# Using Data Science and Remote Sensing to Understand Ship Traffic with Application to Detecting Forced Labor (Human Trafficking) Final Report by Thomas Keeley & Harry Newton

## Abstract

The research presented in this report includes a lightweight Machine Learning algorithm based on Keras to identify ships in satellite images. To evaluate the effectiveness of this object, a procedure to use public-domain ship navigational signals data is employed using Big Data Analysis to find navigational signals for the same time and location as the image. In addition to providing detection information, the signals data also provides all-weather, continuous coverage and ship identifications, but only for the subset of ships broadcasting the Automated Information System, which varies depending on the ship size and it’s country of registration (flag). An application is offered to show the value of this signals data to find ships with patterns of behavior associated with Human Rights, Forced Labor risks.

Figure 1 Overall Methodology

## Introduction

By using Machine Learning (ML) on readily available imagery and signals collected by satellites, the presence of ships and their activity can be partially understood. In this report, we first showcase our work to improve an ML algorithm that operates on imagery based to identify the location of ships. These results beat the winning submission for a Kaggle competition to do so. We tune this algorithm with training data based on assuming the ground truth by the observation of a navigation signal detected at the same location and time. This navigation signal is the Automatic Identification System (AIS) that most ships emit continuously to comply with international agreements designed to avoid collisions at sea or in port. The imagery and signals data that we used are described in Table 1. We describe this first model as our Ship Imagery Model because once tuned, it only requires imagery to identify ships.

We develop a corresponding Ship Signals Model to summarize voyages that can be discerned by AIS data alone, then compare the results for “scenes” where we have corresponding imagery and signals at nearly the same time. We use this Differences Model to find where the location of ships in time and space does not align in the data.

We conclude with an application of the models to the Human Trafficking problem of forced labor crews and propose a risk model and data visualization to highlight ships that have long voyages and gaps in their signals data. This application, like several similar ones identified in the literature review below, showcases the value of combining data from multiple sources.

## Problem Statement

1. Develop a Machine Learning (ML) model to identify ships in imagery data  
2. Develop a pattern analysis Data Visualization (DV) based on ship navigation signals   
3. Combine these models and data to improve the ML model    
4. For an application of these models and data, propose a risk assessment framework for Forced Labor sometimes present in the fishing industry.

Models like the ones presented in this research can provide insights into the activity of ships conducting fishing, transport, mining, etc. There are efforts to use data science for each of these areas.

## Outline of the Report

1. Lightweight Object Detection in Satellite Imagery Using Convolutional Neural Networks
2. Combined Imagery and Signals Model
3. Application to Forced Labor detection.

To prepare for eventual publication of individual articles, the literature reviews and specific data used will be included in each section.

This report documents the following contributions:

* A new ML ship detection model which can be implemented in open-source software
* An approach to use public-domain signals data to compare with ship detection models
* Analysis of ship voyages based on public-domain signals data to detect risk factors for Human Trafficking (Application)

# Lightweight Object Detection in Satellite Imagery Using Convolutional Neural Networks

**Abstract:** Current efforts in conducting object detection in satellite imagery requires a strong base knowledge of Deep Learning frameworks and respectable computing resources. The ability to perform this type of analysis has matured significantly along with the concepts involved. Deep Learning application of object detection in satellite imagery presents a high cost of entry. This paper presents a Python program that serves as a lightweight solution to conducting object detection in satellite imagery that requires novice Deep Learning intuition and limited computing resources.

## Introduction

The dynamic application of Deep Learning frameworks in the field of Computer Vision has evolved tremendously across domains. Computer Vision is a subset of artificial intelligence that aims at training computers to interpret images and gain a greater understanding of visual processes. This type of digital analysis is being explored by applying machine learning techniques to train algorithms that are capable of improving a human’s ability to process various images and conduct object detection on numerous types of targets.

One domain of research and analysis that poses a need for this type of automation is the field of Geospatial Intelligence and in particular the application to satellite imagery. A large share of actionable intelligence within the Geospatial domain comes from the high-volume acquisition of overhead satellite imagery. The availability of commercial satellite imagery has grown exponentially over the last decade and has produced an enhanced ability to monitor the world at a heightened temporal rate. With the compounding collection of data comes the need to streamline the process of imagery analysis and object detection.

