Tampering and Sound Based IoT Framework for Real-Time Thermo-Mechanically Treated (TMT) Steel Cargo Theft Detection

Hashan D. Fernando,[1] Lahiru J. Ekanayake,[1] Prabath Gunathilake,[1] Roshan D. Yapa,[1] Amalka J. Pinidiyaarachchi,[1] Ruwan D. Nawarathna,[1]

[1] Department of Statistics and Computer Science, University of Peradeniya, Peradeniya 20400, Sri Lanka.

Correspondence should be addressed to Ruwan D. Nawarathna; ruwand@pdn.ac.lk

Abstract

Thermo-Mechanically Treated (TMT) steel bars are an essential need for all sorts of construction work. A substantial amount of construction is being done as a result of the population increase. Therefore, the global steel market is expected to witness rapid growth owing to the strong demand from the construction and transportation sections. The steel industry is highly dependent on shipping and transportation. Steel and metal cargos are most often moved using ships, trains and trucks or barges. Cargo theft is increasing progressively and has become one of the significant problems in the world. This study aimed to design, assemble and evaluate a real-time intelligent theft detecting device using a sophisticated smart seal (Module 1) and cargo audio data analysis (Module 2). Most of the existing theft detecting devices mainly focus on cargo door security, yet no device monitors the sounds inside steel cargo. In our past studies, we have developed a smart protector using ESP8266 for sealed packages. In this study, we have transformed the smart protector into an enhanced steel-based seal protector. The enhanced smart seal communicates with centralized Wi-Fi, and in the case of a seal damage, the proposed system will identify tampering attempts and notify them in real-time. The audio detection system detects the theft using convolution neural networks (CNN) and Raspberry Pi modules, notifies the owner by obtaining real-time data from the devices. Raw audio is transformed into Mel-frequency cepstral coefficients (MFCC) and Mel-spectrograms. Using MFCC as input, the basic CNN model is trained. Using Mel-spectrogram, different classification models are trained with the use of transfer learning. Among the trained models, DenceNet201 showed 93.75% accuracy with the used audio dataset. The functionality of the entire system was verified by developing a prototype design. Results showed that the smart theft detecting device has a good performance, and with the help of the suggested IoT framework, it is possible to identify theft in near real-time.

Introduction

Known as the "silent" or victimless crime, cargo theft costs the world economy several billion dollars annually. Across the world, trailers loaded with expensive goods are being stolen at a higher rate. At present, scientists can estimate the magnitude of the theft but are unable to detect the theft in real-time with a high success rate. Cargo theft gangs turn huge profits for crimes that carry minimal risks. They operate throughout the world, actively chasing expensive products that can be sold easily on the black market. While definitions of cargo theft vary and estimates of product loss values range from $8 to $30 billion per year, it is evident that this problem significantly impacts companies' revenues [1].

It is more important to understand the different types of thefts in cargo. Sometimes, the kind of theft can even suggest who the thieves are and what their interests are. As mentioned in [2], cargo theft can be broken into event types such as theft of full truckloads, pilferage, facility thefts, fictitious pickup and hijacking. The most common type of theft is full truckload theft, and it accounts for 62% of total thefts for million-dollar-plus cargo. It could be the theft of all the goods in the cargo with or without the container in which they were packaged. This often occurs in insecure parking areas. Pilferage is the theft of a few pieces or a small quantity of a relatively large shipment. It also accounts for 29% of the total theft. Facility thefts require much more planning and investigation. Thieves are looking for more profitable methods. Although facilities offer expensive goods in large quantities, facility theft imposes a higher risk.

Internet of Things (IoT) is described as a game-changer in theft detecting solutions due to their remote monitoring capabilities. IoT creates a widespread network of interconnected devices and allows the ingestion and sharing of data among these devices over a secure platform. IoT provides practical solutions for the logistics companies to remotely manage their fleet, monitor cargo condition, and safe product delivery. Devices like temperature sensors and GPS are attached to the cargo to track its condition and location. This data is transferred via gateways to a platform where fleet operators and cargo handlers can monitor and manage shipments. They gain visibility into their transportation operations which allows them to make intelligent decisions to improve the efficiency of their supply chain and ensure timely delivery of products. It is clear that all the parties in the global shipping ecosystem see what lies ahead for these technologies, and this transition and digitalization is already taking place. Combining IoT with transportation increases the efficiency of supply chains and helps fleet operators manage their cargo handling operations safely.

Despite various technologies that have been introduced in recent years to detect and track cargo thefts, the effectiveness of these detecting systems has not reduced cargo thefts that much. There are many issues in the existing detecting systems that delay theft identification, which is advantageous for thieves. Real-time detection has become a requirement in cargo theft detection systems. There are two types of devices for tracking vehicles; GPS tracking and GPS tracking along with immobilizer. In these devices, if theft is detected, the microcontroller and the GSM module send a notification to the user and receive the commands via the user's reply. The microcontroller converts the reply signal, which is then applied to the relay circuit. The communication system responds to this reply message and controls the vehicle's power. This approach is an application of connected devices or the Internet of things in transportation [3]. Sometimes when we use the GSM modules with these microcontrollers, Low signal levels result in poor system performance and slow response. Because of these reasons, when designing a theft detecting device, we need to specially consider the network connectivity.

Some of the currently available anti-theft devices can detect a moving object's motion, which has a body temperature similar to humans[4]. Such anti-theft devices can be actively used throughout the transportation process. The concept of motion detection, along with the body temperature sensors, was the basis for the detection of objects. The sensor circuits send a signal that triggers the receiver circuit producing an audible sound that serves as an alarm. IoT has been successfully used in theft detecting devices to capture images and monitor the status in real-time [5].

In our past studies [6], we have proposed a method to identify real-time theft detection in transport management. The proposed solution in [6] can track and protect any sealed package using a smart protector in real-time, and it can be easily applied for most packages regardless of size. It includes an advanced communication framework also to send signals to the client when the smart protector gets tampered. ESP8266 works as the system's main component because it has low power consumption and is a low-cost device.

The focus has been switched to solve theft detection problems using machine learning, according to the recent literature. In [7], authors have proposed a method to automatically detect theft attempts using camera footage-based machine learning. The Raspberry Pi 3B and Raspberry Pi Camera module were used as main components. The main intention of this device in [7] is to provide security to specific rooms or objects.

Deep learning has been successfully used to classify the environmental sounds and has generally performed better than traditional methods such as random forest ensemble [8] or support vector machine [9]. Most existing models [10], [11] use spectrogram presentation as input, such as log-mel features, which compress the amplitude with a log scale on mel-spectrograms. Depending on the task, these representations further transform into a more dense form of audio (e.g., MFCC [12], [13]). In [14], speech databases have been used to identify the languages from speech signals. Mel-frequency cepstral coefficients (MFCC) are used as spectral features. Overall, deep learning has considerably improved the performance of sound recognition [15]. The environment sound classification is essential for devices to understand the surroundings.

In this study, a new IoT solution is proposed to resolve the problems of steel bar theft on cargo that consists of three key components: a communication module, an enhanced version of the smart protector introduced in [6], and a deep learning-based audio recognition module. Reducing operating costs and increasing productivity is a primary requirement for cargo owners. Our solution would be helpful to remove the unnecessary use of hardware, and thereby the cost of the entire system will be reduced. The proposed solution provides a tracking device with highly accurate theft detecting capabilities and an efficient communication module that allows owners to send and receive large amounts of data in urban areas and continuously track the products' status and location. The proposed solution will solve the problems of real-time theft detection.

The remainder of the paper is organized as follows. Section 2 presents the methodology used to build the smart theft detecting system. Experiments and results are presented and discussed in Section 3. Finally, Section 4 summarizes the entire study by providing concluding remarks and recommendations.

Materials and Methods

Overview

As mentioned in the Introduction section, the proposed solution includes three main modules, as shown in Figure 1. The three modules are (1) a smart seal based theft detection module, (2) two communication modules and (3) an audio based theft detection module.

