big data and cognitive computing

MDPI

Article

Near-Real-Time IDS for the U.S. FAA's NextGen ADS-B

Dustin M. Mink [1], Jeffrey McDonald [1], Sikha Bagui [2,*], William B. Glisson [3], Jordan Shropshire [1], Ryan Benton [1] and Samuel Russ [1]

[1] School of Computing, University of South Alabama, Mobile, AL 36688, USA; dmink@uwf.edu (D.M.M.); jtmcdonald@southalabama.edu (J.M.); jshropshire@southalabama.edu (J.S.); rbenton@southalabama.edu (R.B.); sruss@southalabama.edu (S.R.)
[2] Department of Computer Science, University of West Florida, Pensacola, FL 32514, USA
[3] Department of Computer Science, Sam Houston State University, Hunstville, TX 77340, USA; glisson@shsu.edu
* Correspondence: bagui@uwf.edu

Abstract: Modern-day aircraft are flying computer networks, vulnerable to ground station flooding, ghost aircraft injection or flooding, aircraft disappearance, virtual trajectory modifications or false alarm attacks, and aircraft spoofing. This work lays out a data mining process, in the context of big data, to determine flight patterns, including patterns for possible attacks, in the U.S. National Air Space (NAS). Flights outside the flight patterns are possible attacks. For this study, OpenSky was used as the data source of Automatic Dependent Surveillance-Broadcast (ADS-B) messages, NiFi was used for data management, Elasticsearch was used as the log analyzer, Kibana was used to visualize the data for feature selection, and Support Vector Machine (SVM) was used for classification. This research provides a solution for attack mitigation by packaging a machine learning algorithm, SVM, into an intrusion detection system and calculating the feasibility of processing US ADS-B messages in near real time. Results of this work show that ADS-B network attacks can be detected using network attack signatures, and volume and velocity calculations show that ADS-B messages are processable at the scale of the U.S. Next Generation (NextGen) Air Traffic Systems using commodity hardware, facilitating real time attack detection. Precision and recall close to 80% were obtained using SVM.

Keywords: Next Generation (NextGen) Air Transportation Systems; Automatic Dependent Surveillance-Broadcast (ADS-B); Intrusion Detection System (IDS); network attack signatures; data mining process; Support Vector Machine (SVM); big data

Citation: Mink, D.M.; McDonald, J.; Bagui, S.; Glisson, W.B.; Shropshire, J.; Benton, R.; Russ, S. Near-Real-Time IDS for the U.S. FAA's NextGen ADS-B. *Big Data Cogn. Comput.* **2021**, *5*, 27. https://doi.org/10.3390/bdcc5020027

Academic Editors: Isaac Triguero and Min Chen

Received: 8 May 2021
Accepted: 3 June 2021
Published: 16 June 2021

1. Introduction

At peak operational times, there are 5000 concurrent flights in the U.S. national airspace [1]. The U.S. Federal Aviation Administration (FAA) indicates that the U.S. Gross Domestic Product (GDP) will increase from $16.3 to $26.2 trillion U.S. dollars from 2015 to 2036 [2] respectively. The U.S. Government Accountability Office (GAO) states that the aviation industry is in the process of implementing the U.S. Next Generation (NextGen) Air Traffic System [3–7]. The NextGen component programs are at various stages of development and include Automatic Dependent Surveillance-Broadcast (ADS-B), Collaborative Air Traffic Management Technologies (CATMT), Data Communication, National Airspace System Voice System, NextGen Air Transportation System Weather, and System Wide Information Management (SWIM). The GAO states that a major element of the system is the ADS-B capability, which is seen to be the future of air traffic control through advancements in aircraft tracking and flow management.

The ADS-B system augments traditional radar and transponder surveillance with ADS-B messages and embedded positioning via Global Positioning Systems (GPS). ADS-B is an unencrypted system, meaning that the national airspace system (NAS) is susceptible to a variety of cyber-physical attacks [8–15]. Though the focus of this research is on U.S.

National Airspace, with the investigation of solutions for ADS-B, these solutions could also apply to the EU, since the EU uses the same technologies.

The goal of this paper is to determine flight patterns, including patterns for possible attacks, using a data mining process in the context of big data. Flights outside the flight patterns are possible attacks. In this work, OpenSky provides the data source of ADS-B messages, NiFi provides data management, Elasticsearch is the log analyzer, Kibana visualizes the data for feature selection, and a Machine Learner, Support Vector Machines (SVM), is used for classification. This work shows that ADS-B network attacks can be detected using network attack signatures, and volume and velocity calculations show that ADS-B messages are processable at the scale of NextGen. The volume and velocity of the ADS-B network data are presented in the context of required computing resources and hence will facilitate taking appropriate action on attack detection in real time, improving flight safety.

1.1. Big Data

Big Data is defined by five characteristics, also known as the 5 v's: volume, variety, velocity, veracity, and value. Data volume measures the scale of the data within the system. Data variety refers to the different structures and sources of data. Data velocity is the analysis of the data as the data are generated. Data veracity illustrates the uncertainty of the data, and data value is the evaluation of the impact the data has on research [16].

1.2. Data Mining and Machine Learning

Data Mining (DM) searches for patterns or correlations that provide understanding or predictive power [17,18]. Machine Learning (ML) is a class of computer algorithms that allow computers, without being explicitly programmed, to learn and classify information and recognize patterns. This work uses the SVM machine learner to classify information and recognize patterns. SVM, which is more computationally intensive than many other less sophisticated classification algorithms, has the advantage of working well on datasets that are not linearly separable. SVM finds the best hyperplane that separates observations of one class from those of the other class. The best hyperplane is the one with the largest margin between two classes of observations [19].

1.3. Mitigation Solution Being Addressed

The two mitigation solutions used to-date with ADS-B include intrusion detection [20–22] and encryption Public Key Infrastructure [8,23–25] implementations. Existing literature on NextGen security focuses largely on encryption, while minimal research investigates big data NextGen solutions [20–25].

The novelty of this research is in providing a third alterative solution for attack mitigation by packaging a machine learner, SVM, into an intrusion detection system and further calculating the feasibility of processing U.S. ADS-B messages in near real time.

The remainder of this paper is organized as follows. Section 2 presents the background, that is, the mechanics of ADS-B. Section 3 presents related works on ADS-B, focusing on works that look at attack mitigation strategies and techniques and works done in the context of Big Data and Machine Learning. Section 4 presents the methodology, including the system architecture and the data mining process used. Section 5 presents the results and discussion. Results are presented in terms of visualization and machine learning results as well as data volume and velocity. Section 6 presents the conclusion and Section 7 is a future works section.