Object detection in high resolution satellite imagery has been explored extensively over recent years and significant benchmarks have been set on improving the ability to detect high interest objects. Target objects typically include vehicles, roads, buildings and vessels. The innovation of vessel detection in high resolution satellite imagery over recent years has produced very high performing, pre-trained models that can be deployed within a user’s computing environment and applied to a personal use case. Though these high performing models produce benchmark results, they also require a significant amount of computing resources and a respectable degree of domain knowledge in Deep Learning frameworks. Analysts within the Geospatial Intelligence domain possess unique skills that allow them to reveal hidden insights in geographic clarity but may not possess the understanding of Deep Learning algorithms that is needed to conduct this type of analysis.

The application presented in this paper will provide the capability to develop and deploy a simpler, more lightweight object detection model that produces accurate results and can be applied to open-source imagery. The user of this application will be able to either produce their own training data or import from another source, develop and train a Deep Learning model using the Keras framework, deploy the model to run predictions on desired imagery, and finally produce an object detection output with geographic attribution. This type of capability within GIS frameworks has been developed in proprietary software such as ArcGIS. However, the ability to conduct object detection is currently limited in open source GIS software such as QGIS. This application presents the capability to perform this type of analysis on an open source platform with minimal Deep Learning understanding using limited computing resources.

## Theoretical Framework and Literature Review

Deep Learning has grown in popularity in the development of artificial intelligence and has generated a significant interest throughout the tech community. Deep Learning as a concept often appears mystifying to those that have not developed a strong foundation in Data Science and Machine Learning. In general, Deep learning is like many other Machine Learning methods in that it takes input data and generates a prediction. The algorithm aims to understand the patterns shared between the inputs and outputs to then be able to generalize the input data that the model has yet to be introduced to. Deep Learning algorithms are composed of neural networks which comprise connections of input, hidden and output layers by nodes. Information is transferred between these layers and once the input is pushed to the output, the model is able to evaluate its performance and make adjustments to weights in the algorithm.

One of the most popular and powerful Deep Learning methods in image processing is Convolutional Neural Networks (CNN). CNNs are able to analyze the importance of nearby pixels in an image by applying a filter of a specified dimension. A value is calculated for each pixel in the image using this filter process through a convolution operation. This type of operation is geared towards extracting features from an image. An early application of this is observed in handwritten digit recognition. A classically known dataset in Deep Learning is MNIST which is made up of 700,000 handwritten digits represented as a grayscale array of 28x28. CNNs have been extensively applied to this dataset. Kumar et.al. [1] compared other approaches to this classification problem by developing a 7-layer Keras sequential model for digit recognition and yielding improved results over other methods such as Hidden Markov model and Multilayer Perceptron. This approach shows the capability to classify image chips based on the features displayed in the pixel values.

Modern application of Deep Learning in satellite imagery takes a more finite approach when it comes to object detection. CNN models are being used to simultaneously detect multiple types of objects in images by developing multi-level algorithms [2]. The availability of very-high resolution satellite imagery has improved the ability the automatically process and analyze for various targets. Guo et.al. [3] showcase the ability to target multiple classes of objects in satellite imagery such as airplanes, vehicles, and baseball fields. This is partly made possible by large training datasets that have been compiled to create benchmark models in conducting object detection on many targets of interest. Rather than classifying an image chip, these benchmark datasets compile annotated images with bounding locations of various object. Guo et.al. proceed to execute a multi-scale CNN that combines the functionality of multiple object detection methods and presents the ability to detect objects location and size of bounding box for each object.

The increased availability of higher resolution imagery, large training datasets [4] such as xView, and heavily constructed pre-trained models has come with the need for robust computing resources and a strong understanding of Deep Learning application. The program presented in this paper leverages the simplicity and lightweight nature of a Keras Sequential model [5][6] as presented in handwritten digit recognition. Rather than classifying image chips of digits, this model will display that ability to detect vessels in satellite imagery using a similar methodology. The technical cost of entry to conducting object detection is discounted and lends to an expanded user population.

## Data and Methods

This paper presents the option to either utilize an existing dataset containing labeled features or create custom training data to fit the users particular use case. Labeled datasets can be found through various sources including Kaggle competitions [7]. The approach taken in this research was to present the ability to create custom training images using simple techniques in open-source GIS software. Given a satellite image that contains that target object, in this case vessels, the user is enabled to label the known location of objects with a centroid point. A square buffer of desired radius is then created around each labeled point. The square bounding box feature is then used to clip the portions of the larger satellite image that contain the target object into individual image chips of the object. These image chips will serve as the positive labeled data.

