:33....	1
"22.549999"	"1011.755249"	"12.452352"	"44.570312"	"2020-08-22T17:12:33....	1
"22.549999"	"1011.760315"	"12.410064"	"44.629883"	"2020-08-22T17:12:33....	1
"22.540001"	"1011.772217"	"12.310889"	"44.675781"	"2020-08-22T17:12:33....	1
"22.490000"	"1011.739197"	"12.586264"	"44.718750"	"2020-08-22T17:12:33....	1
"22.490000"	"1011.734070"	"12.629055"	"44.778320"	"2020-08-22T17:12:32....	1
"22.420000"	"1011.677673"	"13.098770"	"44.832031"	"2020-08-22T17:12:32....	1
"22.450001"	"1011.718933"	"12.754914"	"44.882812"	"2020-08-22T17:12:32....	1
"22.440001"	"1011.767944"	"12.346632"	"44.916992"	"2020-08-22T17:12:32....	1
"22.420000"	"1011.753540"	"12.466447"	"44.962891"	"2020-08-22T17:12:32....	1

Figure 13. A sample of the telemetry data stored in the SQL database.

Once the telemetry data gets stored in the database, it can be analyzed using Azure Functions. An Azure Function is an event driven, compute-on-demand experience that extends the existing Azure application platform with capabilities to implement code triggered by events occurring in Azure or third-party service as well as on-premises systems. It allows developers to take action by connecting data sources or messaging solutions thus making it easy to process and react to events. In this study the Azure function is used as a backend computation unit for hosting the trained DNN model and conducting the activity identification on the raw sound file that is streamed by the sensor device and stored in the database. It is triggered by a new entry of telemetry data into the database. Once triggered it pulls the raw sound file from the database, analyses the file based on the trained model that is already stored on the Azure platform and saves the output back into the database for the corresponding sensor device. Figure 14 shows the output of the Azure Function stored in a table specific to the sensor in the SQL database.

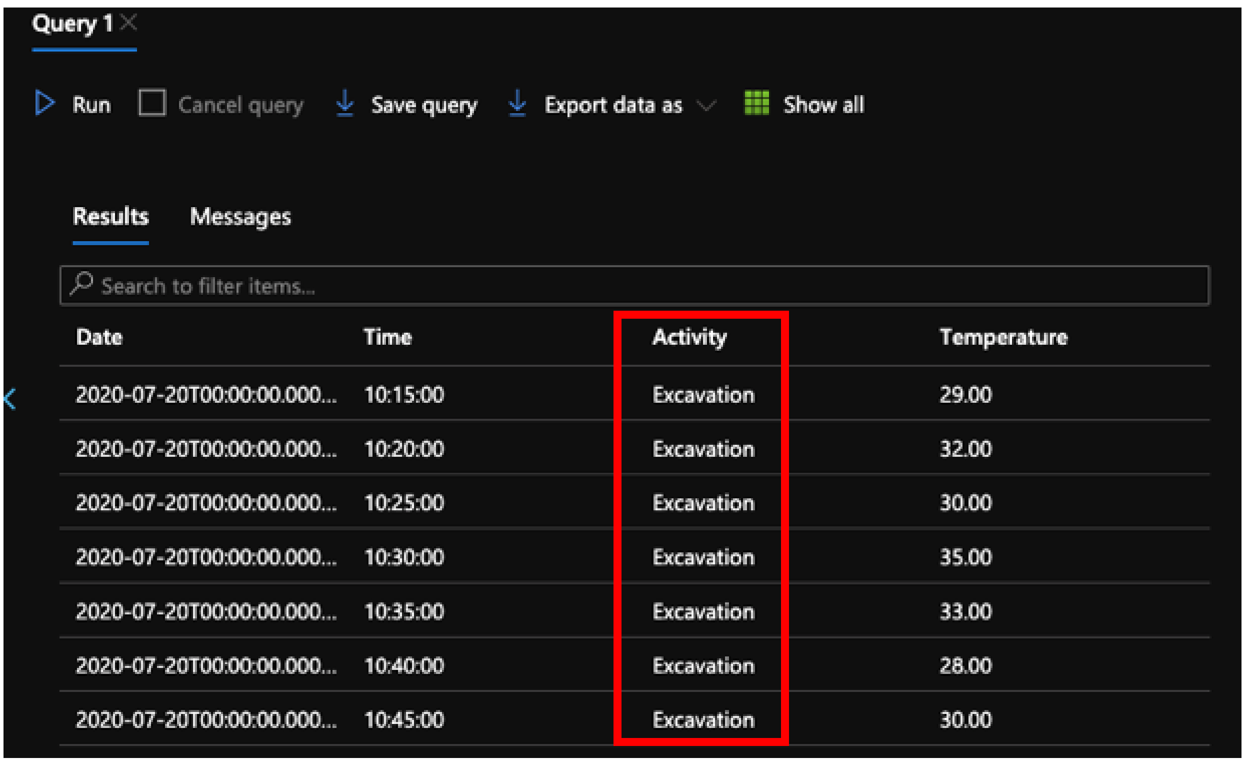


Figure 14. Snapshot of output from Azure Function stored in the SQL database.

4.6. Phase Six: Developing a web-based application for visualization

In this study, a web application was developed and hosted on Azure for providing a user interface for visualizing. The platform allows for the following visualizations:

4.6.1. The location of each sensor on a geographical world map

Since the sensors deployed in this study are wearable sensors, it is expected that the position of the sensors will keep changing as the project progresses. Thus, the sensors have been installed with GPS technology so as to enable their tracking during the entire construction phase. The location tracking in transportation construction projects will also enable supervisors to ensure whether the project is on schedule. Since the transportation projects are horizontal and linear, the movement of the sensors along the project path as the project progresses can be a useful method for detecting project status. The web-application allows the pointers which depict the location of the sensors to continuously update their position on the map as per the GPS coordinated received in the telemetry message from the sensors. Every time the website is refreshed the location of the pointers will shift to the new GPS coordinates received from the sensor device. Figure 15 shows the interface of the web-application developed for the users.

Figure 15. The user-interface of the web-application.