2. Background: Automatic Dependent Surveillance-Broadcast

This section presents the mechanics of ADS-B. The message types of U.S. NextGen Air Transportation are Mode A, Mode C, Mode S, and ADS-B In and Out [26–28]. Mode S has three message types: (i) Data Block Surveillance Interrogation and Reply Message Format; (ii) Data Block Surveillance and Communication Interrogation and Reply—Communication-

A and Communication-B Message Format; and (iii) Data Block Surveillance Communication Interrogation and Reply—Extended Length Message Format. The ADS-B system inherits its message types from Mode S; hence, ADS-B has three different message types.

The Mode S Data Block Surveillance Interrogation and Reply Message Format comprises three parts [26–28]: Format Number, Surveillance and Communication Control, and Address and Parity, as shown in Table 1. The Format Number is a 5-bit message representing the sequence number of the message. The Surveillance and Communication Control is a 27-bit message, which includes commands and flight information. The Address and Parity is a 24-bit message, intended to represent a unique aircraft identifier.

Table 1. Mode S Data Block Surveillance Interrogation and Reply Message Format [29].

Format Number	Surveillance and Communication Control	Address and Parity
5 bits	27 bits	24 bits

The Mode S Data Block Surveillance and Communication Interrogation and Reply—Communication-A and Communication-B Message Format comprises four parts: Format Number, Surveillance and Communication Control, Message Field, and Address and Parity, as shown in Table 2. The Format Number is a 5-bit message representing the sequence number of the message. The Surveillance and Communication Control is a 27-bit message, which includes command and flight information. The Message Field is a 56-bit message that contains additional flight information. The Address and Parity is a 24-bit message, intended to represent a unique aircraft identifier [26–28].

Table 2. Mode S Data Block Surveillance and Communication Interrogation and Reply—Communication-A and Communication-B Message Format [29].

Format Number	Surveillance and Communication Control	Message Field	Address and Parity
5 bits	27 bits	56 bits	24 bits

The Mode S Data Block Surveillance Communication Interrogation and Reply—Extended Length Message Format comprises four parts [26–28]: Format Number, Communication Control, Message Field, and Address and Parity, as shown in Table 3. The Format Number is a 2-bit message representing the sequence number of the message. Communication Control is a 6-bit message, which includes commands. Message Field is an 80-bit field that contains additional flight information. Address and Parity is a 24-bit message, intended to represent a unique aircraft identifier.

Table 3. Mode S Data Block Surveillance Communication Interrogation and Reply—Extended Length Message Format [29].

Format Number	Surveillance and Communication Control	Message Field	Address and Parity
2 bits	6 bits	80 bits	24 bits

The ADS-B message field can contain information on traffic, weather, and flights. ADS-B vulnerabilities pertain to confidentiality, integrity, and availability. Anyone with an ADS-B radio can transmit and receive messages, thus precluding confidentiality. Data integrity is affected by attacks such as Ghost Aircraft Injection, Aircraft Disappearance, Virtual Trajectory Modification, and Aircraft Spoofing. Ghost Aircraft Injection occurs when an ADS-B radio transmits fake messages and other aircraft think there is an aircraft. Aircraft Disappearance happens when skillfully timed malformed ADS-B messages are sent

with a real aircraft's identification, resulting in ADS-B messages with the particular aircraft to be disregarded. In other words, the remaining aircraft do not believe this particular aircraft exists. Virtual Trajectory Modification is the act of jamming an aircraft or ground station to create false alarms. Aircraft Spoofing is using another aircraft's identification to send ADS-B messages with false information. Finally, availability is a loss associated with Ground Station and Ghost Aircraft Flooding. Ground Station Flooding occurs when ground-based radios are jammed. Ghost Aircraft Flooding happens when a large number of fake ADS-B messages are sent. This makes it such that there are too many real and fake aircraft, and nothing is distinguishable [8–15]. This research will build on the already well-documented vulnerabilities of ADS-B by reproducing these vulnerabilities at the data layer and mixing these messages with OpenSky ADS-B messages.

3. Related Works

Ref. [9] discusses ADS-B as a new technology for air traffic monitoring. This holds the promise of achieving high precision and is envisioned to replace conventional radar systems. Several works have looked at the security issues related to ADS-B. Ref. [10] looked at both the theoretical and practical efforts that have been used for ADS-B security protocols and discussed the inherent lack of security measures in ADS-B protocols. Refs. [11,13–15] shed further light on the practicality of different threats on ADS-B. Ref. [26] examined encryption schemes and discussed the challenges associated with implementing the encryption schemes for the ADS-B environment. Ref. [27] described a low-cost solution for ADS-B–based real-time air traffic monitoring systems implemented on a software-defined radio platform. This provided an integrated hardware and software solution for rapidly prototyping high-performance wireless communications systems. Ref. [28] constructed a schema for batch verification of ADS-B systems. Ref. [20] developed a passive fingerprinting technique that accurately and efficiently identifies wireless implementations by exploiting variations in transmission behavior. Ref. [8] addressed the mitigation solution by building an authentication framework through introducing a new online/offline identity-based signature scheme. The scheme introduced by Ref. [8] resolved the public-key infrastructure issue by using the identities of aircrafts as public keys.

3.1. Works Related to ADS-B in the Big Data Environment

Research shows that a Hadoop-based solution analyzes billions of ADS-B radio messages in approximately 35 min [30]. The results of this research were visualized using density maps. For the betterment of the solution in the context of cybersecurity or digital forensics, the ability to filter messages according to keywords or phrases would reduce noise [31]. A reduction in computational times would assist with real-time processing aspirations.

Ref. [29] provides insight into unaddressed Big Data issues in NextGen, such as identifying issues with the velocity, variety, and veracity of NextGen. SWIM is the data sharing digital backbone of NAS for NextGen; it does not address the veracity of the data received via the ADS-B protocol. NAS has a single point of information in near real-time speed. SWIM includes a surface visualization tool, which allows the Air Traffic Center (ATC) to manage surface traffic.

Ref. [30] presents Big Data platforms that are able to process ADS-B messages. This work shows that conceptually, using a statistical methodology, and in the context of big data, ADS-B messages are processable at the scale of NextGen.