Next, a similar process is conducted to extract negative labeled images. Rather than labeling point locations of the target object, random points throughout the image are generated and the corresponding bounding box buffers are used to clip image chips of the larger satellite image. This process creates the negative labeled image chips that represent non-object background portions of the image. Given the generation or acquisition of training image chips, the program developed in this project presents the capability to conduct object detection in satellite imagery using a collection of modules that allow the user to define the parameters of input data, develop the training data using the image chips, compiling and training a Deep Learning model, running model predictions across a satellite image using the trained model, and finally producing a geographically attributed dataset that contains bounding boxes of model predictions.

First, the formation of training data is conducted using image chips. This is done by importing the images and cropping to consistent dimensions. Next, the user is given the option to augment the images. Augmenting consists of creating additional training data by transforming an image in a process of flipping, rotating, and transposing. This process presents the ability to create a more robust training dataset by increasing the number of samples at varying orientations. The image chips are then converted to arrays of 8-bit integers that represent the pixel values. These arrays are also known as tensors. A separate array is then created that represents the binary labels of the training data as either containing the object or not containing the object.

Next, the training tensors and labels are compiled, shuffled and split into training and testing data. Additionally, the data values are normalized to a scale of 255. A model is then defined and compiled. The model used in this program is a Keras sequential model with two fully connected Convolutional Neural Networks. An example architecture can be seen in Figure 1. This type of model is flexible and allows the user to customize the parameters, optimizers, activation techniques, and even add additional layers to tune the results of the model. The user will have control over customizing the model or optionally importing a pre-trained model that leverages benchmark results. The model is then trained on the pre-defined training data and optionally saved to avoid continuously training for similar application.

A close up of a sign

Description automatically generated

**Figure 1.** Architecture of Keras CNN with two convolutional layers, two pooling layers, and a fully connected layer.

The trained model is now ready to be leveraged in conducting object detection in a desired satellite image. This program gives the user the option to import satellite images as targets for object detection or alternatively tap into an API to download imagery based on user defined parameters. The API utilized in this program provides access to satellite imagery captured by The European Space Agency Sentinel satellites. This imagery is open source with a valid login. Sentinel provides global imagery at 10-meter resolution. The API is accessed directly from the Python program and available imagery is queried by defining geographic bounds for searching, date range for image capture, as well as desired threshold for cloud cover percentage (typically desired to be <10%). The program is then automatically able to access the product list, download the associated products, and extract the 10-meter true color image from the zipped directory.

With the satellite imagery either uploaded into the program or internally downloaded from the Sentinel API, the object detection portion of the program is prepared for initiation. The satellite image is first transformed into a tensor array of normalized 8-bit integers. The user is given the option to crop the image to specified geographic bounds prior to processing. Additionally, given the prior knowledge of areas to disregard in the image, a mask layer may be imported and applied to the satellite image. In the case of detecting vessels, a mask layer of land features may be imported to exclude non-water pixels from model consideration. An additional processing step may be performed that enhances processing speed in the form of image segmentation. Using K-Means Clustering algorithm, the image pixels can be grouped in n number of classes. This type of processing increases the efficiency of the object detection algorithm by isolating regions of the satellite image that likely contain the object and ignores other regions of the image such as water.

The object detection function conducts predictions of the presence of the target object in the satellite image by implementing a moving window. The moving window essentially scans the image and uses the trained model to conduct a prediction on a portion of the image and continues to traverse until predictions have been made across the entire image. This can be an expensive process depending on how large the image is. The step size in which the window moves across the image may be adjusted to accommodate computing resources. If the prediction value of a window is greater than the user defined threshold, the location coordinates of the window are stored in a list.

Assuming the successful detection of objects in the image, the list of coordinates will likely contain multiple positive predictions for the same object. To address the duplication of detections, each coordinate bounding box is analyzed relative to the overlapping bounding boxes. This is done by calculating the intersection over union (IOU) which is the amount of intersection area relative to the total area of the box. If two boxes have an IOU greater than a desired threshold, the two boxes are assumed to be detecting the same object. The box with the highest prediction score is then stored and the other is discarded. This is conducted for all potentially overlapping predictions to produce a final coordinates list.