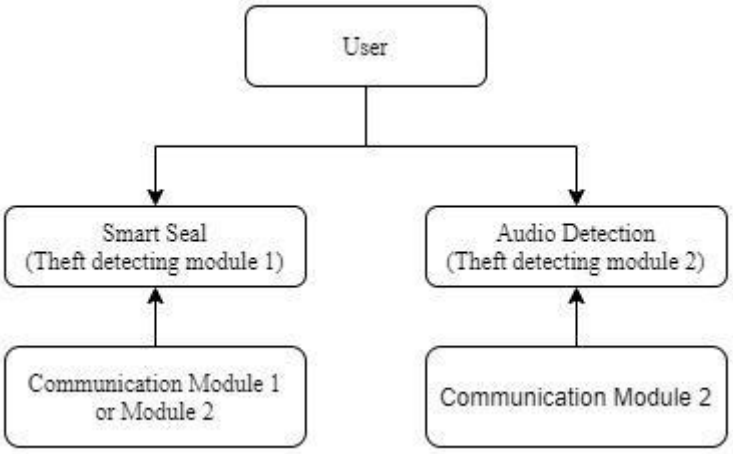


Figure 1: Three main modules of the proposed solution.

Three modules in the proposed solution work as follows. In the event of any suspected action related to cargo, the Theft detecting module 1 is activated, and the received data is evaluated. The established connection between cargo container and the user, created by the Communication Module 1 or 2, enables the cargo to notify the owner in a situation of theft. The user can take necessary actions to prevent theft. Also, if there are no authorized persons nearby, the system can send all records of the activities through the communication modules in real-time. It is vital to keep active communication between the transport provider and the cargo container by using minimal data and power consumption. Our communication modules ensure that data and power consumption is low. The current IoT market and audio recognition systems are growing very fast. Even though there are many anti-theft devices available in the market, most of them are specific to one particular case, such as “open-door” detection. There is no such solution that uses the sounds inside the cargo for theft detection. Theft detecting module 2 (See Figure 1) can detect theft in real-time using audio recognition. It uses a trained convolutional neural network (CNN) for audio recognition using an audio dataset created using existing audio data sets and a newly formed audio data set. Communication Module 2 notifies the output of Theft detecting module 2 to the user.

In the following subsections, we present the details of the three main components of the solution, starting with communication modules followed by the smart seal based theft detection module and finally, the audio based theft detection module.

Communication Modules

As shown in Figure 2 and 3, there will be multiple theft identification devices configured in multiple cargoes (containers). For example, in a shipping line, within close proximity, there will be more containers with the same theft identification devices. There are areas in the sea

where no mobile communication is available. Hence, having a direct internet connection in these kinds of situations might interrupt the communication between devices. Therefore, each theft identification device is connected to one centralize Wi-Fi available in the ship and connected to a centralized server on the service provider side. The server will manage communication between devices which is subscribed to the system. The framework will send alert messages to relevant parties efficiently when necessary.

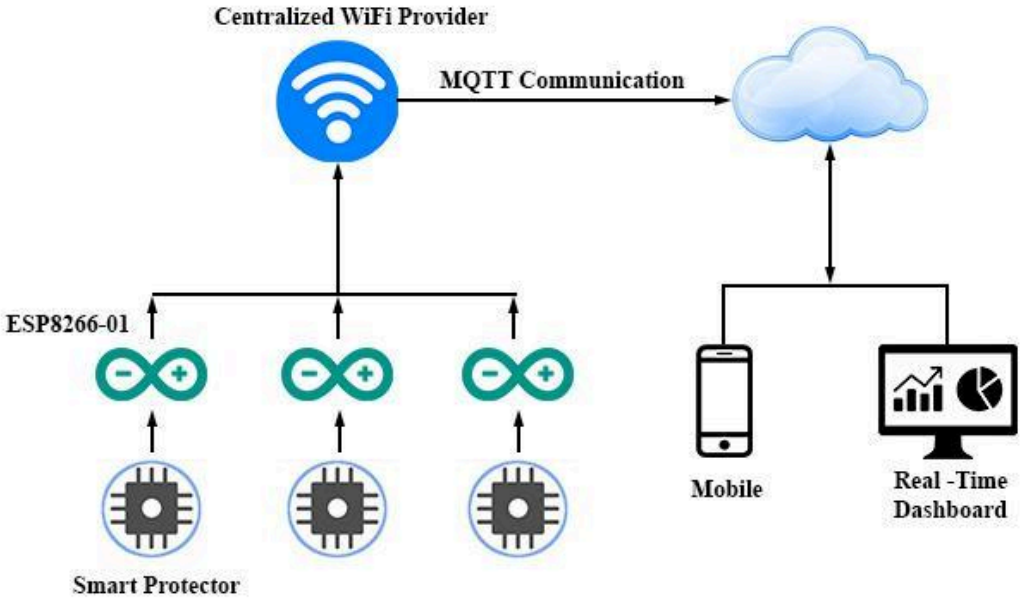


Figure 2:Architecture overview of Communication Module 1. Smart Seal (ESP8266) Communicates with Centralized Wi-Fi.

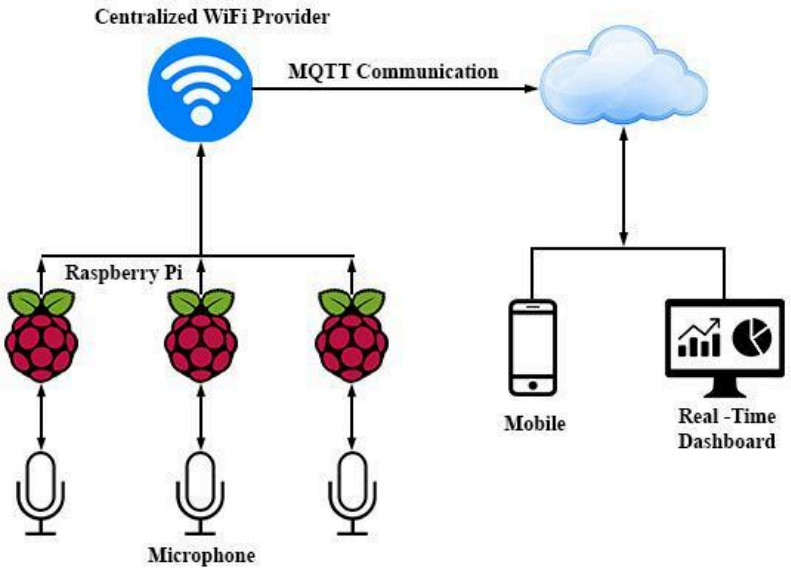


Figure 3: Architecture overview of Communication Module 2. Raspberry-Pi Communicates with Centralized Wi-Fi.

The container transportation and road transportation are inseparable because it is a process of end transportation mode. Whether it is air transport, sea transport or rail transport, at the beginning and at the end of the journey, road transportation of containers is required. Supplying required electricity to devices to communicate in these situations is a very difficult task because these devices are required to be always active. Raspberry Pi devices and GPS sensors consume a considerable amount of power. However, when road transportation is used, power can be gained from vehicles, but when sea transportation is used, renewable energy sources should be used.

Smart seal based theft detecting module

ESP8266P and GPS module are the system's (Figure 4) main components. Smart seals will be activated when the system is turned on. If any smart seals get damaged, the device will sends an alert message to the device owner. The smart theft detecting device transfers a few amounts of information to Message Queuing Telemetry Transport (MQTT) channel to get the maximum benefit of data usage. Periodically the data is sent to the MQTT channel. When the device is running in deep sleep mode, the GPS will not consume power. The below flowchart (Figure 5) gives the complete flow of working of the system from theft detection to sending the alert message to the owner.