4.6.2. Analyzed telemetry data from the sensors

The web-application is developed in a way that it can pull any set of data from the database, both raw data from the telemetry messages and analyzed data relevant to each sensor. It can display the data using visual analytics in a way suitable for users to quickly interpret meaningful information. For example, it tabulates the weather conditions like temperature data into a line graph. The supervisor can easily look at the graph and determine whether the climate conditions are safe enough for workers to continue the job or not. Figure 16 shows the way the data connected to the sensor device will be displayed on the application interface. The source code of the web-based user interface application is provided in Appendix B.

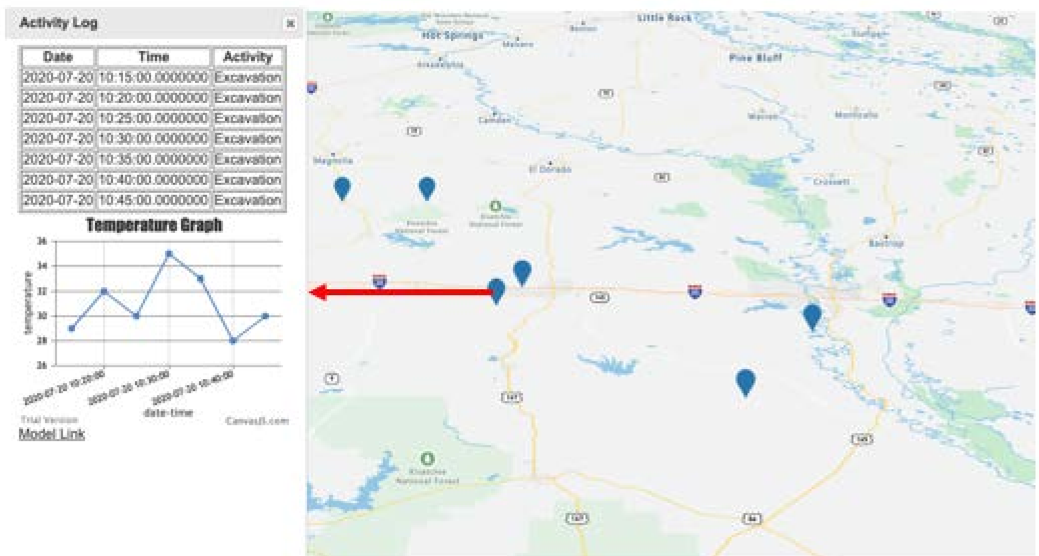


Figure 16. Sensor displaying data on the user interface.

4.6.3. Project Model

In this study, we used Autodesk Forge and Postman for hosting the 3D BIM model of the project. Autodesk Forge is a cloud-based platform that allows the integration 3D models into web applications, whereas Postman is a web-based collaborative platform used for developing Application Programming Interface (API). The Forge application allows for visualizing the model from various angles and different levels of zooming. It also allows for sectional analysis, walkthroughs, taking measurements, and viewing properties of various items in the model. Once the 3D BIM model visualization application is created on Forge, it is hosted as a separate web application on Azure. The link of this new web application for the 3D model is then bootstrapped into the original web-application that is developed as the user interface. Only a section of the project model that is relevant to the current location of the sensor is hosted in the application. Figure 17 shows the way in which the model viewing interface is accessed. The link appears as ‘Model Link’ in the dialog box along with activity log and temperature graph. This is a clickable link which when clicked displays the 3D model of the project on a separate screen. The source code of the web-application for hosting the 3D BIM model is provided in Appendix C.

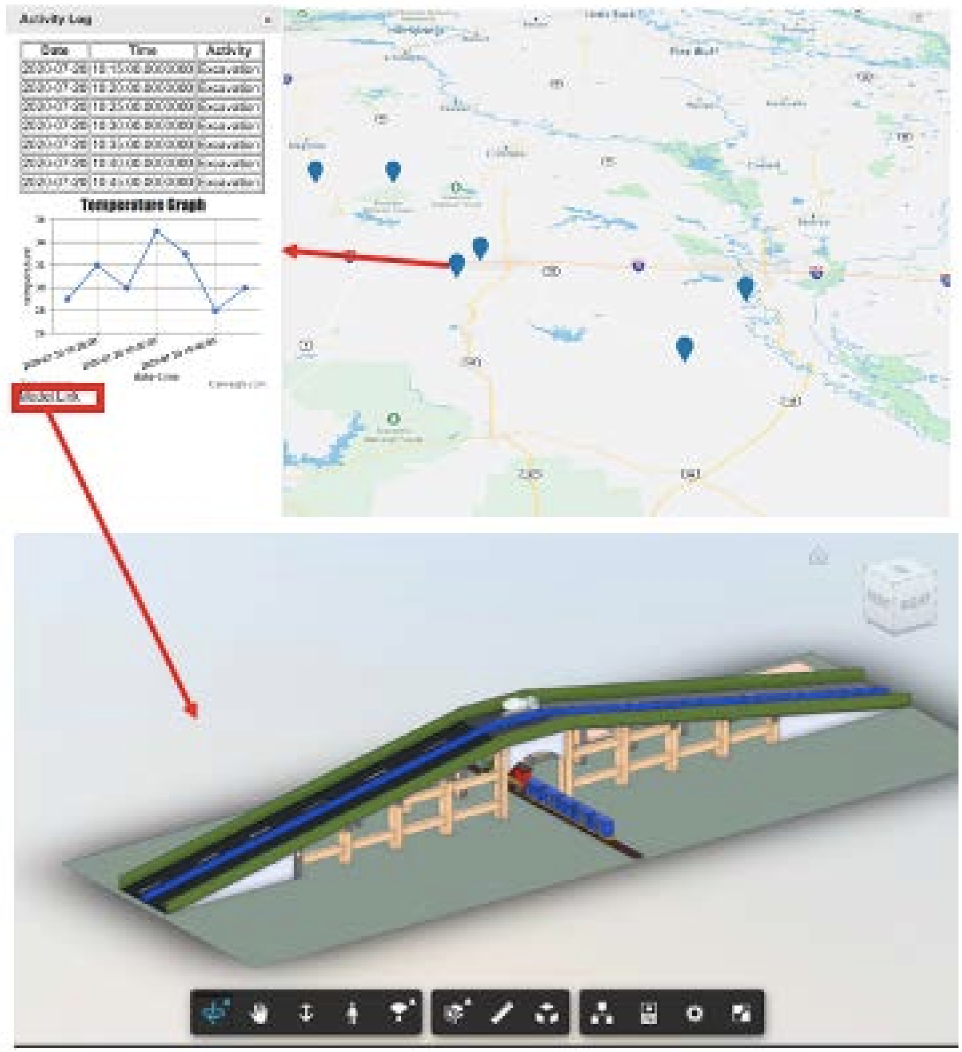


Figure 17. Accessing the model viewing interface.