Finally, the trimmed results are converted to geographic features. This is done by referencing the original attribution of the satellite image such as geographic bounds, coordinate reference system, and image resolution. The final coordinates are converted into Well Known Text (WKT) strings which are formatted in a way that allows the storing of geographic attribution and ingestion by a GIS software program.

## Impact

One of the primary goals of this program is to provide a customizable yet simple platform to conduct object detection in satellite imagery with minimal knowledge of Deep Learning application. This goal is achieved with great success. The program requires limited user input in terms of parameters and is able to conduct a complete object detection operation with minimal computing resources. The results of the object detection are highly dependent upon the quality and volume of training data that is provided to the program. The program provides the user with the capability to save and compile training data to improve object detection accuracy. The speed at which the algorithm performs depends on the size of the satellite imagery being analyzed. The moving window predictions can be conducted on a 10-kilometer by 10-kilometer image at 10-meter resolution in approximately one minute.

Most modern satellite image object detection benchmarks are achieved using very-high resolution sub-meter imagery which is often not publicly available and expensive to acquire. This program presents a completely open source solution to conducting object detection in satellite imagery with targeted object acquisition in custom training data creation.

## Results

The ability to conduct object detection in 10-meter resolution imagery with high accuracy is proven through testing the program on multiple areas of interest. Image 1 displays successful detection of 35 vessels with an overall image accuracy of 97%. The algorithm was able to process this 225,000 km2 Sentinel-2 image [5] in less than three minutes. Image 3 displays the difficulty in detecting vessels found in coastal locations as the algorithm will often falsely classify objects such as docks. 10-meter, open source imagery is presented as an optional import in this program but the program itself is not limited to high resolution imagery.

A picture containing light, mountain, dark, table

Description automatically generated

**Image 1**

A picture containing green, clock

Description automatically generatedA picture containing fence, outdoor, person, young

Description automatically generated**Image 2 Image 3**

## Conclusion

Given the availability of very-high resolution imagery, this program can be applied to detecting various objects aside from seaborn vessels. The spectral properties and geometric shapes of target objects can be learned the same by this type of algorithm and applied to objects such as airplanes, solar panels, or swimming pools. This program is also not limited to the simplicity of a Keras Sequential model. The framework put into place lends to the introduction of a pre-trained model that yields benchmark results and highly accurate masking of objects.

Though the intention of this program is to be a light introduction to conducting object detection in satellite imagery at minimal computing cost, the foundation is formed to extend the program to state of the art techniques in computer vision and higher lever GPU processing. The tools to conduct object detection in satellite imagery exist in proprietary GIS software. This type of capability presents the notion that Deep Learning concepts in imagery analysis can be achieved across many levels of expertise at a significantly lower cost.

## Next Steps

This program is delivered as a Python module that presents the capability to be executed from the command line. The code can be accessed directly to input custom parameters and utilize various pre-trained models. Though the module requires minimal user control aside from the processing parameters, the need for a software interface is apparent. The code infrastructure is in place to create a custom plugin for an open source GIS software called QGIS. The intentional progression of this project is to develop this capability into a more manageable solution in conducting common Geospatial analysis through a user interface directly deployed in QGIS.

## References

[1] Kumar, K. Senthil, Suman Kumar, and Aabhash Tiwari. Realtime Handwritten Digit Recognition Using Keras Sequential Model and Pygame. No. 3364. EasyChair, 2020.

[2] Cai, Zhaowei, et al. "A unified multi-scale deep convolutional neural network for fast object detection." European conference on computer vision. Springer, Cham, 2016.

[3] Guo, Wei & Yang, Wen & Zhang, Haijian & Hua, Guang. (2018). Geospatial Object Detection in High Resolution Satellite Images Based on Multi-Scale Convolutional Neural Network. Remote Sensing. 10. 131. 10.3390/rs10010131.

[4] Lam, Darius, et al. "xview: Objects in context in overhead imagery." arXiv preprint arXiv:1802.07856 (2018).

[4] Ketkar N. (2017) Introduction to Keras. In: Deep Learning with Python. Apress, Berkeley, CA. <https://doi.org/10.1007/978-1-4842-2766-4_7>

[5] Gulli, Antonio, and Sujit Pal. Deep learning with Keras. Packt Publishing Ltd, 2017.

[6] [Kaggle](https://www.kaggle.com/rhammell/ships-in-satellite-imagery) "Planet Team (2017). Planet Application Program Interface: In Space for Life on Earth. San Francisco, CA. https://api.planet.com."