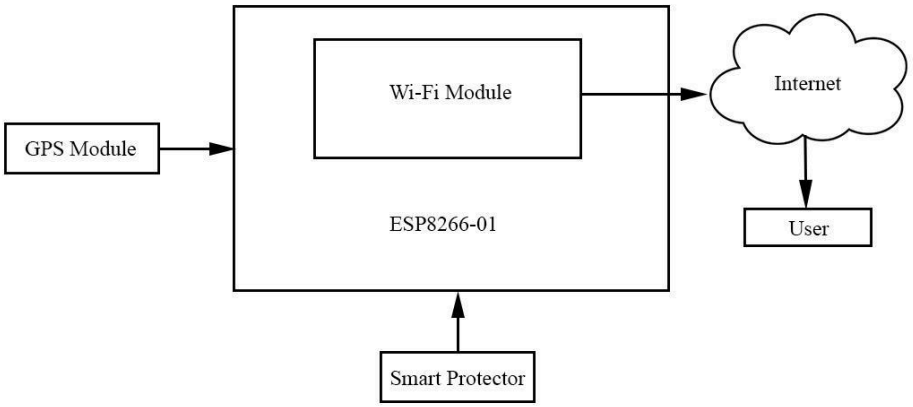


Figure 4: System Block Diagram of Smart Seal. ESP8266 used as Main Device

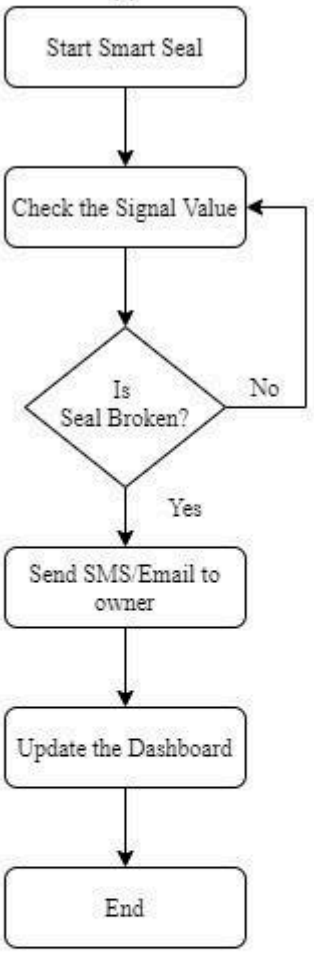


Figure 5: Flowchart of the System (Theft Detecting Module 1).

Smart seal

The smart seal is connected to a power source and consumes power from it, as shown in Figure 6. The smart seal transfers an optimal amount of information to the MQTT channel to get the maximum benefit of power consumption. The sensor node is running in deep sleep mode and, it will transfer information to MQTT channel periodically. After each period, it is required to know whether a tampering attempt occurred or not. Therefore, if no tampering attempts have taken place, the sensor node will send a signal "1" to verify the current status. Within each period (15 – 60 sec), the sensor collects data from the smart protector and stores the collected data. If any tampering attempt occurred, the device will act immediately and push the signal "0" to the MQTT channel to indicate a tampering attempt. Figure 7 explains how the device pushes data periodically and what happens in any tampering attempt. We can use the smart seal with the communication module 2, but then ESP8266 modules should communicate with the Raspberry Pi and Raspberry Pi will communicate with the centralized Wi-Fi.

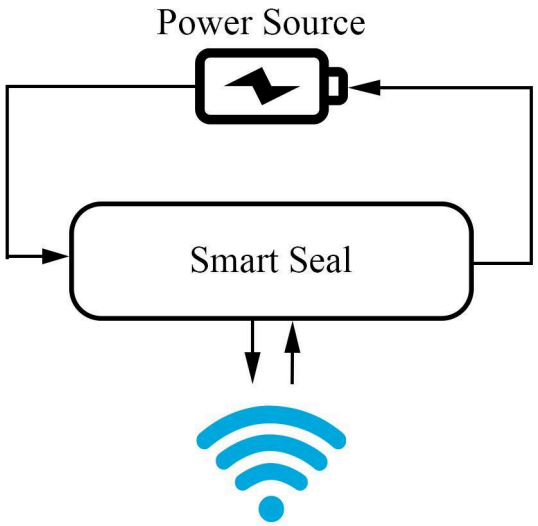


Figure 6: Power consumption and connectivity of the smart seal.

According to Figure 7, the sensor node pushes the data periodically in every 15 seconds. But we can identify that there is a breakdown of the pattern in the 40th second. That means, there should be a tampering attempt in the 40th second. Just after the smart theft identification device identifies any security breach, it will act immediately without considering the cycle.

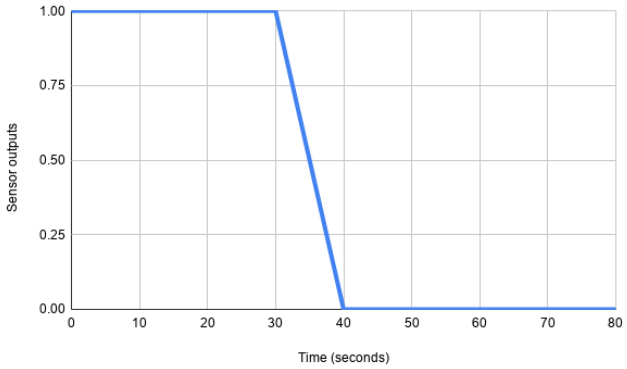


Figure 7: Data transfer graph when tampering attempt occurred.

Identification of Smart seal tampering attempts

The current IoT market is proliferating and there are a variety of sensors available in the market for different purposes, as well as sensors already available to identify tampering attempts. But most of them are specific to one scope, such as open-door detection for cargo, electricity meter tampering detection. There is no unique solution attached with steel bar containers to identify steel bar thefts. If we have a strip-shaped wrapper around steel bars that we can use to wrap any steel at any time on any location and if that wrapping material got damaged, the relevant parties would be informed according to the configured order.

Usually, steel bars are stacked up when transported in a container. Due to the heaviness of the steel bars, it is impossible to take out one from the bottom of the stack. As a solution for this, we can keep steel bars wrapped with enhanced smart seals as the top layer of the stack, as shown in Figure 8. This would make it impossible for the thieves to take the steel bars, irrelevant to its position on the stack. If we have a mesh type protection in the form of a strip around a set of steel bars and if that meshed protection is damaged, we should be able to know the exact place where it got damaged. When the number of columns and rows increase in the mesh covering, we can have more secure protection, which means there is a more meshed covering area over a set of steel bars through which we can detect the tampering attempts.

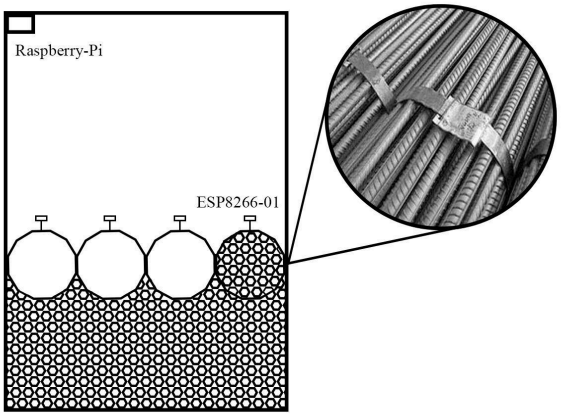


Figure 8: Proposed method of smart seal wrapping around steel bars

This meshed skeleton has been designed using a conductive element, and it can be covered by any kind of material that is suitable. But in this specific scenario, we tend to use a strip of rubber wrapper around an enhanced smart seal to prevent any kind of voltage leakage. There will be a small voltage between this mesh, and if any kind of break occurs to this continuous electronic signal, we can easily figure out that the container is not secure anymore. Since there are more horizontal or vertical electronic signals passing through this mesh, we can identify the exact location of the damaged point. But, when there are more electronic signals, we should have more inputs to our electronic devices. Similarly, due to the increasing inputs, the device (microcontroller) is also getting larger in order to have enough input pins.

Reading electronic signal break

The input voltage of 3.3 is always supplied to one of the conductive wires in the mesh from the microcontroller (E). Due to the resistance of the conductive wire and applied resistor (R), there will be a small current passing through the wire. The microcontroller (E) will always read the availability of the current at the end of the wire (D) (See Figure 9).