5. ANALYSIS AND FINDINGS

In this study, we used the DNN algorithm for training the model for activity identification. The sound library was developed as per the project schedule and was then used for training and testing the model. Using a specific schedule enhanced the accuracy of the model. However, the model is capable of handling any number of equipment and machinery included in the schedule. The DNN model when tested in laboratory environment produced over 90% accuracy. The model was also tested with new sound samples and it almost always identified the activity correctly. The algorithm used for training the model is technology advanced and does not require a large number of epochs for training the model. However, though the accuracy and efficiency of the DNN model has been excellent in the laboratory setting, the same cannot be claimed in the real world. Since, construction sites are dynamic in nature and multiple activities take place simultaneously, it is possible that in real world the sound sensors may capture sound from multiple equipment or machinery at the same time or may capture background noise along with the equipment sound, making classification difficult for the DNN algorithm.

The deep neural network model also has its own limitations. One of the major limitations is its requirement of a huge data set for training the model. Since construction sites are extremely dynamic where multiple activities often take place simultaneously, it was significantly difficult to build a sound data library of adequate size for the DNN model. The process of data collection and cleaning was not only tedious but time-consuming. It was also observed during the data cleaning process that when the model was trained with sound data that was perfectly free of background noises from other equipment, workers talking on site, idle time, etc., the accuracy of the model improved significantly. Since the sound library had to be thoroughly cleaned manually, this made the job of collecting and building an accurate sound data library for the model even more difficult. Furthermore, the sound produced by each equipment and machinery is different. In fact, the sound produced by the equipment or machinery changes with different brands of the same equipment or machinery. The sound also considerably changes when used with different material and in different environments. As such, the same sound library cannot be used for activity identification and progress monitoring for all projects. A new sound library needs to be created each time for every new project which is specific to the project regarding the equipment and machinery used, the materials used and the construction environment. In addition, a construction site is susceptible to changes which may lead to the addition of new activities and thereby new classes of sound in the library later in the project. The present framework is not robust to any change of expected activities in a construction site. The classifier needs to be regenerated and the entire system reconfigured to handle such situations.

The current prototype of the wearable sensors developed for this project also has certain limitations which include no on-board storage and no dedicated GPS antenna. Furthermore, we are only able to change or modify the code using a wired USB connection with a computer. The next iteration will include the following key features:

- On board storage to store audio clips and other data to be sent over time.
- Complete remote access to a desktop environment on the board in order to modify code and fetch any on-board data for troubleshooting.

- A dedicated GPS antenna for greatly increased accuracy of geolocation.
- GPS antenna board also includes an accelerometer for additional environmental parameters to be reported, if necessary.

Currently there are also certain limitations of the cloud-based web-application developed as the user-interface which are as follows:

- The IP address of the system from which the website or user-interface (https://sound-app.azurewebsites.net/) is accessed must be added to the firewall settings on the Microsoft Azure platform.
- Every time the 3D model-view platform is accessed, a new access token needs to be generated using the Postman platform and replaced with the old token in the source code editor for the 3D model web-application on Microsoft Azure.

6. CONCLUSIONS

This research aims to establish a prototype system for autonomous monitoring and surveillance of transportation construction and maintenance projects using low-cost and light-weight wearable sensors that are capable of capturing sound data as well as other information such as GPS location, altitude, climatic conditions like temperature, pressure, humidity, etc. The purpose of this research is to enable DOT practitioners and project managers to remotely monitor multiple project sites and access real-time data that can help gain important insights regarding the project's progress, status, working conditions and emergency and make timely decisions so as to keep the project on-track. A DNN algorithm was adopted for sound classification and construction activity identification. Based on this approach, the types of dynamic training dataset have been considerably restricted to only planned types of sound data. The model produced more than satisfactory results under laboratory settings where accuracy and efficiency was concerned, as is evident from the results. However, the same may not be true in real-world settings. In addition, the DNN model also has few limitations and difficulties related to building the required sound library.

The research outcome involves new knowledge on the implications of sound recognition in the transportation construction sector for real-time monitoring and forecasting of work progress and task performance without any human effort. In addition, the new framework has significant impacts on workflow optimization, work performance monitoring, and the cost/schedule overrun mitigation of diverse construction projects for roads, highways, bridges, and tunnels. It is expected that the developed theoretical framework for detecting construction activities through building the schedule-based dynamic sound data training, can be possibly extended in other areas having the potentiality of adopting acoustic data analysis and recognition, such as fault detection, and unmanned manufacturing. Automated monitoring is one of the most promising methods for accurate and continuous monitoring of safety performance on a construction site. Since diverse safety risk and unexpected urgent situations are common in a vulnerable construction site, any framework for avoiding severe accidents should be considered for leveraging strict safety monitoring and supporting quick safety measures in real construction practice.

In addition, the project also provides a user-friendly interface for visualizing all project related data streamed to a cloud-based computing platform in real-time. The raw data is processed in the cloud computing platform and the results are displayed using visual analytics to enable supervisors and project managers to easily identify issues and use the information for timely decision making. The interface also allows for viewing the BIM model of the project where the supervisors can visually compare the scheduled progress of the project with the actual progress. It can be expected that the developed theoretical framework will add the innovative scientific knowledge and the new logical theory which pose significant impacts on the workflow optimization and the accurate task-performance measurement of civil infrastructure construction projects for roads, highways, bridges, and tunnels. The future course of action for this project is to improve the limitations of the sensor device and the visualization platform. For future projects, the research team will strive to achieve the following:

- To develop the framework to make it robust to any changes in a construction schedule and update its classifier to adjust the changes rather than regenerating the classifier and reconfiguring the system.
- Ensure high accuracy of model in real construction-sites as well.
- Recognition of sounds that do not belong to any of the categories such as the recognition of idle time.
- Integrate more types of data from the construction site that can be streamed in real-time and analyzed to provide useful insight.
- Improve the efficiency of the entire prototype system.