[7] [Copernicus](https://www.esa.int/Applications/Observing_the_Earth/Copernicus) Sentinel data 2018, processed by ESA.

# **Combined Imagery and Signals Model**

By combining the signals and imagery data, we can write a model that benefits from both types of data. As a quick recap of these advantages, the imagery is effective at detecting ships regardless of whether they are transmitting navigational signals, however it relies on clear weather and the collection of it is more sparse and only in daylight. On the other hand the signals data does not rely on clear weather and is equally prevalent at night, plus, in addition to detection, also provides the identity of the ship from a combination of the signal and lookup tables of ship registries for each country. So, the signals data is more useful in “tracking” applications of maintaining awareness of particular vessels.

The disadvantage of the AIS data is that some ships may not meet their country of registrations requirements for transmitting AIS (and so may not even have the equipment). A typical requirement for AIS is ships of over 30 gross tons. Even for ships that do meet their countries requirements, there is little means to enforce that they transmit when outside of their country’s sovereign waters. Notably, GFW documented the extent of this problem (2016) and the French news source *24* in 2019. In these cases, remote sensing using other types of sensors is important, rather than relying only on AIS.

# Data Sources

|  |  |
| --- | --- |
| Data | Source |
| Sentinel-2 Satellite Imagery | From Copernicus (European Space Agency) and where not found there, from Google Cloud Public Data |
| Sentinel-2 Index | Google Cloud Public Data |
| 2011-2016 AIS data | Global Fishing Watch (GFW) Public Data |
| 2018 AIS data | Global Fishing Watch (GFW) “Dark Fleet” Public Data |
|  |  |
|  |  |

For the results in Figure 2, all of the Sentinel-2 imagery data in December 2018 was searched (using an index available in Google Big Query) for an area spanning the cost of China, Russia, and North Korea. The images reported along the x-axis have 10% or less cloud cover and are daylight observations, which is required for the Ship Detection model.

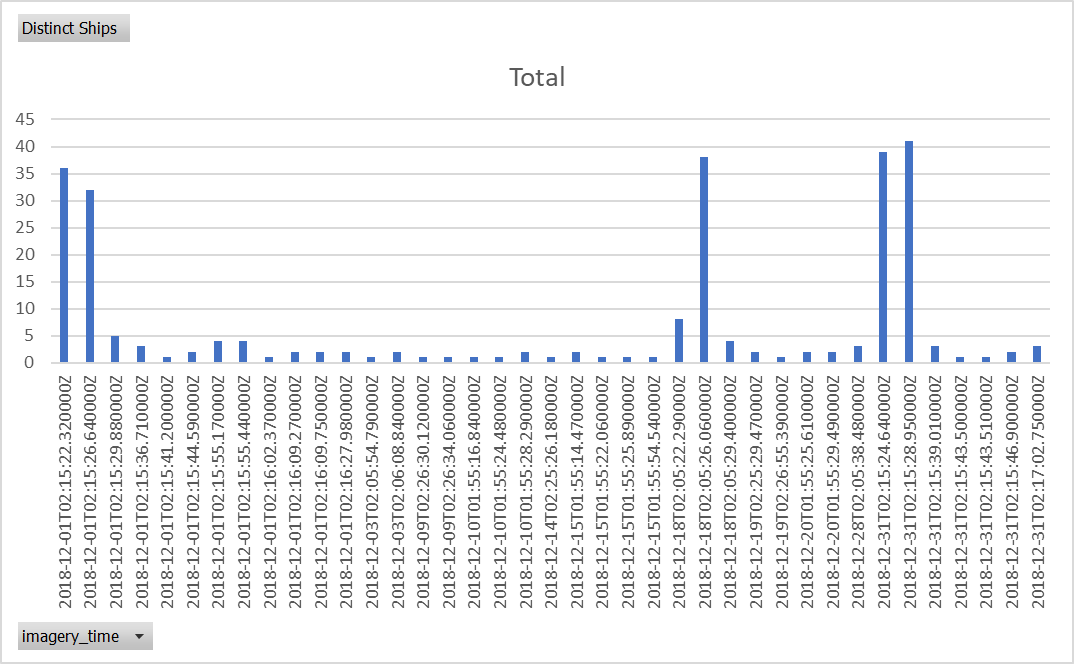


Figure 2 Simultaneous Signals & Imagery data in Dec 2018

For each of these images, the corners of the image were used to define a search box for signals data that was recorded within a two hour time window of the image. The blue bar represents the number of unique ships transmitting one or more signals within two hours of the image.