3.2. Works Related ADS-B Using Machine Learning

Ref. [32] presents Spoofing Detector for ADS-B (SODA), a two-stage Deep Neural Network (DNN)–based machine learner that detects ADS-B spoofing messages and hence is also an aircraft classifier [32]. Experimental results show that SODA detects ground-based spoofing attacks with a precision of 99.34%, with a false positive rate of 0.43% [32]. SODA outperforms other machine learning techniques, such as XGBoost, Logistic Regression, and SVM. It also identifies individual aircraft with an average F-score of 96.68% and an

accuracy of 96.66% [32]. Though SODA DNN is a very promising machine learner, it does not address the feasibility of processing U.S. ADS-B messages in near real time, which this research evaluates.

3.3. Addressing the Gap in the Research

DM/ML is used in cybersecurity and is beginning to be leveraged for the intersection of cybersecurity and ADS-B. However, studies have not focused on feasibility in the context of aviation and Big Data and the required computing resources, which this paper assesses. In addition to providing a solution for attack mitigation by packaging a machine learner, SVM, into an intrusion detection system, none of the previous works look at the feasibility of processing ADS-B messages in near real time, which this study addresses.

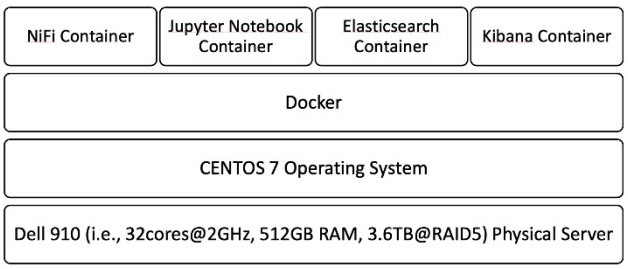
4. Methodology

This section presents the system architecture, the data capture and data engineering, and finally the data mining process.

4.1. System Architecture

The system architecture of this environment consists of users, hardware, and software interfaces.

As shown in Figure 1a, the user interface is a HyperText Markup Language version 5 (HTML5) presentation of NiFi, Kibana, and Jupyter Notebooks via an SSH tunnel to the Docker container exported ports. The hardware interface includes a Dell 910. The software interface includes CentOS and Docker. Docker orchestrated the creation of four required servers to conduct the experiment: Elasticsearch, Kibana, Apache NiFi, and Jupyter Notebook. Configuration changes to NiFi were required due to the vast size of the JSON files. Since the default memory setting of 512 MiB for Java Virtual Machine (JVM) is not capable of deserializing JSON OpenSky ADS-B messages, memory was increased to 30 GiB. These configuration changes allowed the NiFi processors to process JSON files. Figure 1b presents the specifications of the architecture.



(a)

Layer	Name	Version
Containers	Apache NiFi	1.7.1
	Elasticsearch	6.4.2
	Kibana	6.4.2
	Jupyter Notebook Scientific Python Stack with Tensorflow	59b402ce701d
Services	Docker	17
Operating System	CENTOS	7
Hardware	CPU	32cores@2GHz
	RAM	512GiB
	Storage	3.6TiB@10K RAID5

(b)

Figure 1. (**a**) Overview of the system architecture. (**b**) Specifications of the architecture.

4.2. Data Capture and Data Engineering

One of the biggest gaps in this kind of research is the availability of labelled data. Data were downloaded from OpenSky ADS-B archives for 24 h on 11 March 2019. NiFi flow created and fetched each hour-long archive with a GetFile NiFi Processor, followed by an unpackcontent NiFi processor to untar the archive (Figure 2). The archives consisted of several files, for which routeonattribute NiFi processor was used to forward only GZip compressed files that contained the actual ADS-B messages. The splitJSON NiFi processor takes the JSON list of ADS-B JSON objects and flattens them into individual JSON objects in its NiFi flow file. The replacedtext NiFi processor properly formats OpenSky JSON to a JSON format accepted by JSON Search. PutElasticsearchHttp NiFi processor then inserts

the NiFi flow files into Elasticsearch using the Elastic Search REST API. After data ingest into Elasticsearch, Kabana visualizes the data using line charts, displaying the minimums, maximums, and peaks (Figure 3). Recording the minimum and maximum values of each field, the ADS-B attack generator is more effective.

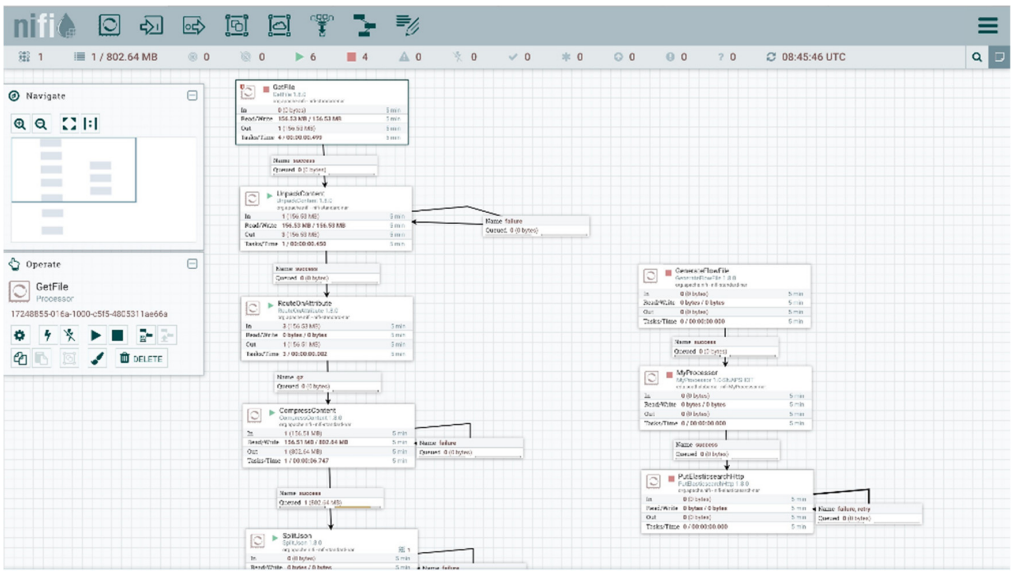


Figure 2. Apache NiFi Dataflow for ingest of OpenSky ADS-B Traffic and Generated Attack Traffic.

Figure 3. Elastic Kibana Visualizations of Statistics of OpenSky ADS-B Traffic.