REFERENCES

1. Melzner, J., S. Zhang, J. Teizer and H. Bargstädt. A Case Study on Automated Safety Compliance Checking to Assist Fall Protection Design and Planning in Building Information Models. *Construction Management and Economics*, 2013. 31(6):661-674.
2. The US Bureaus of Labor Statistics. News Release Bureau of Labor Statistics. *The National Census of Fatal Occupational Injuries Survey*, 2015.
3. Naikander, I.O. and E. Eloranta. Preliminary Signals and Early Warnings in Industrial Investment Projects. *International Journal of Project Management*, 1997. 15(6):371-376.
4. Zhang, J.P. and Z.Z. Hu. BIM and 4D-based Integrated Solution of Analysis and Management for Conflicts and Structural Safety Problems during Construction: Principles and Methodologies. *Automation in Construction,* 2011. 20(2):155-166.
5. Cheung, W.F., T. H. Lin, and Y. C. Lin. A Real-Time Construction Safety Monitoring System for Hazardous Gas Integrating Wireless Sensor Network and Building Information Modeling Technologies. *Sensors*, 2018. 18(2):436.
6. Park, J., K. Kim and Y. K. Cho. Framework of Automated Construction-Safety Monitoring Using Cloud-Enabled BIM and BLE Mobile Tracking Sensors. *Journal of Construction Engineering and Management,* 2016. 143(2).
7. Brilakis, I., H. Fathi, and A. Rashidi. Progressive 3D Reconstruction of Infrastructure With Videogrammetry. *Automation in Construction*, 2011. 20(7):884-895.
8. Golparvar-Fard, M., F. Peña-Mora, and S. Savarese. D4AR–a 4-Dimensional Augmented Reality Model for Automating Construction Progress Monitoring Data Collection, Processing and Communication. *Journal of Information Technology in Construction*, 2009. 14(13):129-153.
9. Park, J.W., E. Marks, Y.K. Cho, and W. Suryanto. Performance Test of Wireless Technologies for Personnel and Equipment Proximity Sensing in Work Zones. *Journal of Construction Engineering and Management, ASCE,* 2015. 1(142).
10. Turkan, Y., F. Bosche, C.T. Haas, and R. Haas. Automated Progress Tracking Using 4D Schedule and 3D Sensing Technologies. Automation in Construction, 2012. 22: 414-421.
11. Teizer, J., D. Lao, and M. Sofer. Rapid Automated Monitoring of Construction Site Activities Using Ultra-Wideband. *Proceedings of the 24th International Symposium on Automation and Robotics in Construction, Kochi, Kerala, India,* 2007. pp. 19-21.
12. Cheng, T. and J. Teizer. Real-time Resource Location Data Collection and Visualization Technology for Construction Safety and Activity Monitoring Applications. *Automation in Construction*, 2013. 34: 3-15.
13. Cheng, C. F., A. Rashidi, M.A. Davenport, and D.V. Anderson. Activity Analysis of Construction Equipment Using Audio Signals and Support Vector Machines. *Automation in Construction,* 2016. 81:240-253.
14. Sharan, R.V. and T.J. Moir. An Overview of Applications and Advancements in Automatic Sound Recognition. *Neurocomputing,* 2016. 200:22-34.
15. Park, J., K. Kim, and Y.K. Cho. Framework of Automated Construction-Safety Monitoring Using Cloud Enabled BIM and BLE Mobile Tracking Sensors. *Journal of Construction Engineering and Management,* 2016. 143(2).

16. Almadeed, N., M. Asim, S. Al-Maadeed, A. Bouridane, and A. Beghdadi. Automation Detection and Classification of Audio Events for Road Surveillance Applications. *Sensors,* 2018. 18(6):1858.
17. Xu, X., H. Gao, J. Yu, Y. Chen, Y. Zhu, G. Xue, and M. Li. ER: Early Recognition of Inattentive Driving Leveraging Audio Devices on Smartphones. *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, 2017.
18. Kubera, E., A. Wieczorkowska, T. Slowik, A. Kuranc, and K. Skrzypiec. Audio-based Speed Change Classification for Vehicles. *International Workshop on New Frontiers in Mining Complex Patterns*, 2016.
19. Rani, U.J. and T. H. Sreenivas. Remote Vehicle Tracking System through Voice Recognition App Using Smart Phone. *International Journal of Science, Engineering and Computer Technology,* 2015. 5(6):200.
20. Peng, Y. T., C.Y. Lin, M.T. Sun, and K.C. Tsai. Healthcare Audio Event Classification Using Hidden Markov Models and Hierarchical Hidden Markov Models. IEEE International Conference, 2009. pp. 1218-1221.
21. Doukas, C., L. Athanasiou, K. Fakos, and I. Maglogiannis. Advanced Sound and Distress Speech Expression Classification for Human Status Awareness in Assistive Environments. *The Journal on Information Technology in Healthcare,* 2009. 7(2): 111-117.
22. Sunkpho, J., J. J. H. Garrett, and A. Smailagic. Opportunities to Use Speech Recognition for Bridge Inspection. *Construction Congress VI: Building Together for a Better Tomorrow in an Increasingly Complex World*, 2000.
23. Cheng, C. F., A. Rashidi, M.A. Davenport, and D. Anderson. Audio Signal Processing for Activity Recognition of Construction Heavy Equipment. *Proceedings of the International Symposium on Automation and Robotics in Construction,* 2016. 33:1.
24. Cho, C., Y.C. Lee, and T. Zhang. Sound Recognition Techniques for Multi-Layered Construction Activities and Events. *Computing in Civil Engineering* 2017. pp. 326-334.
25. Choi, I., K. Kwon, Bae, S.H., and N.S. Kim. DNN-based Sound Event Detection with Exemplar-based Approach for Noise Reduction. *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop,* 2016. pp 16-19.
26. Chen, T.E., S.I. Yang, L.T. Ho, K.H. Tsai, Y.H. Chen, Y.F. Chang, Y.H. Lai, S.S. Wang, Y. Tsao, and C.C. Wu. S1 and S2 Heart Sound Recognition Using Deep Neural Networks." *IEEE Transactions on Biomedical Engineering,* 2017. 64(2):372-380.
27. Sigtia, S., A.M. Stak, S. Krstulovic, and M.D. Plumbley. Automatic Environmental Sound Recognition: Performance vs Computational Cost. *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* 2016. 4(11): 2096-2107.
28. Hayashi, T., M. Nishida, N. Kitaoka, and K. Takeda. Daily Activity Recognition Based on DNN Using Environmental Sound and Acceleration Signals." *23rd European Signal Processing Conference (EUSIPCO)*, 2015.
29. Akhavian, R. and A.H. Behzadan. Construction Equipment Activity Recognition for Simulation Input Modeling using Mobile Sensors and Machine Learning Classifiers. *Advanced Engineering Informatics*, 2015. 29:867-877.