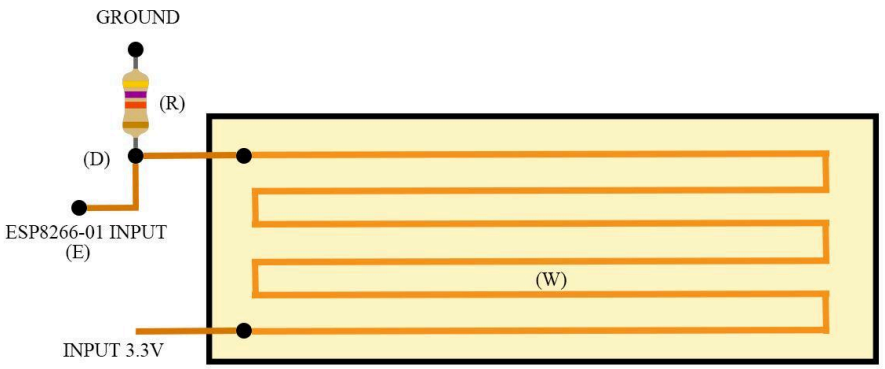


Figure 9: Identification of the electronic signal break.

If there are more input wires placed in specified areas of the wrapper, in case of a tampering attempt, the microcontroller will be able to identify the damaged wire and specific area of the

mesh that got damaged (Figure 9). But if the requirement is only to identify the "tampering attempt “in the first place, it is not required to have more input wires in the wrapper. By setting one input wire inside the wrapper as a unique mesh pattern, the microcontroller will be able to identify the anomalies of the signal.

Shape of the conductive mesh

The shape of the conductive material inside the wrapper is crucial for finalizing the design of security, size of microcontroller, and cost of the product. Mesh of diagram one contains rows and columns of conductive wires inside the wrapper. Providing inputs for each row and column in the microcontroller will identify the exact place of tampered points using the grids of the conductive mesh. Without a conductive mesh grid, we can set one conductive wire through the entire wrapper with this design (See Figure 10). This will simplify the wrapper design, and if the wrapper got damaged at least once, we could not use the wrapper anymore. According to Figure 9, if one wire of the conductive mesh got destroyed, the microcontroller will read other wires independently. However according to Figure 9, it will not be possible to use a conductive strip once damaged.

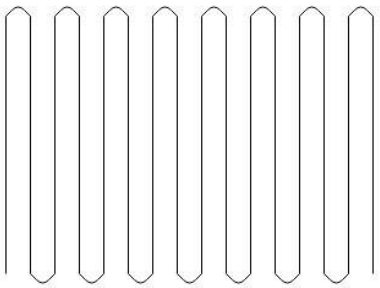


Figure 10: Shape of the conductive mesh.

Smooth power consumption

ESP8266 consumes more power when connecting to WiFi to perform various tasks. We noted that it could consume 50 mA ~ l70 mA. In this project, ESP8266 runs in deep sleep mode, and it does not require a constant WiFi connection. Using ESP8266 with rechargeable Li-ion or LiPo batteries is an excellent solution. Consider a LiPo or Li-ion battery, which has an output of 4.2 V approximately when fully charged. Anyhow, when the battery begins to discharge, the voltage starts to drop, as illustrated in Figure 11.

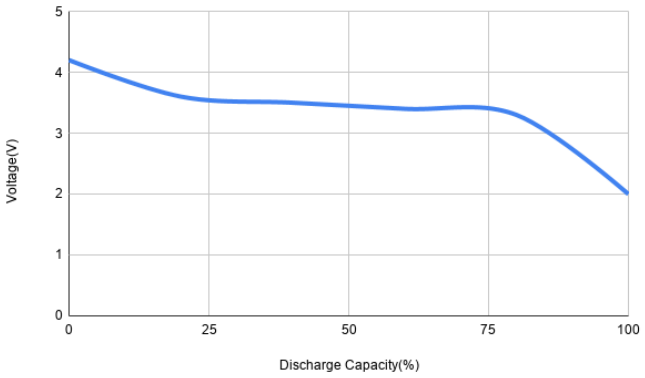


Figure 11: Voltage discharge curve.

The ESP8266 recommended operating voltage is 3.3 V, but it can operate with voltages 3 V - 3.6 V. So, we cannot plug the LiPo or Li-Ion battery directly into an ESP8266. Therefore, we need a voltage regulator. One of the good LDOs (Low-dropout regulator) that we were able to test was the MCP1700-3302E. Figure 12 shows the circuit diagram of the setup. The ceramic capacitor and electrolytic capacitor are in parallel to GND. The use of a ceramic capacitor and an electrolytic capacitor connected in parallel to GND is to smooth the voltages peaks. The capacitors prevent unstable behaviour and unexpected resets for ESP8266. There is a pushbutton connected to the RESET pin of the ESP8266-0l so that the module can be quickly restarted without unplugging pins.

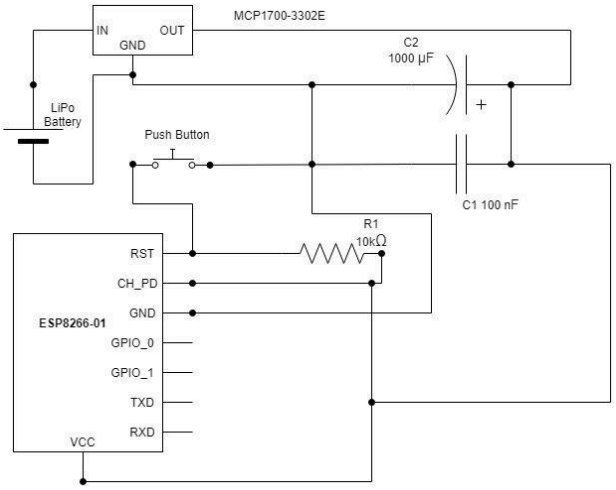


Figure 12: Circuit diagram of smooth power usage.

Audio based theft detecting module

A system block diagrams was used as the framework of the system. The Raspberry Pi 3, GPS module, and microphone are the system's (Figure 13) main components. When the system is turned on, audio device will be turned on and it will feed audio chunks to Raspberry Pi. As shown in the Figure 15, Raspberry Pi will check average amplitude of audio chunk and if it's greater than silent threshold value then audio chunk will be use to theft prediction using the TensorFlow library of Deep Learning. This process detects theft using audio recognition.

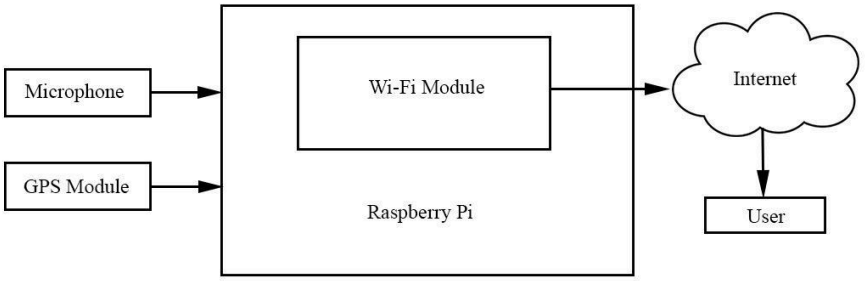


Figure 13: System Block Diagram of Audio Detection device. Raspberry Pi used as Main Device.