After inspection of OpenSky data, the work required the creation of a traffic generator using NiFi to create the ADS-B spoofing attack at the data layer. Since there is no ADS-B traffic generator in the NiFi Library, a custom NiFi processor was required and was built using Java. To create the build environment, a maven Project Object Model (POM) build file was necessary. Using Java allowed the fields within the ADS-B message to be configurable in NiFi. The fields within the ADS-B are time, latitude, longitude, velocity, vertrate, barometric altitude, and geoaltitude. The minimum and maximum values were specified within the NiFi processor, and this allowed more effective attacks. GenerateFlowFile NiFi processor used a predetermined amount of ADS-B messages. MyProcessor custom NiFi processor set all the values in the message. PutElasticsearchhttp NiFi processor sent messages to Elastic Search via REST API. With enough values randomly generated over time, the values spread evenly between the minimum and maximum values for all features (Figure 5).

4.3. The Data Mining Process

This section is divided into data preprocessing, extraction of patterns using data mining, and post processing of data, that is, what was done to present the findings.

4.3.1. Data Preprocessing

The first step in the DM process is data preprocessing. Data preprocessing includes data cleaning, data integration, and data transformation [17,18]. The first step of data cleaning is the removal of noise and data inconsistencies. In this work, this first step of data cleaning was accomplished by removing events with the same aircraft unique identifier by dropping additional rows with the same icao24. As part of data integration, the Pandas join (Figure 2) was used. OpenSky ADS-B traffic was combined with NiFi generated attack ADS-B network traffic. From this dataset, all features were analyzed for feature selection by graphing each feature using the Kibana visualization tool. Distinct patterns were found in velocity, baroalititude, geoalititude, vertrate, and geo.

The data were cleaned using the ELK stack [33]. ELK is an end-to-end technology stack providing a complete analytical solution. Since neither baroalititude nor geoalititude showed any statistical advantage over the other, baroalititude was selected, along with velocity and vertrate. Geo, while statistically relevant, would require significant preprocessing using time-series analysis techniques and therefore was not selected as a feature. ADS-B events with no values in baroalititude, velocity, and vertrate were filled with nan values, and rows with nan values were dropped. –index field is a built-in Elasticsearch field that contains the name of the index. The index schema used in this work assigned each hour of the day with its own index. Since values within –index are strings, and ML requires numeric values, one-hot-encoding was used to assign the numeric value of zero to ADS-B benign or normal network traffic and the numeric value of one to ADS-B attack network traffic. Finally, the data set was split 50/50 for training and testing.

Data selection is the retrieval of relevant data from the data sources. For data selection, a custom NiFi processor was used to ingest the OpenSky ADS- B network traffic data (Table 4) via JSON REST API [34,35] or JSON flat file [36]. Table 4 presents the OpenSky ADS-B JSON Object Definitions (data structure and definitions). Apache NiFi comes with approximately 260 processors, providing a range of processes such as to get, convert, and put. However, Apache NiFi does not come with an ADS-B traffic generator able to produce known attacks on ADS-B networks such as spoofing or injection. Since Logstash, provided by Elasticsearch, is not as versatile for complete data preprocessing since it only ingests data, a custom NiFi processor was the best choice for this work.

Table 4. OpenSky ADS-B JSON Object Definition [35].

Index	Property	Type	Description
0	icao24	String	Unique ICAO 24-bit address of the transponder in hex string representation.
1	Callsign	String	Callsign of the vehicle (8 chars). Can be null if no callsign has been received.
2	Origin county	String	Country name inferred from the ICAO 24-bit address.
3	Time position	Int	Unix timestamp (seconds) for the last position update. Can be null if no position report is received by OpenSky within the past 15 s.
4	Last contact	Int	Unix timestamp (seconds) for the last update. This field is updated for any new valid message received from the transponder.
5	Longitude	Float	WGS-84 longitude in decimal degrees. Can be null.
6	Latitude	Float	WGS-84 latitude in decimal degrees. Can be null.
7	Geoaltitude	Float	Geometric altitude in meters. Can be null.
8	On ground	Boolean	Boolean value which indicates if position was retrieved from a surface position report.
9	Velocity	Float	Velocity over ground in meters per second. Can be null.
10	True track	Float	True track in decimal degrees, clockwise from north (north = 0 degrees). Can be null.
11	Vertical rate	Float	Vertical rate in meters per second.

The third step of data preprocessing—data transformation—is the consolidation of data into appropriate forms for DM by the aggregation of the data [17]. The data were transformed for DM by a custom enrichment NiFi processor.

4.3.2. Extracting Patterns Using Data Mining

The second step of the DM process is the employment of intelligent methods to extract patterns from the data. Classification techniques have the ability to group data with similarities, such as attacks, hence this was considered the best option for processing this data. In this work, classification techniques using a custom NiFi processor were used. A customized NiFi processor was used for creating, sending, receiving, transforming, routing, splitting, merging, and processing FlowFiles. FlowFiles, pieces of data that the user brings into NiFi for processing and distribution, consist of two parts: Attributes and Content. Attributes are key-value pairs that are associated with User Data. Content is user data. The custom NiFi processor is an algorithm written in Python using DM/ML libraries to process FlowFiles.

Jupyter Notebook fetched 10,000 real ADS-B messages from Elasticsearch REST API and 10,000 generated attack ADS-B messages (Figure 3). The results were combined into one pandas data frame for training and testing (Figure 4). Preprocessing dropped all rows with the same icao24. The drop method removed all fields except velocity, baroaltitude, vertrate, and index. The –index field determines if the row is an OpenSky collected ADS-B message or a NiFi Custom Processor generated attack. The replace method filled empty fields with the null value of np.nan. The dropna method dropped all fields with np.nan. One-hot encoding replaced the labels within the index numerical representation. Velocity, baroaltitude, and vertrate were numerical values and hence were not one-hotencoded.

4.3.3. Post-Processing

The third step of the DM process is post-processing. Post-processing includes pattern evaluation and knowledge representation. Pattern evaluation, expressed by the comparison of test data and labelled data, identifies relevant patterns leading to knowledge based on interestingness measures. The second step of post-processing—knowledge presentation—is the visualization of the knowledge for presentation to users. The knowledge is presented using Kibana from ELK for visualization [37].