30. Akhavian, R. and A.H. Behzadan. Smartphone-based Construction Workers' Activity Recognition and Classification. *Automation in Construction*, 2016. 71:198-209
31. Hinton, G. E., L. Deng, D. Yu, G. Dahl, A.R. Mohamed, N. Jaitly, V. Vanhoucke, P. Nguyen, B. Kingsbury, and T. Sainath. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine*, 2012. 29:82-97.
32. Krizhevsky, A., I. Sutskever, and G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 2012. pp. 1097-1105.
33. Gencoglu, O., T. Virtanen, and H. Huttunen. Recognition of Acoustic Events Using Deep Neural Networks. *Signal Processing Conference (EUSIPCO), Proceedings of the 22nd European. IEEE*, 2014. pp. 506–510.

APPENDIX A. SOURCE CODE OF DNN MODEL

A.1. Model Training

```
import glob
import os
import librosa
import librosa.display
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import specgram
# %matplotlib inline
plt.style.use('ggplot')
plt.rcParams['font.family'] = 'DejaVu Sans'
plt.rcParams['font.serif'] = 'Ubuntu'
plt.rcParams['font.monospace'] = 'Ubuntu Mono'
plt.rcParams['font.size'] = 12
plt.rcParams['axes.labelsize'] = 11
plt.rcParams['axes.labelweight'] = 'bold'
plt.rcParams['axes.titlesize'] = 14
plt.rcParams['xtick.labelsize'] = 10
plt.rcParams['ytick.labelsize'] = 10
plt.rcParams['legend.fontsize'] = 11
plt.rcParams['figure.titlesize'] = 13
def extract_feature(file_name):
    X, sample_rate = librosa.load(file_name) #loading data
    stft = np.abs(librosa.stft(X))
    mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T,axis=0)
    chroma = np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T,axis=0)
    mel = np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)
    contrast = np.mean(librosa.feature.spectral_contrast(S=stft, sr=sample_rate).T,axis=0)
```

```
    tonnetz = np.mean(librosa.feature.tonnetz(y=librosa.effects.harmonic(X), sr=sample_rate).T,axis=0)
    zero_cross = np.array(librosa.feature.zero_crossing_rate(X).T)
    zero_cross.shape = (zero_cross.shape[0],)
    rms = np.array(librosa.feature.rms(y=X).T)
    rms.shape = (rms.shape[0],)
    return mfccs,chroma,mel,contrast,tonnetz,zero_cross,rms

def parse_audio_files(parent_dir,sub_dirs,file_ext='*.wav'):
    features, labels = np.empty((0,367)), np.empty(0)
    for label, sub_dir in enumerate(sub_dirs):
        for fn in glob.glob(os.path.join(parent_dir, sub_dir, file_ext)):
            mfccs, chroma, mel, contrast,tonnetz,zero_cross,rms = extract_feature(fn)
            ext_features = np.hstack([mfccs,chroma,mel,contrast,tonnetz,zero_cross,rms])
            features = np.vstack([features,ext_features]
 labels = np.append(labels, fn.split('\\')[2].split('_')[2])
     return np.array(features), np.array(labels, dtype = np.int)

def one_hot_encode(labels):
    n_labels = len(labels)
    n_unique_labels = len(np.unique(labels))
    one_hot_encode = np.zeros((n_labels,n_unique_labels))
    one_hot_encode[np.arange(n_labels), labels] = 1
    return one_hot_encode
parent_dir = 'Highway-Construction'

sub_dir = ['Excavator1','ConcreteMix2','Compact3','Bulldoze4','Dumper5','Dragline6','Vibrator7','Crusher8','Shovel9','Roller10','Grader11','Distributor12','Paver13','Scraper14']
features, labels = parse_audio_files(parent_dir,sub_dir)
```

```
labels = one_hot_encode(labels)
train_test_split = np.random.rand(len(features)) < 0.70
train_x = features[train_test_split]
train_y = labels[train_test_split]
test_x = features[~train_test_split]
test_y = labels[~train_test_split]
import tensorflow as tf
from sklearn.metrics import precision_recall_fscore_support
from tensorflow.keras import Sequential,Model
from tensorflow.keras.layers import Dense
l=[]
n_dim = features.shape[1]
loss_history = np.empty(shape=[1],dtype=float)
n_classes = 14
training_epochs=50
inputX=tf.keras.Input(shape=(None,n_dim))
inputY=tf.keras.Input(shape=(None,n_classes))
W=tf.Variable(tf.ones(n_dim))
model=Sequential()
model.add(Dense(100,activation='tanh',kernel_initializer='he_normal',input_shape=(n_dim,)))
model.add(Dense(100,activation='sigmoid',kernel_initializer='he_normal'))
model.add(Dense(4,activation='softmax'))
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
history=model.fit(features,labels,batch_size=32,epochs=training_epochs)
(loss,accuracy)=history.history.values()
results=model.evaluate(test_x,test_y,verbose=0)
saved_location=model.save('saved_model.h5')
y_1=model.predict(test_x)
y_pred=tf.argmax(y_1,1)
```

```
y_true=tf.argmax(test_y,1)
fig = plt.figure(figsize=(10,8))
plt.ylabel("Cost")
plt.xlabel("Epoch")
plt.axis([0,training_epochs,0,np.max(loss_history)])
plt.savefig("/Users/aderia/Desktop/AMF/Fig6.png")
p,r,f,s = precision_recall_fscore_support(y_true, y_pred, average='micro')
print("F-Score:", round(f,13))
```