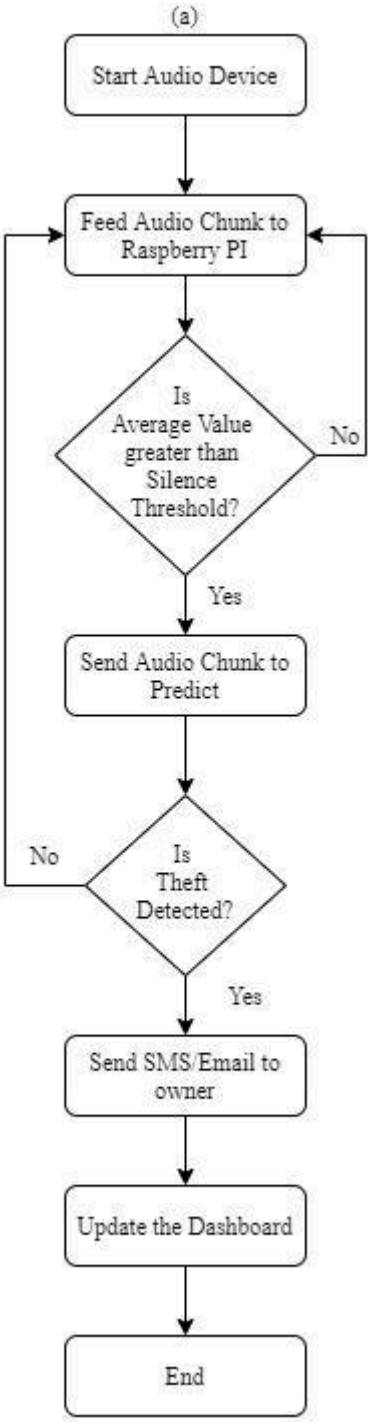


Figure 14: Flowchart of the Audio Detection System (Theft Detecting Module 2).

Dataset

Unfortunately, publicly available datasets of environmental recording are very limited. This is due to high cost of manual annotation. In this research, 3 datasets (Steel bar's dragging audio

dataset, Urbansound8K [12] dataset and YorNoise [16] dataset) were used together as one whole dataset. 1747 steel bar dragging audio clips were recorded with a maximum length of 4 seconds. When training the basic CNN model and transfer learning models, 1107 and 1558 samples were used from the steel bar dragging dataset respectively. Figure 16 shows an image of a scaled shipping container model bar(Figure 16) which was used to demonstrate steel bar (Figure 15) sounds.



Figure 15: Thermo-Mechanically Treated (TMT) Steel Bars.



Figure 16: Scaled Shipping Container Model.

The UrbanSound taxonomy [12] is a proposed taxonomy of sound sources, developed based on analysis of noise complaints in New York City between 2010 and 2014. The same authors have also created the Urbansound dataset [12], selecting and manually labeling audio from the Freesound [17] repository. 10 different classes from the Urbansound taxonomy were selected and 1302 different recordings were labeled, for a total of 18.5 hours of labeled audio. A curated subset with 8732 audio clips with a maximum length of 4 seconds is known as Urbansound8k. For this study, only 8 classes from UrbanSound dataset were used. Below table shows the complete data representation,

Table 1: Audio Dataset.

Class	Number of Samples	Average Time (sec)
Iron	1746	2.14
Rail	620	4.00
Traffic	907	4.00
Air conditioner	1000	3.99
Children playing	1000	3.96
Dog barking	1000	3.15
Jackhammer	1000	3.61
Drilling	1000	3.55
Engine idling	1000	3.94
Siren	929	3.91
Street music	1000	4.00

Mel-Frequency cepstral coefficients (MFCCs)

The human auditory system processes audio signals in a nonlinear fashion. It is widely studied that the lower frequency component of an audio signal contains more information [18]. Because of that, nonlinear mel-scale filter has been designed to indicate lower frequency component over higher frequency ones. Mel frequency cepstrum is a representation of the short-term power spectrum of an audio frame using linear cosine transform of log power spectrum on a nonlinear mel frequency scale. `

Several steps [14] have been used to gain mel-frequency cepstral coefficients (MFCCs) from an audio signal. First, we need to Fourier Transform a given audio segment. Then, we need to compute the powers of the above-mentioned spectrum within the triangular overlapping windows placed according to the mel scale. Then, we take logs of the power at each of the mel frequencies and compute the discrete cosine transform (DCT) of the list of mel log powers. Finally, the amplitudes of the resulting spectrum give MFCCs.

Mel-Spectrogram

The processing power for training and inference increases when input of the machine system becomes complex. Thus, one would like to reduce the dimensions of the input as much as possible. A linear spectrogram has reasonable amount of redundant information between adjacent frequency bins and is often reduced to many frequency bands using a filter bank. The mel scaled filters are often used for audio classification. The spectrogram that results for applying a Mel-scale filter bank is often called a Mel-spectrogram. Figure 17 shows Mel-Spectrogram representation of used audio classes. Mel frequency cepstral coefficients (MFCC) is a feature representation computed by forming a discreate cosine transform (DCT) on a mel spectrogram. MFCC have been very popular in speech recognition tasks, however in general sound classification tasks mel- spectrograms tend to perform better.

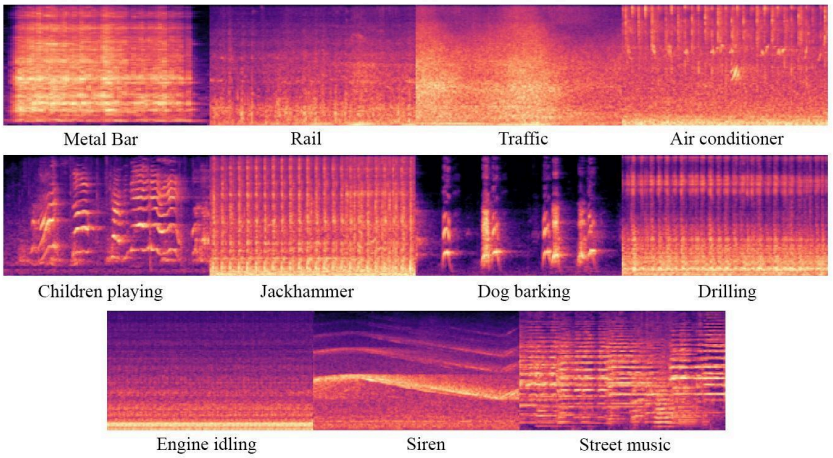


Figure 17: Mel-Spectrograms of the sound categories.

Basic CNN Model and Transfer Learning

In this study, selected datasets were chosen in order to obtain required identification. The raw audio set was transformed into spectrogram images. The dataset compromised of 10,452 unique spectrograms across 11 classes out of which 70% were reserved for training, 15% for validation and 15% for testing.

Each raw audio waveform was first processed by resampling it to 16 kHz and then trimming the silent part of the audio clip which was further trimmed to a length of 2 seconds. This is a necessary condition in CNN because the models require a constant input dimension. Next, the trimmed audio clip was transformed into two input types: MFCC and Mel-Spectrogram (Figure 18). Each MFCC input is two dimensional NumPy array of size 199 × 13 and each Mel-Spectrogram has a size of 334 × 217 which was resized to fit the desired size depending on the model while preserving the quality of image. Basic CNN model (Table 2) training was performed using MFCC as input. The categorical cross-entropy loss and adam optimizer with learning rates of 0.001 or 0.0001 were used. A model was trained for 150 epochs (batch size 64) and the model that showed the highest validation accuracy was chosen.

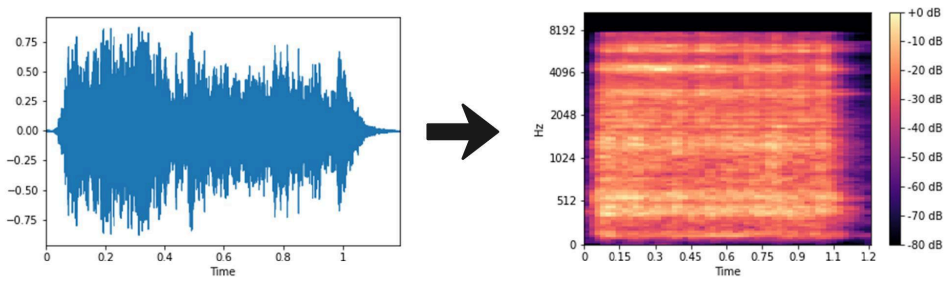


Figure 18: Raw audio signal and corresponding Mel Spectrogram.