Figure 3 Combined View of Signals and Imagery shows the image collected with boxes to annotate the ships detected by the Object Detection Model and red dots to reflect the signals data. In many cases there are multiple red-dots on top of each other. In cases where the signals data was dense, only the closest observation time to the image is displayed.

Also note that in Figure 3, the masking of the land area by the object detection model means that it does not detect six of the ships that are docked and instead considers them part of the land or pier. This deficit was covered in the last section as well.

The analysis can be repeated for the remaining images, but even the one image proves the point that the ship detection algorithm is validated by the signals data and vice versa. The other important point to make is that the combination of both imagery and signals means that the area that can be analyzed by this combination of models includes night and bad weather (when the imagery is not effective) and open-ocean where imagery is not typically tasked because most of the image would be water, yielding little information. This is particularly true of the Sentinel-2 satellite since its multi-spectrum sensor package and ability to yield color images is better spent on images where the spectrums can be used to study crop growth and other changes on land. However, as demonstrated in this paper, the Sentinel-2 capabilities (especially its 10-meter resolution) enable efficient ship detection.



Figure 3 Combined View of Signals and Imagery

The results our joint model are shown in the table below for the images with the most ships shown by AIS data (the tallest blue bars in Figure 2). <May replace this table with a randomized sample taken from all of 2018.>

Table 1 Sample of Cloud-free images & corresponding signals

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Image Timestamp | AIS Distinct Ships | Object Detection | AIS but no object detected | Object detected but no AIS signal |
| 2018-12-31T02:15:28.950000Z | 41 |  |  |  |
| 2018-12-31T02:15:24.640000Z | 39 |  |  |  |
| 2018-12-18T02:05:26.060000Z | 38 |  |  |  |
| 2018-12-01T02:15:22.320000Z | 36 | 39 | 9 | 21 |
| 2018-12-01T02:15:26.640000Z | 32 |  |  |  |

## Impact

This combined imagery & signals model shows the advantage of adding additional data sources. From Table 2 and Figure 3, the combined data shows ships in port confused in the object detection with the land and pier structure, as well as small ships anchoring in port not transmitting a navigation signal within an hour of the image. The converse is that the total number of the cloud-free images from the data source used was X,XXX for all of 2018, each representing an instantaneous, daylight view of a small portion of the area studied. For the ships that transmit navigational signals, the coverage is almost continuous.

## Next Steps

Adding additional satellite and terrestrial sources will continue to improve results for both aggregate and detailed analysis of behavioral patterns of ships. Also, a more careful analysis of the AIS data itself, which has identification as well as positional data imbedded has uses as we demonstrate in the next section on an application to Human Trafficking in the fishing industry.

# Application to Forced Labor Risk Factors

According to Liberty Shared, data on Forced Labor both domestically and internationally is practically non-existent. This is not for a lack of concerned parties (both governments and non-governmental organizations (NGOs). In the case of forced labor on Fishing vessels there are many complicating factors that are explained in this section and the references.

Traditional wisdom is that Forced Labor is generally aligned with more traditional commerce, so at the end of the supply chain, the goods are sold using traditional currencies to customers and companies that operate in the open and are willing to stop buying a product or service once a legitimate claim is made about forced labor. This stands in stark contrast to sex labor which is often has ties to organized crime and crypto currencies. A reference on this is the recent report by the Financial Threats Council of the Intelligence and National Security Alliance (published May 2020) entitled “Using Intelligence To Combat Trade-Based Money Laundering.”

Modelling Approach:

* Correlate AIS observations for the same unique ship ID into voyages
* From the correlated data, Identify Human Trafficking risk factors, which include
  1. Gaps in signals (both time gaps and distance gaps)
  2. Longer voyages than most vessels of the same type
* Additional research which may be possible with a full AIS dataset. Spire provided a sample for this research for one day in 2019.
  1. Changes in registration (this was not possible because of anonymization)
  2. Operations of Cargo Ships, referred to as “Reefers” which is short for refrigeration ships that can join with the fishing ship at sea to swap it’s cargo for supplies and sometimes new crew members. Reefers were not included in the dataset.