A.2. Model Testing

```
import tensorflow as tf
import librosa
import numpy as np
from tensorflow import keras
def extract_feature(file_name):
    X, sample_rate = librosa.load(file_name) #loading data
    stft = np.abs(librosa.stft(X))
    mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T,axis=0)
    chroma = np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T,axis=0)
    mel = np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)
    contrast = np.mean(librosa.feature.spectral_contrast(S=stft, sr=sample_rate).T,axis=0)
    tonnetz = np.mean(librosa.feature.tonnetz(y=librosa.effects.harmonic(X),
    sr=sample_rate).T,axis=0)
    zero_cross = np.array(librosa.feature.zero_crossing_rate(X).T)
    zero_cross.shape = (zero_cross.shape[0],)
    rms = np.array(librosa.feature.rms(y=X).T)
    rms.shape = (rms.shape[0],)
    return mfccs,chroma,mel,contrast,tonnetz,zero_cross,rms
def parse_audio_files(file_name,label1):
```

```
    features, labels = np.empty((0,367)), np.empty(0)
    mfccs, chroma, mel, contrast,tonnetz,zero_cross,rms = extract_feature(file_name)
    ext_features = np.hstack([mfccs,chroma,mel,contrast,tonnetz,zero_cross,rms])
    features = np.vstack([features,ext_features])
    for i,lb in enumerate(label1):
        labels = np.append(labels,i)
    return np.array(features), np.array(labels, dtype = np.int)
file_name='03_Paver_2_2.wav'
label1=['Excavator','ConcreteMix','Compact','Bulldoze','Dumper','Dragline','Vibrator','Crusher','Shovel','Roller','Grader','Distributor','Paver','Scraper']
features, labels = parse_audio_files(file_name,label1)
test_input4=features
model = keras.models.load_model('C:/Users/aderia2/Desktop/DNN4/saved_model.h5')
results=model.predict(test_input4)
res=np.asarray(results)
final_result=np.max(res)
print(results)
print(final_result)
for x in range (0,14):
    if results[0][x]==final_result:
        ans=x
print (label1[ans])
```