Table 2: Basic CNN Layer-Wise Architecture

Layer (type)	Output Shape	No. of Parameters
conv2d_1(Conv2D)	(None, 199, 13, 16)	160
conv2d_2(Conv2D)	(None, 199, 13, 32)	4640
conv2d_3(Conv2D)	(None, 199, 13, 64)	18496
conv2d_4(Conv2D)	(None, 199, 13, 128)	73856
Max_pooling2d_1(MaxPooling2D)	(None, 99, 6, 128)	0
Dropout_1(Dropout)	(None, 99, 6, 128)	0
Flattern_1(Flatten)	(None, 76032)	0
Dense_1(Dence)	(None, 128)	9732224
Dense_2(Dence)	(None, 64)	8256
Dense_3(Dence)	(None, 11)	715

Convolutional neural networks are capable of achieving state-of-the-art results in image and audio classification. If the input data exhibits the properties of locality (where features in the data have a local spatial support) and translation invariance (where features in the data are independent of location), then we can use CNNs. Because of those mentioned features, spectrogram is a particularly suitable candidate for training through CNNs. However, the size of the labeled dataset tends to be insufficient for acceptable training through a deep CNN. This issue can be solved through transfer learning, where the initial learnt weights from a CNN trained on one particular task are extracted and frozen, and only later layers are fine tuned. Transfer learning has the benefit of shortening training time and improving performance. The success of this knowledge transfer is dependent on the level of similarity between the two tasks.

For the transfer learning experiment, Several CNN models were chosen which were trained on ‘imagenet’ dataset. These models are known to be successful for image classification. When these models are applied for transfer learning in other image classification tasks, it is common to only re-train the fully connected layers. Thus, in this study, the final fully connected layers were re-trained in each model. The deep learning models chosen for the experiment were VGG16, VGG19, MobileNet, MobileNetV2, ResNet50 and DenseNet201.

Software architecture in the framework

There are many messaging protocols that support this framework as input sources. Kafka and MQTT are widely used protocols in IoT industry and the suggested solution is also carried out by using the MQTT protocol. Different data formats such as JSON, XML, etc., support this framework. Real time data are stored in a reliable storage and there are several advanced databases such as RDBMS, MongoDB, etc., that support this framework. These data can be used in future to do further analysis and predictions.

Cloud service providers such as google, amazon have implemented whole IoT architecture in the cloud. As shown in the Figure 19 Raspberry Pi or ESP8266 devices send data securely through Google Cloud IoT Core using MQTT protocol, then Firebase Cloud Functions process the data in an event-based way that update the device current state in Firebase real-time database and save the raw data in BigQuery. The stored data can be accessed through Data Studio dashboard and through a custom WebApp hosted on Firebase Hosting. This will show how each theft detecting device can be easily connected to deploy products that scale on demand.

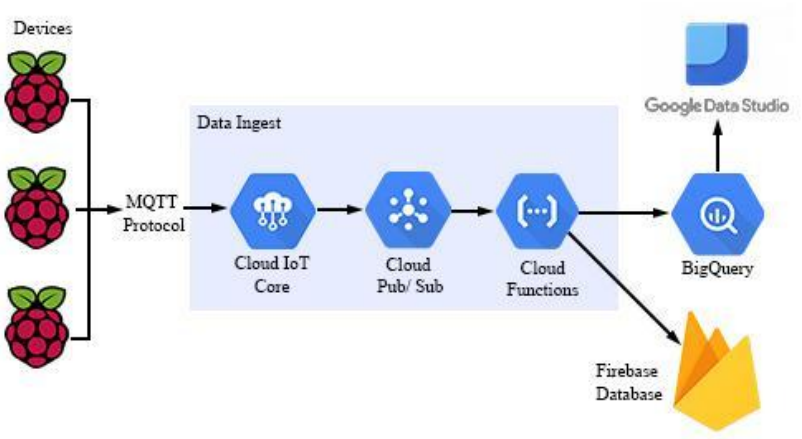


Figure 19: Software architecture of the proposed System. Devices can be Raspberry Pi or ESP8266, depending on the theft detecting module.

Results and Discussion

The research work that was carried out in this study is mainly focused on the design and development of practical and efficient audio detection surveillance and further development of a smart protector; i.e., a smart theft detecting device to solve security issues which will help to reduce theft in cargoes. Though a large amount of research has been done in the past to solve cargo security problems, it is still really challenging to develop security systems due to various theft actions.

The proposed theft detecting system was tested with multiple sensor nodes and multiple tampering attempts at the same time. Smart seals (ESP8266) and Raspberry Pi were capable of pushing data through MQTT protocols without any delay. The status of the devices was identified in real time without any conflict and when there were any tampering attempts, the theft detecting system was capable of alerting relevant owners. Figure 20 demonstrates a prototype of the smart seal. Depending on the customer's requirements we have developed this device using ESP 8266 – 12F Module.

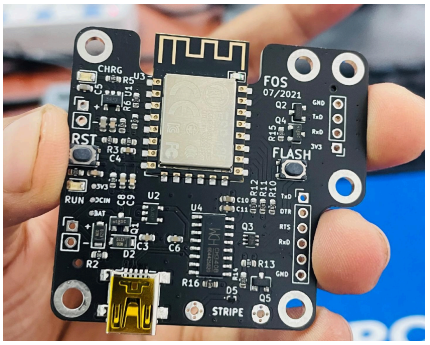


Figure 20: Smart Seal Prototype PCB

In the transport industry, steel bar containers travel inside ships for months. Therefore, recharging or changing power source in smart seal is an impractical thing. Powering up a

sensor for a long time using a limited power source is a challenging task. There should be a single power source that can power up smart seal during whole journey.

To achieve this target, we tested the smart seal with different types of power sources, such as direct power from service provider, power banks, UPS, Lithium-ion batteries, LiPo batteries and solar panels. Most of them can provide consistent power supply to the sensors. As many of the end users look for a plug and play solution for smart seal, Lithium Polymer (LiPo) battery was a good power source for the smart seal. There are 3.7 V LiPo batteries available in the industry which have a capacity of 30 mAh-10000 mAh. In the audio detection module and communication module, only one Raspberry-Pi device has to be used for one container (as shown in the Figure 8) and consistent power should be provided from the service provider to Raspberry-Pi.

Performance and reliability are the most critical aspects in the IoT solutions. In general, all the topics that are published from IoT sensor must be received into the relevant server accurately and reliably. Our framework has been implemented based on Google IoT Core and it uses public key (or asymmetric) authentication. Device state data can be structured or unstructured, and flows only in the device-to-cloud direction. In the google IoT Core, the protocol bridge has built-in support for secure connection over industry standard protocols such as MQTT and HTTP. The protocol bridge publishes all device telemetry to Cloud Pub/Sub, which can then be consumed by downstream analytic systems. Overall, Google IoT core perform well with sensor devices without any difficulties.

When the number of devices gets increased, the load becomes higher for the servers. The default maximum devices per a Google IoT project is 100,000 and can be increased. In order to increase the devices, it is required to expand the hardware resources and Google will do that seamlessly. Google Cloud IoT Core limits the projects that generate huge amounts of loads when retry field operations without waiting. For retry operations, Google recommends to implement a truncated exponential backoff algorithm with introduced jitter. We cannot load balance MQTT brokers like we load balance web servers. A MQTT bridge allows us to connect two or more MQTT brokers together and increase the high availability. These brokers are sharing messages between their systems.