# Data Sources

|  |  |
| --- | --- |
| Data | Source |
| 2011-2016 AIS data | Global Fishing Watch (GFW) Public Datasets in Google Cloud, which provides a sample of 28.5M AIS signals and anonymized information on the associated vessels. |
| 2018 AIS data | Global Fishing Watch (GFW) “Dark Fleet” Public Datasets in Google Cloud, which cover individual AIS signals and information on the categories of vessels. |

## Literature Review

The research presented in this paper builds on recent Big Data analysis undertaken on data on ship registrations, satellite data, and national fishing activity databases.

The application to Human Trafficking and Forced Labor builds on several efforts by international organizations to combat this human rights violation. Among those is the Associate Press which published a series of articles under the heading “Seafood from Slaves” for which they won the Pulitzer Prize for Public Service on April 18, 2016. The articles and actions generated by them freed 2,000 slaves, according to the AP Press Release that day.

Of these articles, McDowell, Mendoza, and Mason [2015] reports on work by Kroodsma, Miler, and Roan [2017] to use AIS data to deduce likely rendezvous with smaller fishing vessels (which may have slave crews) with transshipment ships. Their work generated a global map showing these likely transhipments and analytic insights on the countries of registration for both the fishing vessel and the transport ship, as well as the next port visited to presumably offload the cargo. Many of these transhipments were in or near areas with fishing restrictions in place; therefore are widely viewed as likely violations of the fishing laws, according to this series.

This series of articles, and the underlying research they reported, uncovered a few fishing ships that were seized and proven to Human Trafficking violations. Their Forced Labor crews were freed. But, as reported these examples of rescues are considered a tiny fraction of the human rights violations in the fishing industry. These articles also describe the difficulty that nations have enforcing their laws and show cases where if one country cracks down, the offending ships simply move outside of that countries jurisdiction.

Table 2 Risk Factors used

|  |  |
| --- | --- |
| **Risk Factor** | **Analysis Approach** |
| Long voyages | Based on the ships identification number, separated the AIS observations into voyages between port visits. Determined which of the ships had voyages longer than average for their vessel type (in the top quartile). |
| Time Gaps between signals | Within each voyage computed the time delta between each signal and assessed a risk for ships with average gaps in the top quartile for their vessel type |
| Distance Gaps between signals | Within each voyage computed the distance delta between each signal and assessed a risk for ships with average gaps in the top quartile for their vessel type |

Using Google Datalab, Google BiqQuery, and Python, the data for each voyage was aggregated for the risk factors, then a further aggregation was made across the voyages for 2018 for each ship. The resulting trends can be observed in the Sankey diagram below which shows the pattern at an even higher level by the fishing vessel types.

A screenshot of a cell phone

Description automatically generated

To identify specific ships, these risk factors can be reported out using a scheme such as demonstrated in the Table below.

# Analysis of Results

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Future Direction

In work

## References

Jaeyoon Park, Jungsam Lee, Katherine Seto, Timothy Hochberg, Brian A. Wong, Nathan A. Miller, Kenji Takasaki, Hiroshi Kubota, Yoshioki Oozeki, Sejal Doshi, Maya Midzik, Quentin Hanich, Brian Sullivan, Paul Woods, David A. Kroodsma, Illuminating Dark fishing fleets in North Korea," Science Advances, July 22, 2020.

Kroodsma, D.A., N.A. Miller, and A. Roan 2017. “The Global View of Transshipment and Revised Preliminary Findings, Global Fishing Watch and SkyTruth, July 2017. Available online at <http://globalfishingwatch.org>.

“An AP investigation helps free slaves in the 21st century,” a series of articles over 18 months for which they won the Pulitzer Prize for Public Service on April 18, 2016.

McDowell, R., M.Mendoza, and M. Mason. 2015. AP tracks slave boats to Papua New Guinea available at https://www.ap.org/explore/seafood-from-slaves/ap-tracks-slave-boats-to-papua-new-guinea.html

Greenpeace. 2016. Turn the tide. Human Rights Abuses and Illegal Fishing in Thailand’s Overseas Fishing Industry. <http://www.greenpeace.org/australia/Global/australia/reports/Turn-The-Tide.pdf>

Cutlip, K. 2017. Skytruth Blog. <http://skytruth.org/2017/01/satellites-leave-no-place-to-hide-forrogue-thai-fishing-fleet/>

Investigating how North Korean fishermen plunder foreign waters, *France 24*, a French state-owned international news television network, May 8, 2020 available at <https://www.youtube.com/watch?v=ytbEt8do3r8>