APPENDIX B. SOURCE CODE FOR WEB-BASED USER INTERFACE

```
<!DOCTYPE html>

<?php

// PHP Data Objects(PDO) Sample Code:

// SQL Server Extension Sample Code:

try {

        $conn = new PDO("sqlsrv:server = tcp:cicilab.database.windows.net,1433; Database = Soundsensors", "cicilab", "Cicipw2020");

        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    }

    catch (PDOException $e) {

        print("Error connecting to SQL Server.");

        die(print_r($e));

    }

            // SQL Server Extension Sample Code:

    $connectionInfo = array("UID" => "cicilab", "pwd" => "Cicipw2020", "Database" => "Soundsensors", "LoginTimeout" => 30, "Encrypt" => 1, "TrustServerCertificate" => 0);

    $serverName = "tcp:cicilab.database.windows.net,1433";

    $conn = sqlsrv_connect($serverName, $connectionInfo);

                $tablename1 = "ActivityLog_Sensor1"; // tablename should be ActivityLog_Sensornumber whichever sensor is clicked on the map


                $tablename2 = "ActivityLog_Sensor2";

    $sql = "SELECT * FROM [dbo].[ActivityLog_Sensor1]";

    $sql1 = "SELECT * FROM [dbo].[ActivityLog_Sensor2]";

    $sql2 = "SELECT * FROM [dbo].[SensorData]";

                $options = array('ReturnDatesAsStrings' => true);

                $stmt = sqlsrv_query($conn, $sql,array(),$options);
```

```
$stmt1 = sqlsrv_query($conn, $sql1,array(),$options);
                                        $stmt2 = sqlsrv_query($conn,$sql2);
$Sensor1X=0.0;
$Sensor2X=0.0;
$Sensor1Y=0.0;
$Sensor2Y=0.0;
                                        while($row=sqlsrv_fetch_array($stmt2))
{
   if($row['SensorID']=='1')
   {
      $Sensor1X=(float)$row['Latitude'];
      $Sensor1Y=(float)$row['Longitude'];
   }
   if($row['SensorID']=='2')
   {
      $Sensor2X=(float)$row['Latitude'];
      $Sensor2Y=(float)$row['Longitude'];
   }
}
                                        $year=array();
$month=array();
$day=array();
$hour=array();
$minute=array();
$second=array();
$temperature=array();
                                        $iter2 = (int)0;
?>
<html lang="en">
```

```
<head>
    <title>Accessible popups - Azure Maps Web SDK Samples</title>


    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="IE=Edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
    <meta name="description" content="This sample shows how to use popups in a way that users can easily access them using keyboard shortcuts or on mouse click." />
    <meta name="keywords" content="Microsoft maps, map, gis, API, SDK, popups, infobox, infowindow, events, mouse, accessibility" />
    <meta name="author" content="Microsoft Azure Maps" />
    <link rel="stylesheet" href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
    <!-- Add references to the Azure Maps Map control JavaScript and CSS files. -->
    <link rel="stylesheet" href="https://atlas.microsoft.com/sdk/javascript/mapcontrol/2/atlas.min.css" type="text/css" />
    <style>
        #showChart {
            background-color: #5bb85b;
            color: #ffffff;
            padding: 100px;
            border: 0px;
            border-radius: 8px;
            font-size: 10px;
            outline: none;cursor: pointer;
        }
        #container {
            position: fixed;
            top: 10%;
            left: 20%;
            transform: translate(-100%, -100%);
```

```
            text-align: center;
            margin-top: -41px;
        }
    </style>
    </head>
<body onload='GetMap()'>
    <div id="myMap" style="position:absolute;width:100%;min-width:290px;height:1000px;">
</body>
     <script src="https://atlas.microsoft.com/sdk/javascript/mapcontrol/2/atlas.min.js"></script>
    <script type='text/javascript'>
        var map, popups = [], datasource;
        // Feed that contains the data that we want to map
         var Sensor1X = "<?php echo $Sensor1X;?>";
        var Sensor2X = "<?php echo $Sensor2X;?>";
        var Sensor1Y = "<?php echo $Sensor1Y;?>";
        var Sensor2Y = "<?php echo $Sensor2Y;?>";
            var simple = "<?php echo "<table
border='1'><tr><th>Date</th><th>Time</th><th>Activity</th></tr>";
             while($row=sqlsrv_fetch_array($stmt))
            {
                $temptime = (string)$row['Date'];
                $temptime = $temptime.' '.(string)$row['Time'];
$temptime2 = new DateTime($temptime);
                array_push($year,$temptime2->format('Y'));
                array_push($month,$temptime2->format('m'));
                array_push($day,$temptime2->format('d'));
                array_push($hour,$temptime2->format('H'));
                array_push($minute,$temptime2->format('i'));
                array_push($second,$temptime2->format('s'));
                array_push($temperature,(float)$row['Temperature']);
```

```
                echo "<tr>"
echo ("<td>".$row['Date']."</td>");
                echo ("<td>".$row['Time']."</td>");
                echo ("<td>".$row['Activity']."</td>");
                echo "</tr>";
                }
                echo "</table>";?>";
                        console.log(simple);
        var tempshow;
        var sampleLocationData = [
            {
                position: [-92.701673830, 32.542210000],
                description: simple  //THis is the area for the printing for Sensor 1
            },
              {
                position: [-92.301673830, 32.142210000],
                description: 'Sensor not yet activated.'  //THis is the area for the printing for Sensor 1
            }, {
                position: [-92.7481097, 32.4652978],
                description: 'Sensor not yet activated.' //THis is the area for the printing for Sensor 2
            },
            {
                position: [-92.1481097, 32.3652978],
                description: 'Sensor not yet activated.' //THis is the area for the printing for Sensor 2
            },
            {
                position: [-92.9481097, 32.9652978],
                description: 'Sensor not yet activated.' //THis is the area for the printing for Sensor 2
            },
```

```
{
    position: [-93.1481097, 32.9652978],
    description: 'Sensor not yet activated.' //THis is the area for the printing for Sensor 2
}
];
var Dat = [{ x: new Date("<?php echo $year[0];?>", Number("<?php echo $month[0];?>") - 1, "<?php echo $day[0];?>", "<?php echo $hour[0];?>", "<?php echo $minute[0];?>", "<?php echo $second[0];?>"), y: Number("<?php echo $temperature[0];?>") }]
<?php $len = count($year);?>
<?php for ($iter = 1; $iter < $len; $iter++)
  { ?>
      var New = [{ x: new Date("<?php echo $year[$iter];?>", Number("<?php echo $month[$iter];?>") - 1, "<?php echo $day[$iter];?>", "<?php echo $hour[$iter];?>", "<?php echo $minute[$iter];?>", "<?php echo $second[$iter];?>"), y: Number("<?php echo $temperature[$iter];?>") }];
      Dat = Dat.concat(New);
    <?php
  }
?>
function GetMap() {
  //Initialize a map instance.
  map = new atlas.Map('myMap', {
    center: [-91.96, 30.98],
    zoom: 9,
    pitch: 60,
    view: 'Auto',
//Add your Azure Maps key to the map SDK. Get an Azure Maps key at https://azure.com/maps. NOTE: The primary key should be used as the key.
    authOptions: {
      authType: 'subscriptionKey',
      subscriptionKey: 'tc9F_2FDR9gIZ81rgaho61YyTaVX2Pr4pEy8UnD6B3w'
```

```
        }
      });
      //Wait until the map resources are ready.
      map.events.add('ready', function () {
       //Create a data source and add it to the map.
        datasource = new atlas.source.DataSource();
        map.sources.add(datasource);
        //For each sample location, create a popup, store it in an array.
        //Then create a point feature, add the popups array index properties of the point feature,
then add the add it to the data source.
        for (var i = 0; i < 1; i++) {
          //Create a popup.
          popups.push(new atlas.Popup({
            content: `<div id="dialogBox" style="display:none">
                ${sampleLocationData[i].description}
                <div id="chartContainer" class="dialog" style="height: 250px; width:
100%;">
                </div>
                <div><a href ="https://temperaturegraph.azurewebsites.net" target =
"_blank"> Model Link</a></div>
</div>`,
          }));
          //Create a point feature and store the popups index from the array.
          datasource.add(new atlas.data.Feature(new
atlas.data.Point(sampleLocationData[i].position), {
            popupIdx: popups.length - 1
          }));
        }
        for (var i=1; i<6; i++) {
          //Create a popup.
          popups.push(new atlas.Popup({
```

```
                    content: sampleLocationData[i].description
                }));
//Create a point feature and store the popups index from the array.
                datasource.add(new                                                    atlas.data.Feature(new
atlas.data.Point(sampleLocationData[i].position), {
                    popupIdx: popups.length - 1
                }));
}
            //Add all the popups to the map.
            map.popups.add(popups);
            //Add a symbol layer to display the point features on the map.
            var layer = new atlas.layer.SymbolLayer(datasource);
            map.layers.add(layer);
            //Add a click event the
           // map.events.add('click', layer, showPopup);
map.events.add('click', layer, function () {
        var options =
         {
            animationEnabled: true,
            title: {
                text: "Temperature Graph"
            },
            axisX: {
title: "date-time",
                gridThickness: 0.5,
                interval: 10,
                intervalType: "minute",
                valueFormatString: "YYYY-MM-DD HH:mm:ss",
                labelAngle: -20
            },
```

```
        axisY: {
          title: "temperature"
        },
        data: [
          {
            type: "line",
            dataPoints: Dat,
          }
        ]
  };
        $("#dialogBox").dialog({
          open: function (event, ui) {
$(".ui-widget-overlay").bind("click", function (event, ui) {
              $("#dialogBox").dialog("close");
            });
          },
          closeOnEscape: true,
          resizable: true,
          draggable: true,
          title: "Activity Log",
          width: 350,
          height: 700,
          position: top,
          modal: false
        });
$(".ui-widget-overlay").css({ "background-color": "#111111"});
        $("#chartContainer").CanvasJSChart(options); //use innerhtml code here to develop graphs
              });
      });
```

```
        }
        function showPopup(e) {
            //Get the point feature the click event occured on.
            var point = e.shapes[0];
            //Get the popup for the point feature.
            var popup = popups[point.getProperties().popupIdx];
//Open the popup.
            popup.open(map);
        }
                </script>
        <script src="https://canvasjs.com/assets/script/jquery-1.11.1.min.js"></script>
    <script src="https://canvasjs.com/assets/script/jquery.canvasjs.min.js"></script>
    <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
</html>
```