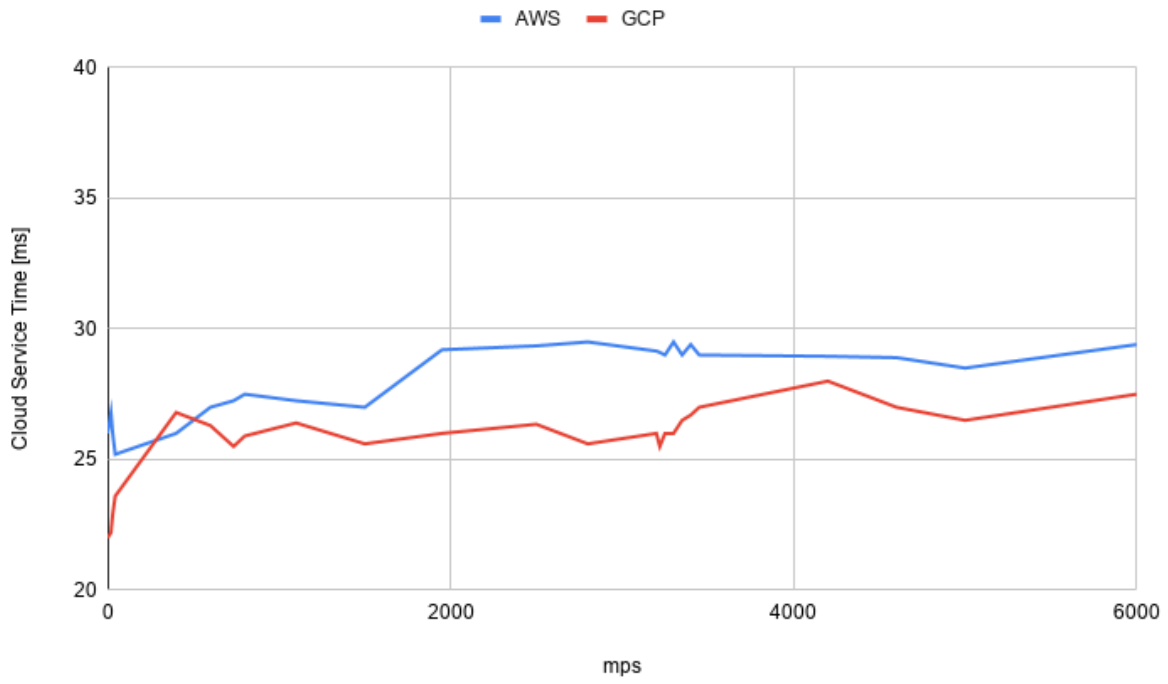


Figure 21: Cloud Service Time comparison between Google Cloud Platform and Amazon Web Services/ Point to Point Scenario.

The Google IoT core has been tested for benchmarking [19] by increasing the sending message rate to only one client up to 100mps. Initially sending frequency set to 10mps and increasing the connected clients between 1 to 600. Google IoT Core responds faster than other platforms like AWS, Azure (Figure 20).

In order to solve above performance issues and to get the maximum benefits from the available resources, a new scheduling algorithm is suggested. IoT devices can use the deep sleep mode for less power consumption and waking up period can be dynamically assigned based on the load of the servers. This will reduce the server idle time. Due to real time data unavailability, this solution will not be suited for synchronized communication.

Performance of Audio Classification and Theft Detecting System.
This study was started by training basic CNN model and it achieved 76.23% accuracy. Table 3 lists the accuracy achieved by the transfer learning models. The DenceNet201 achieved accuracy of 93.75%, which is the highest among trained transfer learning models. Hyperparameters including batch size and learning rate were tuned whenever possible. Figure 22 show the training-validation accuracy and loss of each transfer learning model respectively.

Table 3: Training Configurations and Results for all CNN Architectures.

Architecture	Image Size	No. Of Epochs	Optimizer	Learning Rate	Batch Size	Accuracy (%)
DenseNet201	224 × 224 × 3	200	Adam	0.0005	128	93.75
MobileNetV2	224 × 224 × 3	200	Adam	0.0005	128	91.00
MobileNet	224 × 224 × 3	200	Adam	0.0005	128	88.84
VGG16	224 × 224 × 3	400	Adam	0.0005	128	86.54
VGG19	224 × 224 × 3	200	Adam	0.0005	128	82.65
ResNet50	224 × 224 × 3	200	Adam	0.0005	128	59.76

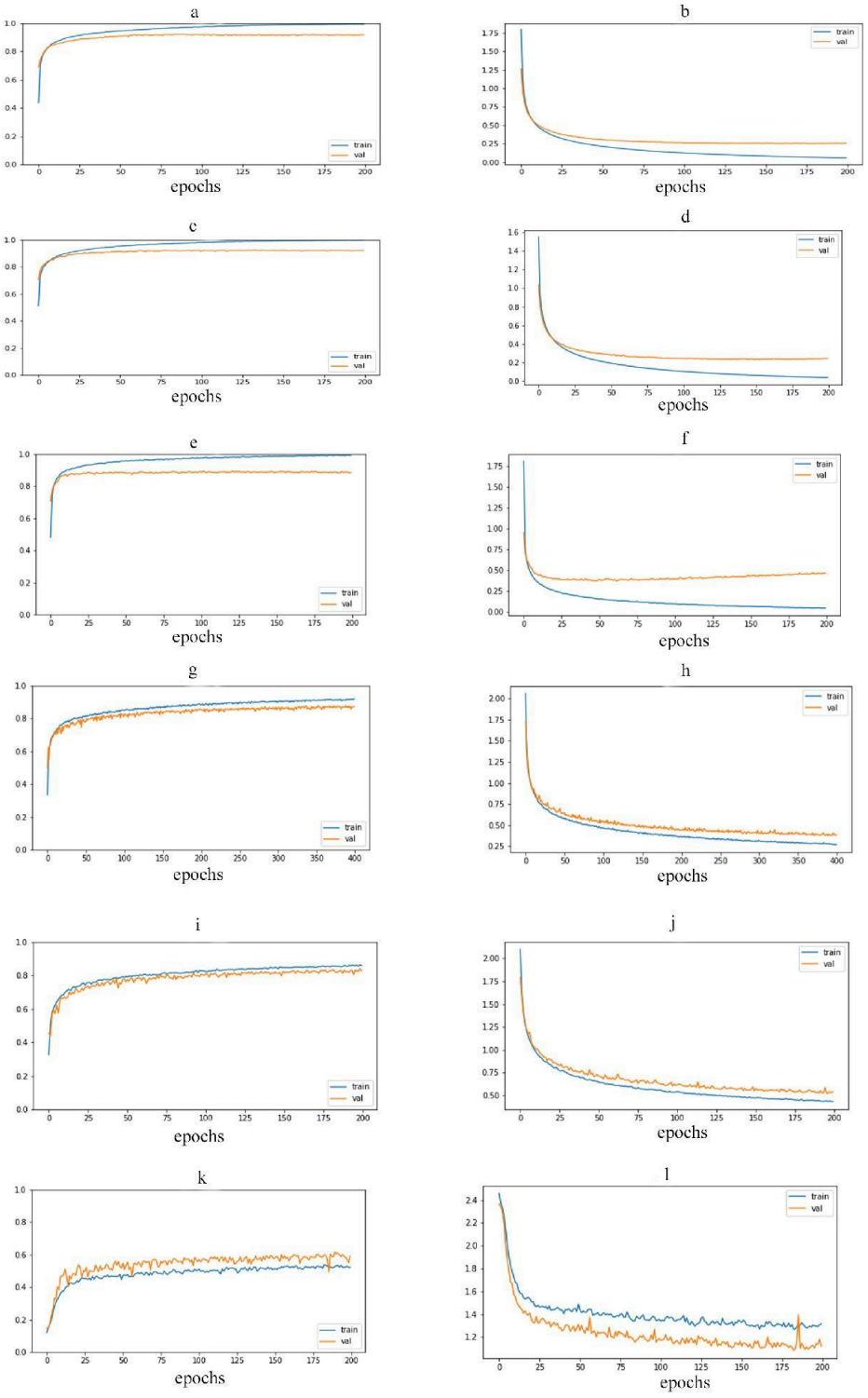


Figure 22: Training-Validation Accuracy and Loss of the Transfer Learning Models. (a) DenseNet201 Accuracy. (b) DenseNet201 Loss. (c) MobileNetV2 Accuracy. (d) MobileNetV2 Loss. (e) MobileNet Accuracy. (f) MobileNet Loss. (g) VGG16 Accuracy. (h) VGG16 Loss. (i) VGG19 Accuracy. (j) VGG19 Loss. (k) ResNet50 Accuracy. (l) ResNet50 Loss.