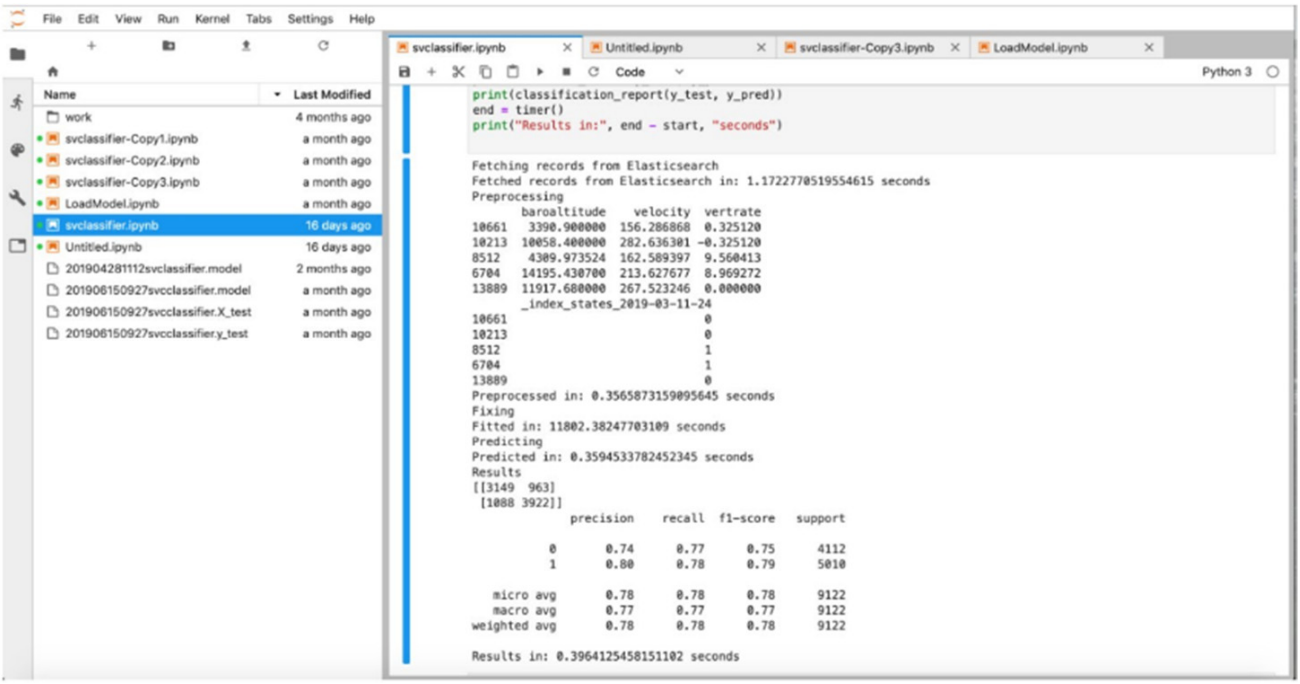


Figure 4. Jupyter Output of Sample Data and Confusion Matrix.

5. Results and Discussion

The results are presented in terms of visualizations, machine learning results, and volume and velocity calculations.

5.1. Data Exploration

For data exploration, Kibana visualization was used to categorize the characteristics of OpenSky ADS-B Traffic (Figure 5). The features selected were velocity, baroaltitude, and vertrate. Velocity has a minimum of zero and a maximum of 324.844. Baroaltitude has a minimum of −335.28 and a maximum of 36,941.762. Vertrate has a minimum of −41.615 and a maximum of 28.611. The distinct spikes indicate a definite pattern in the data, indicative of flight patterns in U.S. National Air Space.

Kibana visualization was also used to categorize the characteristics of generated ADS-B network attacks. The features selected were also velocity, baroaltitude, and vertrate. The ADS-B attack data are randomly generated. Given a significant count of ADS-B messages, the visualization presents the pattern using averages (versus peaks). For example, flooding airspace would not be indicative of U.S. NAS flight patterns. Each feature's minimum and maximum from the ADS-B traffic was used as the minimum and maximum for the randomly generated network attack ADS-B messages. The minimums and maximums from Kabana visualizations of generated ADS-B from OpenSky are similar to Kabana visualizations of generated ADS-B network attacks. The data from Kabana visualizations of generated ADS-B network attacks show velocity has a minimum of 0.075 and a maximum of 320.887. Baroaltitude has a minimum of −303.888 and a maximum of 38,069.102. Vertrate has a minimum of −28.284 and a maximum of 27.634.

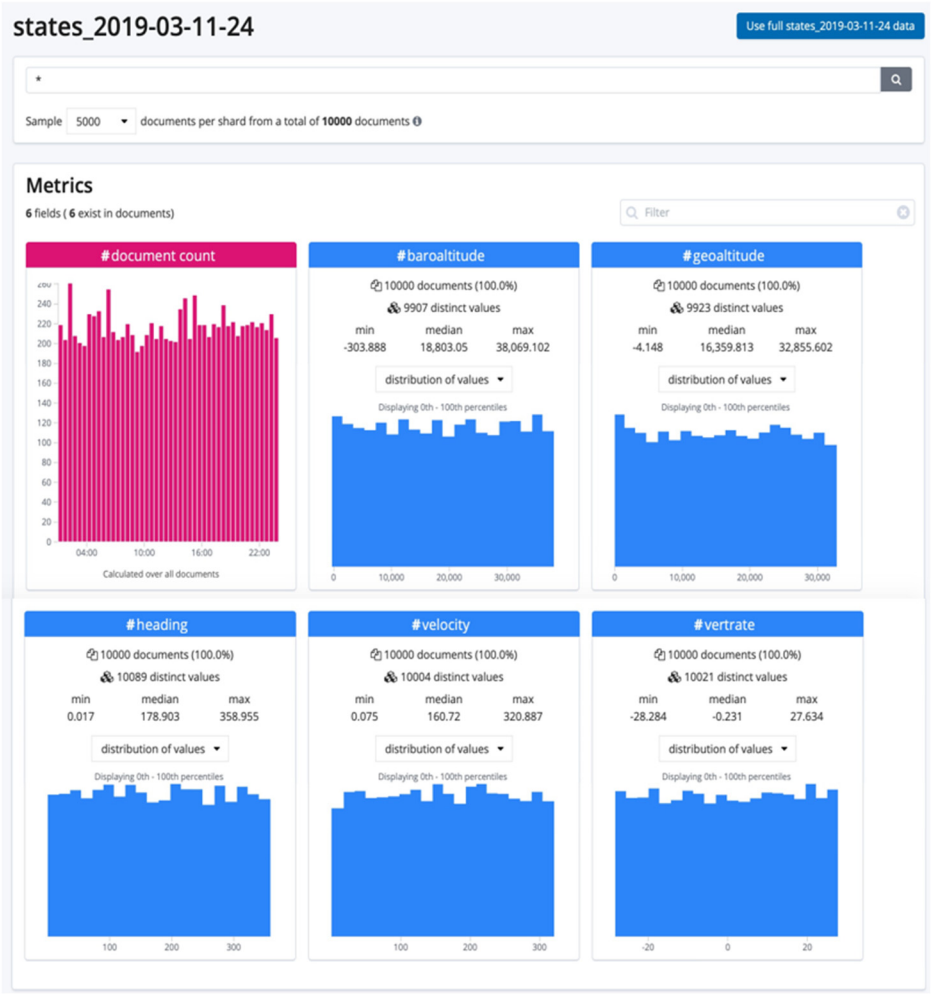


Figure 5. Elastic Kabana Visualizations of Statistics of Generated ADS-B Network Attacks.