APPENDIX C. SOURCE CODE FOR 3D-MODEL VIEWING APPLICATION

```
<head>
    <meta name="viewport" content="width=device-width, minimum-scale=1.0, initial-scale=1, user-scalable=no" />
    <meta charset="utf-8">
    <!-- The Viewer CSS -->
    <link rel="stylesheet" href="https://developer.api.autodesk.com/modelderivative/v2/viewers/6.*/style.min.css" type="text/css">
    <!-- Developer CSS -->
    <style>
        body {
            margin: 0;
        }
        #MyViewerDiv {
            width: 100%;
            height: 100%;
            margin: 0;
            background-color: #F0F8FF;
        }
    </style>
</head>
<body>
    <!-- The Viewer will be instantiated here -->
    <div id="MyViewerDiv"></div>
    <!-- The Viewer JS -->
    <script src="https://developer.api.autodesk.com/modelderivative/v2/viewers/6.*/viewer3D.min.js"></script>
```

```
<!-- Developer JS -->
<script>
    var viewerApp;
    var options = {
        env: 'AutodeskProduction',
        getAccessToken: function(onGetAccessToken) {
            //
            // TODO: Replace static access token string below with call to fetch new token from your backend
            // Both values are provided by Forge's Authentication (OAuth) API.
            //
            // Example Forge's Authentication (OAuth) API return value:
            // {
            //    "access_token": "eyJhbGciOiJIUzI1NiIsImtpZCI6Imp3dF9zeW1tZXRyaWNfa2V5In0.eyJzY29wZSI6WyJkYXRhOnJlYWQiLCJkYXRhOndyaXRlIiwiZGF0YTpjcmVhdGUiLCJidWNrZXQ6cmVhZCIsImJ1Y2tldDpjcmVhdGUiXSwiY2xpZW50X2lkIjoiNG1EbDlqNDNzU3paNWFSaGJxZk95NzV0R2d3T0szV0wiLCJhdWQiOiJodHRwczovL2F1dG9kZXNrLmNvbS9hdWQvand0ZXhwNjAiLCJqdGkiOiJRMFE0UGlXUnVhYVhlOGdBcGE4akl3NlFKeEx0dzB0TEhGTlJkZU5tdnBuSlVrMGVMeHZsNnZlTHlxNVVnMzVDIiwiZXhwIjoxNTk3NDI3MTk5fQ.9hYyYUNQjONlOHJztlv0-ZErDxEpJu7uzlcaQyaR7rQ",
            //    "token_type": "Bearer",
            //    "expires_in": 86400
            // }
            //
            var accessToken = 'eyJhbGciOiJIUzI1NiIsImtpZCI6Imp3dF9zeW1tZXRyaWNfa2V5In0.eyJzY29wZSI6WyJkYXRhOnJlYWQiLCJkYXRhOndyaXRlIiwiZGF0YTpjcmVhdGUiLCJidWNrZXQ6cmVhZCIsImJ1Y2tldDpjcmVhdGUiXSwiY2xpZW50X2lkIjoidTdaSFJ5TU5hM0pTNlZjbE5Nbll3Rnp2WUJzcG5XdHgiLCJhdWQiOiJodHRwczovL2F1dG9kZXNrLmNvbS9hdWQvand0ZXhwNjAiLCJqdGkiOiJQOWZFT3JoQUdRbksweU5oTnJiNEVadzFyWldzR2hvSkI1SnQwV3BLcEpzWmllbmNLalM1Z2dqUjk3U1Q3bER1IiwiZXhwIjoxNTk4MzY4NTM4fQ.bVa4P2CPC3JJSgbmFaoaiZhzbL2_WqTXLnOdWweprfc';
```

```
var expireTimeSeconds = 60 * 30;
        onGetAccessToken(accessToken, expireTimeSeconds);
      }
    };
            let config3d = {
     extensions: ['Autodesk.InViewerSearch']
            };
     var documentId = 'urn:dXJuOmFkc2sub2JqZWN0czpvcy5vYmplY3Q6YW5pc2hhXzIwMjAvU2FtcGxlLnJ2dA';
    Autodesk.Viewing.Initializer(options, function onInitialized(){
      viewerApp = new Autodesk.Viewing.ViewingApplication('MyViewerDiv');
      viewerApp.registerViewer(viewerApp.k3D, Autodesk.Viewing.Private.GuiViewer3D, config3d);
      viewerApp.loadDocument(documentId, onDocumentLoadSuccess, onDocumentLoadFailure);
    });
    function onDocumentLoadSuccess(doc) {
      // We could still make use of Document.getSubItemsWithProperties()
      // However, when using a ViewingApplication, we have access to the **bubble** attribute,
// which references the root node of a graph that wraps each object from the Manifest JSON.
      var viewables = viewerApp.bubble.search({'type':'geometry'});
      if (viewables.length === 0) {
        console.error('Document contains no viewables.');
        return;
      }
      // Choose any of the avialble viewables
      viewerApp.selectItem(viewables[0].data, onItemLoadSuccess, onItemLoadFail);
    }
    function onDocumentLoadFailure(viewerErrorCode) {
```

```
            console.error('onDocumentLoadFailure() - errorCode:' + viewerErrorCode);
        }
        function onItemLoadSuccess(viewer, item) {
            console.log('onItemLoadSuccess()!');
            console.log(viewer);
            console.log(item);
            // Congratulations! The viewer is now ready to be used.
            console.log('Viewers are equal: ' + (viewer === viewerApp.getCurrentViewer()));
        }
        function onItemLoadFail(errorCode) {
            console.error('onItemLoadFail() - errorCode:' + errorCode);
        }
    </script>
</body>
```