In this proposed approach to smart security, the end-user has the ability to control the monitoring time period. If any security checking happens, user can pause the system and continue the manual check-ups. When the system is running, smart sensors will run

continuously with deep sleep mode. If any damage or theft is detected, depending on the theft identification module and communication module, system will poll data from sensors and send it for identification. If the theft identification system flags a possible theft, the end-user is notified through an alert message along with GPS coordinates and it will be displayed in real-time dashboard as shown in the Figure 23. As shown in the dashboard, "tamper_status" equal to "1" indicates a detection of a theft. Theft detected device is shown in the red color row and the relevant GPS coordinates and location is shown in the map.

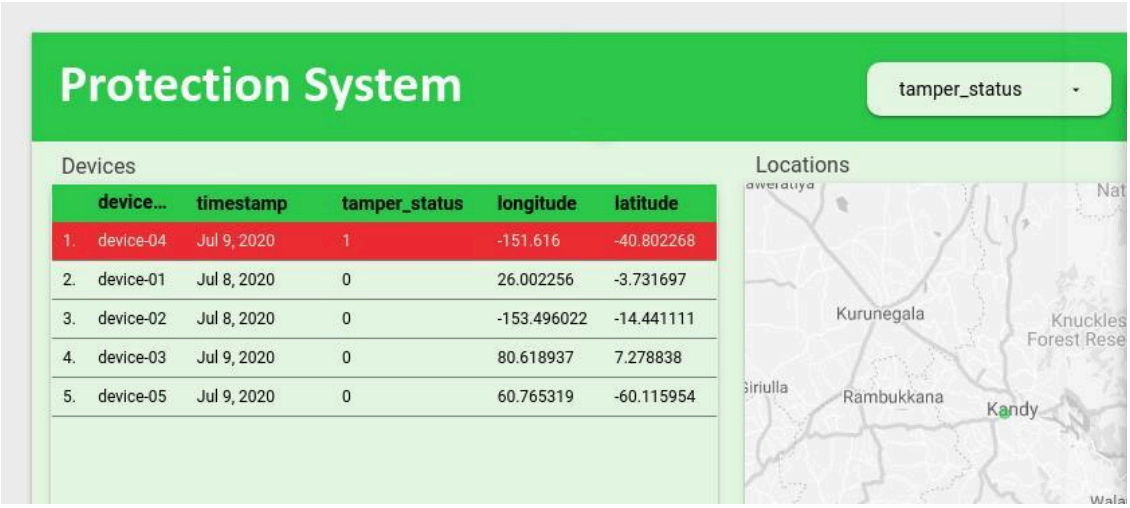


Figure 23: Google Data Studio Dashboard. Shows the Real-Time Status and Location of the Devices.

For further improvement of the system, Raspberry Pi camera can be installed with which the secondary identification images can be captured and then the captured and images or videos can be used as evidence. Along with all these features, the overall results which are expected from the system are as follows.

- Anti-Theft device which can give a real time alert message along with GPS location to the owner.
- A real time Dashboard with tracking.
- A user-friendly application for owner to control the device (i.e., when to start or stop).
- Efficient device that has secondary method to identify real theft and false alarm.
- A device which will give authority to the owner or end-user to take possible actions against the theft through various options.

The system will identify possible theft only when the audio chunk prediction accuracy exceeds a certain threshold that is pre-set in the system. After successfully developing the product, it can be applied for many smart security systems and not just for steel transporting cargoes. This smart security system would be immensely useful in real time theft detection in the feature.

Conclusions

This research is recommended as an appropriate method for gaining a better understanding of an emerging phenomenon. This study is aimed to design, assemble and determine the performance of the smart theft detecting device using enhanced smart seal and audio recognition. In spite of the fact that a considerable amount of research has been carried out in the past to resolve such security issues, it still remains as a challenge due to the increased

complexity of various theft activities. This study successfully developed a design to observe the functionality of the device. This will allow the end user to track and protect their products in real-time by using this device and supporting framework. It can be applicable for any types of steel bars and objects which generate unique sounds.

Data Availability

The data used to support the findings of this study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Statement

This work was supported by the Accelerating Higher Education Expansion and Development Operation (AHEAD) Innovation Commercialization Enhancement (ICE) grants.

References

[1] B. D. Burges, "Cargo Theft, Loss Prevention, and Supply Chain Security," 2013.

[2] Sensitech Inc., "Cargo Theft Report,", *https://landline.media/wp-content/uploads/2020/05/2020_q1_us.pdf*, 2020.

[3] Y. Y. Musa and J. Wang, "Vehicle Tracking and Anti-theft System using GPS-GSM," *International Journal of Engineering Research & Technology*, vol. 1, no. 10, pp. 1–4, 2012.

[4] R. Dellosa, "Development of an Anti-Theft Device using Motion Detection and Body Temperature," *Asia Pacific Journal of Multidisciplinary Research*, vol. 2, no. 6, pp. 121–124, 2014.

[5] A. Ayshwarya, R. Dhanalakshmi, P. U. Rani, S. Haripriya, and R. Joshna, "Image Processing and IoT Based Sand Theft Detection and Indication System," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1S, pp. 170–174, 2019.

[6] L. J. Ekanayake, R. D. Nawarathna, P. Gunathilake, R. D. Yapa, and A. J. Pinidiyaarachchi, "Smart Protector: A Real-time Theft Prevention System for Transportation Management," 2019 14th IEEE International Conference on Industrial and Information Systems, Engineering Innovations for Industry 4.0, pp. 180–185, 2019.

[7] A. Kushwaha, A. Mishra, K. Kamble, R. Janbhare, and P. A. Pokhare, "Theft Detection using Machine Learning," *IOSR Journal of Engineering*, vol. 8, pp. 67–71, 2018.

[8] A. Liaw and M. Wiener, "Classification and Regression by RandomForest," *R News*, vol. 2, no. 2, pp. 18–22, 2003.

[9] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proceedings of the ACM International Conference on Multimedia*, pp. 1015-1018, 2015.

[10] K. J. Piczak, "Environmental Sound Classification with Convolutional Neural Networks" in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2015.

[11] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2721-2725, 2017.

[12] J. Salamon, C. Jacoby and J. Pablo Bello, "A Dataset and Taxonomy for Urban Sound Research," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 1041–1044, 2014.

[13] V. Boddapati, A. Petef, J. Rasmusson, and L. Lundberg, "Classifying environmental sounds using image recognition networks," *Procedia Computer Science*, vol. 112, pp. 2048–2056, 2017.

[14] S. G. Koolagudi, D. Rastogi and K. S. Rao, "Identification of language using mel-frequency cepstral coefficients (MFCC)," *Procedia Engineering*, vol. 38, pp. 3391–3398, 2012.

[15] B. Zhu, C. Wang, F. Liu et al., "Learning Environmental Sounds with Multi-scale Convolutional Neural Network," *in Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pp. 2598–2604, 2018.

[16] F. Medhat, D. Chesmore and J. Robinson, “Masked conditional neural networks for environmental sound classification,” *in International conference on innovative techniques and applications of artificial intelligence*, pp. 21–33, 2017.
[17] F. Font, G. Roma and X. Serra, “Freesound Technical Demo,” in *Proceedings of the 21st ACM international conference on Multimedia*, pp. 411–412, 2013.
[18] A. Srinivasan, “Speaker identification and verification using vector quantization and Mel frequency cepstral coefficients,” *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 1, pp. 33–40, 2012.
[19] P. Pierleoni, R. Concetti, A. Belli and L. Palma, “Amazon, Google and Microsoft Solutions for IoT: Architectures and a Performance Comparison,” *IEEE Access*, vol. 8, no. 1, pp. 5455-5470, 2020.