5.2. Machine Learning: SVM

SVM was used as a kernel-based method, where feature vectors are implicitly mapped into a higher dimensional space where it is easier to find an optimal hyperplane for classifying observations. A linear kernel was used. Given training vectors $x_i \in \mathbb{R}^p$, $i = 1, \ldots, n$, in two classes and a vector $y \in \{1, -1\}^n$, our goal is to find $w \in \mathbb{R}^p$ and $b \in \mathbb{R}$ such that the prediction given by sign $\left(w^T \phi(x) + b\right)$ is correct for most samples. SVC solves the following primal problem [38–40]:

$$\min_{\omega, b, \zeta} \frac{1}{2} \omega^T \omega + C \sum_{i=1}^{n} \zeta_i$$

subject to

$$y_i \left(\omega^T \phi(x_i) + b\right) \geq 1 - \zeta_i, \quad (1)$$

$$\zeta_i \geq 0, \; i = 1, \ldots, n$$

Its dual is

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$

subject to

$$y^T \alpha = 0$$

$$0 \leq \alpha_i \leq C, \; i = 1, \ldots n$$

where e is the vector of all ones, $C > 0$ is the upper bound, Q is an n by n positive semi-definite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, where $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ is the kernel. Training vectors are implicitly mapped into a higher (maybe infinite) dimensional space by the function φ.

The decision function is

$$sgn(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x_j) + \rho) \tag{2}$$

A confusion matrix was used to present a summary of the results predicted for the classifications. There are four counts in a confusion matrix: true positives, false negatives, true negatives, and false positives. True positives are actual true and predicted true. False negatives are actual true and predicted false. True negatives are actual false and predicted false. False positives are actual false and predicted true.

For predicted ADS-B attacks, there were 3922 true positive messages, 963 false positive messages, 1088 false negatives, and 3149 true negatives, as shown in Table 5. The calculations for the classification report present the precision, recall, and F1 score of the machine learner. Precision indicates actual ADS-B attacks are predicted (Equation (3)).

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} = \frac{\text{3922 Messages}}{\text{3922 Messages} + \text{963 Messages}} = 0.802866 \tag{3}$$

Table 5. Confusion Matrix.

	Predicted Attack	**Predicted Message**
Actual Attack	3922 True Positive Messages	1088 False Negative Messages
Actual Message	963 False Positive Messages	3149 True Negatives

Recall indicates ADS-B attacks that are predicted as actual ADS-B attacks (Equation (4)).

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = \frac{\text{3922 Messages}}{\text{3922 Messages} + \text{1088 Messages}} = 0.782834 \tag{4}$$

The objective is to achieve high precision as well as high recall. In this case, the precision was 80.29%, and recall, which is also the attack detection rate, was 78.28%.

The F1-Score represents the harmonic mean of the precision and recall (Equation (5)).

$$\text{F1} - \text{Score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} = \frac{2 * 0.782834 * 0.802866}{0.782834 + 0.802866} = 0.792723469 \tag{5}$$

5.3. Volume

Data volume measures the scale of the data within the system [16]. ADS-B messages persist in 26 Elasticsearch indices. Indices 0 through 23 hold the ADS-B messages from OpenSky. Index 24 holds the generated ADS-B network attack messages. Index 25 holds the machine learner predictions. The volume mean is calculated using the ratio of Elasticsearch index storage size to the count of Elasticsearch documents containing OpenSky ADS-B messages (Equation (6)).

$$\frac{1}{24} \sum_{\text{h}=1}^{24} \frac{\text{StorageSize}}{\text{Documents}_{\text{Hour}}} = 247.7604 \text{ Bytes} \tag{6}$$

The yearly ADS-B volume is 41 TiB, based on the ADS-B specification and FAA statistics on the U.S. NAS (Equation (7)) [29].

$$\frac{\text{247.7604 Bytes}}{\text{1 Message}} * \frac{\text{6.2 Messages}}{\text{1 s}} * \frac{\text{18103000 h}}{\text{1 Year}} = \frac{\text{91 TiB}}{\text{1 Year}} \tag{7}$$

The yearly ADS-B volume is 91 TiB, based on the volume mean. This is calculated using the ratio of Elasticsearch index storage size to the count of Elasticsearch documents containing OpenSky ADS-B messages, ADS-B specification, and FAA statistics on the U.S. NAS (Equation (8)) [29].

$$\frac{112 \text{ Bytes}}{1 \text{ Message}} * \frac{6.2 \text{ Messages}}{1 \text{ s}} * \frac{18103000 \text{ h}}{1 \text{ Year}} = \frac{41 \text{ TiB}}{1 \text{ Year}} \quad (8)$$

5.4. Velocity

Data velocity is the analysis of the data as the data are generated [16].

$$\frac{6.2 \text{ Messages}}{1 \text{ s}} * \frac{18103000 \text{ h}}{1 \text{ Year}} = \frac{404058960 \text{ Messages}}{1 \text{ Year}} \quad (9)$$

Per ADS-B specifications and FAA statistics on the U.S. NAS, the yearly velocity is 404,058,960 messages and 12.81262557 messages per millisecond [29]. The velocity of ADS-B messages and average volume per message will determine the bit rate for the U.S. NAS to ingest ADS-B messages into Elasticsearch [29].

$$\frac{12.81262557 \text{ Messages}}{1 \text{ Millisecond}} * \frac{247.7604 \text{ bytes}}{1 \text{ Message}} = \frac{423.2614982 \text{ bits}}{1 \text{ s}} \quad (10)$$

The velocity was 12.81262557 and the average volume was 247.7604 bytes per message. The bit rate for the U.S. NAS is 423.2614982 bits per second (Equation (10)). In comparison, residentially available Gigabit internet is 1,000,000,000 bits per second, and commercially available Optical Carrier 768 (OC-768) operates at 39,813,000,000.12 bits per second. In this research, the server took 0.3565873159095 s to preprocess 20,000 messages. It took 0.178294 s for the server to preprocess 10,000 records.

$$\frac{10000 \text{ Messages}}{1 \text{ s}} * \frac{0.3565873159095 \text{ s}}{20000 \text{ Messages}} = \frac{56.08714 \text{ Messages}}{1 \text{ MilliSecond}} \quad (11)$$

The server preprocessed 56.08714 messages per millisecond (Equation (11)). The server took 11,802.38247703109 s to fit the model using 10,000 messages to create a machine learning model. The server fits the model in 0.000847 messages per millisecond. It took the server 0.3594533782452345 s to apply the model to 10,000 messages for the predictions. The server can apply the model with 27.82002 messages per millisecond. The U.S. NAS generates 404,058,960,000 messages per year or approximately 13 messages per millisecond [29].

$$\frac{0.178294 \text{ s}}{10000 \text{ Messages}} * \frac{0.3594533782452345 \text{ s}}{10000 \text{ Messages}} = \frac{18.69573609}{1 \text{ MilliSecond}} \quad (12)$$

Using a commercial off-the-shelf server, this system was able to predict 18.69573609 messages per millisecond (Equation (12)). Based on the FAA statistics, the velocity of detectable messages was 16 messages per millisecond. With the Elasticsearch overhead, the model was capable of processing 18.69573609 messages per millisecond. It takes 91 TB of data volume to store a year's worth of ADS-B messages in Elasticsearch.

6. Conclusions

In this work, flight patterns were characterized, including flight patterns for possible attacks. Flights outside the patterns are possible attacks, and ADS-B network attacks can be detected using network attack signatures. A precision and recall of close to 80% was achieved using SVM classification.

It took the server 0.36 s to preprocess the messages, 11,802.38 s to fit the model, and 0.36 s to apply the model for a prediction. While fitting took substantial time, the combination of preprocessing and applying the already fitted model took less than a second to finish,

that is, 0.72 s or 720 milliseconds for 20,000 messages, or approximately 27 ADS-B messages every millisecond. The U.S. National Air Space generates 404,058,960,000 messages per year, approximately 13 messages per millisecond. A commercial off-the-shelf server can process 16 messages per millisecond with the Elasticsearch overhead. This research, which can be applied to GAO identified problems and issues with FAA instantiation of ADS-B for public safety, used a commercial off-the-shelf server to keep up with U.S. NAS velocity of ADS-B messages. These findings will help in taking appropriate action on attacks detected in real time, hence improving flight safety.

7. Future Works

Combating attackers and mimicking those identified flight patterns using adversarial artificial intelligence would be the next step in this research. Some advanced threats, otherwise known as advanced persistent threats, use artificial intelligence to glean patterns from networks such as the FAA's ADS-B network. These threats carefully construct attacks that mimic those legitimate network traffic patterns. Future research is important because the current machine learner would potentially not detect such attacks as they mimic legitimate ADS-B network traffic. Exploration of other machine learners and artificial intelligence algorithms would add to the research. These possible machine learners include Random Forest classifier, Bayesian classifier, or Neural Networks. Additionally, artificial intelligence algorithms addressed through Neural Networks strive for 95 to 99% precision. Increasing precision allows for more accuracy in detecting attacks. Every year air travel increases; therefore, the increased amount of data needs to be processed with more velocity. To address this, consideration of other big data platforms besides Elasticsearch could increase the system's velocity capability. Other Platforms, such as Spunk, Spark, and even server-less architectures such as Lambda, offer opportunities for exploration.

Author Contributions: D.M.M. conceptualized the whole research and did most of the research work. D.M.M. also composed the initial draft of the paper. J.M. provided guidance in every step of the process. S.B. helped with the Data Mining/Machine Learning analysis and in the composition of the paper. W.B.G., J.S., R.B. and S.R. provided guidance and support throughout the research process. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author: Sikha Bagui, email: bagui@uwf.edu.

Acknowledgments: This work has been partially supported by the Askew Institute of the University of West Florida.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Federal Aviation Administration. Air Traffic by the Numbers. 2019. Available online: https://www.faa.gov/air_traffic/by_the_numbers/media/Air_Traffic_by_the_Numbers_2019.pdf (accessed on 21 January 2020).
2. Department of Transportation. Fact Sheet-FAA Forecast Fact Sheet-Fiscal Years 2016–2036. 2017. Available online: https://www.faa.gov/data_research/aviation/aerospace_forecasts/media/fy2016-36_faa_aerospace_forecast.pdf (accessed on 25 April 2019).
3. Federal Aviation Administration. Modernization of U.S. Airspace. 2018. Available online: https://www.faa.gov/nextgen/ (accessed on 3 June 2019).
4. United States Government Accountability Office. AIR TRAFFIC CONTROL: FAA Needs a More Comprehensive Approach to Address Cybersecurity as Agency Transitions to NextGen. 2015. Available online: https://www.gao.gov/products/GAO-15-370 (accessed on 25 April 2019).
5. United States Government Accountability Office. Next Generation Air Transportation System: Improved Risk Analysis could Strengthen Faa's Global Interoperability Efforts. 2015. Available online: https://www.gao.gov/products/GAO-15-608 (accessed on 25 April 2019).

6. Mink, D.; Yasinsac, A.; Choo, K.R.; Glisson, W. Next Generation Aircraft Architecture and Digital Forensic. In Proceedings of the Americas Conference on Information Systems, San Diego, CA, USA, 11–14 August 2016.
7. Mink, D.M. Indicators of Compromise for the United States Federal Aviation Administration Next Generation Air Transportation System Automatic Dependent Surveillance-Broadcast. Ph.D. Thesis, University of South Alabama, Mobile, AL, USA, 2019.
8. Baek, J.; Byon, Y.; Hableel, E.; Al-Qutayri, M. Making air traffic surveillance more reliable: A new authentication framework for automatic dependent surveillance-broadcast (ADS-B) based on online/offline identity-based signature. *Secur. Commun. Netw.* **2015**, *8*, 740–750. [CrossRef]
9. Strohmeier, M.; Schafer, M.; Lenders, V.; Martinovic, I. Realities and challenges of nextgen air traffic management: The case of ADS-B. *IEEE Commun. Mag.* **2014**, *52*, 111–118. [CrossRef]
10. Strohmeier, M.; Lenders, V.; Martinovic, I. On the security of the automatic dependent surveillance-broadcast protocol. *IEEE Commun. Surv. Tutor.* **2014**, *17*, 1066–1087. [CrossRef]
11. Schäfer, M.; Lenders, V.; Martinovic, I. Experimental analysis of attacks on next generation air traffic communication. In *International Conference on Applied Cryptography and Network Security*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 253–271.
12. Chen, T.C. An authenticated encryption scheme for automatic dependent surveillance-broadcast. *IEEE Commun. Mag.* **2012**, 127–131. [CrossRef]
13. Costin, A.; Francillon, A. Ghost in the Air (Traffic): On Insecurity of ADS-B Protocol and Practical Attacks on ADS-B Devices. *Black Hat. USA* **2012**, 1–12. Available online: http://www.s3.eurecom.fr/docs/bh12us_costin.pdf (accessed on 25 April 2019).
14. Sampigethaya, K.; Poovendran, R. Security and privacy of future aircraft wireless communications with offboard systems. In Proceedings of the 2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011), Bangalore, India, 4–8 January 2011; pp. 1–6.
15. McCallie, D.; Butts, J.; Mills, R. Security analysis of the ADS-B implementation in the next generation air transportation system. *Int. J. Crit. Infrastruct. Prot.* **2011**, *4*, 78–87. [CrossRef]
16. Katal, A.; Wazid, M.; Goudar, R.H. Big data: Issues, challenges, tools and good practices. In Proceedings of the 2013 Sixth International Conference on Contemporary Computing (IC3), Noida, India, 8–10 August 2013; pp. 404–409.
17. Jiawei, H.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann Publishers: Waltham, MA, USA, 2011.
18. Bagui, S.; Mink, D.; Cash, P. Data mining techniques to study voting patterns in the US. *Data Sci. J.* **2007**, *6*, 46–63. [CrossRef]
19. Guller, M. *Big Data Analysis with Spark*; Apress: New York, NY, USA, 2015
20. Strohmeier, M.; Martinovic, I. On passive data link layer fingerprinting of aircraft transponders. In Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or Privacy, Denver, CO, USA, 12–16 October 2015; pp. 1–9.
21. Lauf, A.P.; Peters, R.A.; Robinson, W.H. A distributed intrusion detection system for resource-constrained devices in ad-hoc networks. *Ad. Hoc. Netw.* **2010**, *8*, 253–266. [CrossRef]
22. Mitchell, R.; Chen, I. A survey of intrusion detection techniques for cyber-physical systems. *ACM Comput. Surv. (CSUR)* **2014**, *46*, 1–29. [CrossRef]
23. Wesson, K.D.; Humphreys, T.E.; Evans, B.L. Can cryptography secure next generation air traffic surveillance? *IEEE Secur. Priv. Mag.* **2014**. Available online: https://www.semanticscholar.org/paper/Can-Cryptography-Secure-Next-Generation-Air-Traffic-Wesson-Humphreys/94be9dccbb8708a2ca1444ae8b24afb128026762 (accessed on 25 April 2019).
24. Kacem, T.; Wijesekera, D.; Costa, P. Integrity and authenticity of ADS-B broadcasts. In Proceedings of the 2015 IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2015; pp. 1–8.
25. Gauthier, R.; Seker, R. Addressing Operator Privacy in Automatic Dependent Surveillance-Broadcast (ADS-B). In Proceedings of the 51st Hawaii International Conference on System Sciences, Waikoloa, HI, USA, 3–6 January 2018.
26. Finke, C.; Butts, J.; Mills, R. ADS-B encryption: Confidentiality in the friendly skies. In Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop, Oak Ridge, TN, USA, 8–10 January 2013; pp. 1–4.
27. Varga, M.; Polgár, Z.A.; Hedeşiu, H. ADS-B based real-time air traffic monitoring system. In Proceedings of the 2015 38th International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, 9–11 July 2015; pp. 215–219.
28. He, D.; Kumar, N.; Choo, K.R.; Wu, W. Efficient hierarchical identity-based signature with batch verification for automatic dependent surveillance-broadcast system. *IEEE Trans. Inf. Forensics Secur.* **2016**, *12*, 454–464. [CrossRef]
29. Mink, D.; Glisson, W.B.; Benton, R.; Choo, K.R. Manipulating the Five V's in the Next Generation Air Transportation System. In *International Conference on Security and Privacy in Communication Systems*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 271–282.
30. Boci, E.; Thistlethwaite, S. A novel big data architecture in support of ADS-B data analytic. In Proceedings of the 2015 Integrated Communication, Navigation and Surveillance Conference (ICNS), Herdon, VA, USA, 21–23 April 2015.
31. Tassone, C.F.R.; Martini, B.; Choo, K.R. Visualizing digital forensic datasets: A proof of concept. *J. Forensic Sci.* **2017**, *62*, 1197–1204. [CrossRef] [PubMed]
32. Ying, X.; Mazer, J.; Bernieri, G.; Conti, M.; Bushnell, L.; Poovendran, R. Detecting ADS-B Spoofing Attacks using Deep Neural Networks. In Proceedings of the 2019 IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, 10–12 June 2019; pp. 187–195.
33. Elastic. Introducing Machine Learning for the Elastic Stack. 2017. Available online: https://www.elastic.co/blog/introducing-machine-learning-for-the-elastic-stack (accessed on 25 April 2019).
34. The Apache Software Foundation. Apache NiFi. 2018. Available online: https://nifi.apache.org/ (accessed on 25 April 2019).

35. The OpenSky Network. OpenSky REST API. 2018. Available online: https://opensky-network.org/apidoc/rest.html (accessed on 15 April 2019).
36. The OpenSky Network. Index of /Datasets. 2018. Available online: https://opensky-network.org/datasets/ (accessed on 25 February 2019).
37. Elastic. A Picture's Worth a Thousand Log Lines. 2018. Available online: https://www.elastic.co/products/kibana (accessed on 25 February 2019).
38. Guyon, I.; Boser, B.; Vapnik, V. Automatic capacity tuning of very large VC-dimension classifiers. In *Advances in Neural Information Processing Systems*; Hanson, S.J., Cowan, J.D., Giles, C.L., Eds.; Morgan Kaufmann: San Mateo, CA, USA, 1993; Volume 5, pp. 147–155.
39. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
40. Scikit-Learn Developers. sklearn.svm.SVC. 2020. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC (accessed on 19 January 2020).