

# ADS-B Reception Error Correction Based on the LSTM Neural-Network Model

# Jamal Habibi Markani, Syed Ibtehaj Raza Rizvi, Abdessamad Amrhar, Jean-Marc Gagné, René Jr. Landry

Lassena Laboratory, Department of Electrical Engineering, École de Technologie Supérieure (ÉTS), Montreal, Canada Email: jamal.habibi@lassena.etsmtl.ca

How to cite this paper: Markani, J.H., Rizvi, S.I.R., Amrhar, A., Gagné, J.-M. and Landry, R.J. (2023) ADS-B Reception Error Correction Based on the LSTM Neural-Network Model. *Communications and Network*, **15**, 25-42.

https://doi.org/10.4236/cn.2023.152003

**Received:** March 9, 2023 **Accepted:** May 6, 2023 **Published:** May 9, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

http://creativecommons.org/licenses/by/4.0/

CC O Open Access

## Abstract

Standard automatic dependent surveillance broadcast (ADS-B) reception algorithms offer considerable performance at high signal-to-noise ratios (SNRs). However, the performance of ADS-B algorithms in applications can be problematic at low SNRs and in high interference situations, as detecting and decoding techniques may not perform correctly in such circumstances. In addition, conventional error correction algorithms have limitations in their ability to correct errors in ADS-B messages, as the bit and confidence values may be declared inaccurately in the event of low SNRs and high interference. The principal goal of this paper is to deploy a Long Short-Term Memory (LSTM) recurrent neural network model for error correction in conjunction with a conventional algorithm. The data of various flights are collected and cleaned in an initial stage. The clean data is divided randomly into training and test sets. Next, the LSTM model is trained based on the training dataset, and then the model is evaluated based on the test dataset. The proposed model not only improves the ADS-B In packet error correction rate (PECR), but it also enhances the ADS-B In terms of sensitivity. The performance evaluation results reveal that the proposed scheme is achievable and efficient for the avionics industry. It is worth noting that the proposed algorithm is not dependent on conventional algorithms' prerequisites.

# **Keywords**

ADS-B, Long Short-Term Memory, Packet Error Correction Rate, Error Correction, Bit Error Rate

# **1. Introduction**

The use of ADS-B, a new surveillance technology, is increasing globally. An air-

craft broadcasts its own aviation parameters spontaneously and on a regular basis to air traffic ground stations and other nearby aircraft in order to improve air traffic monitoring and management, in contrast to the secondary surveillance radar that operates on an interrogation-reply principle [1] [2]. Numerous advantages have been brought about by the introduction of ADS-B, including better situational awareness, increased traffic capacity and efficiency, and decreased fuel consumption [3] [4]. As a result, many nations, including those in North America, Europe, and Australia, are encouraging the modernization of air traffic management technology and attempting to swap traditional radar systems with ADS-B technology [5] [6] [7].

ADS-B receiver sensitivity level and robustness to interference are key factors in ADS-B performance. On the one hand, the ADS-B robustness has been addressed by the multi-sample method described in [8] [9]. In the event of high SNRs, this approach performs well [9]. On the other hand, satellite-based ADS-B technology, which is required to detect ADS-B signals with low SNR, has recently gained much attention. This multi-sample technique's performance deteriorates dramatically due to the high bit error rate (BER) at low SNRs. While some research has been conducted to enhance the BER and error correction, such error correction approaches are limited by the number of low-confidence bits. More specifically, all the incorrect bits must be classified as low-confidence bits to have successful error correction. In addition, the number of low confidence bits is restricted to threshold 5 for the brute force approach, and this number is limited to 12 for the conservative approach [10].

This research covers the design and implementation of an LSTM error correction model based on actual flight test data. This work involves the extraction of ADS-B data features, the design of a new error correction architecture based on the LSTM model, and presents the results achieved using the proposed scheme. The motivations for writing this article are two-fold:

- Most of the error correction solutions are still highly dependent on the number of low-confidence bits and their distribution;
- Considering ADS-B message correlation and serial correlation features, machine learning algorithms, particularly the LSTM model, can help to enhance error correction performance. Therefore, ADS-B sensitivity in low SNR situations and high interference events will be improved.

This study illustrates the advantages of the LSTM-based error correction model and its implementation. In addition, algorithm evaluation was conducted under real flight data to have a comprehensive observation of the algorithm's and the proposed model's performance.

#### 2. Literature Review

The sliding window error correction technique operates by assessing consecutive 24-bit windows. The selected window starts from bits 89 - 112 of the ADS-B messages. If any of the bits within the window is declared as low confidence,

then its complementary value applies. Otherwise, the window is shifted downward till it reaches the start of the ADS-B messages. In addition, if the number of low-confidence bits within the window is greater than 12, then error correction is not attempted. This approach is useful in low-level interference events; however, its performance is drastically degraded in high False Replies Unsynchronized In Time (FRUIT) environments [10].

A more straightforward strategy called the conservative technique is applied to severe FRUIT circumstances. The prerequisites for the success of error correction by this approach are that there are no more than 12 low-confidence bits, and all the low-confidence bits are contained inside a 24-bit window. Although the computation load is minimal, the error correction rate is low [10].

Another error correction technique is known as the Brute Force approach. As implied from its name, every possible combination of a low confidence bit's error syndrome is computed, and based on the results, the relative correction is applied. This method correction is successful provided that the declaration rule is applied correctly to every bit, however, it has a high computational load [8] [10] [11].

As described in [12], a matching filter algorithm is used for decoding ADS-B signals. In addition, a new error correction algorithm is proposed based on the combination of the Brute Force approach and an N-confidence level. The threshold of confidence declaration is determined by the received signal energy per bit to noise power spectral density ratio  $(E_b/N_0)$  rather than the fixed value. According to the simulation result, BER is improved by applying both match filter detection and the offered error correction algorithms. The proposed error correction solution is based on the confidence bit declaration of the matching filter.

The authors of reference [13] exploited different matching filter schemes to detect ADS-B data. The proposed approach uses leading and trailing chip energy to estimate bit and confidence levels. Therefore, the threshold of confidence level is dynamic. The conservative approach is used for error correction.

In telecommunication systems, both transmitters and receivers that are Artificial Intelligence-based have been studied. AI models pave the way for individuals to optimize a whole transceiver as an autoencoder without prior knowledge of channel coding and modulation [14]. A recurrent neural network (RNN) was also applied to optimize the encoding and decoding together for Additive white Gaussian noise (AWGN) channels with feedback [15]. Additionally, it was found that even when the code structure is not explicitly taught, the NN-based decoder for structured codes can generalize to some untrained codewords. The complexity of an NN-based system is inversely correlated with the number of codewords to be classified. Therefore, the strict latency constrains the NN-based decoder, as the code space increases exponentially with code length. Since the code length of ADS-B data is limited to 112 bits, which is much shorter than the wireless telecommunication data package length, it is feasible to use the NN-based error correction model for real-time purposes. Recently, a hybrid Convolutional Neural Network (CNN) with a Gated Recurrent Unit (GRU) model was proposed to predict flight trajectory based on ADS-B data [16]. According to the experimental results, the prediction error rate is improved considerably. Based on reference [17], the prediction of airplane positioning information, including latitude, longitude, and altitude, by various ADS-B messages is performed based on the deep learning model. The results show that all the positioning element prediction error rates are improved significantly compared to conventional approaches. However, these approaches are not examined for the correction of ADS-B data errors.

#### 3. Methodology and Solutions

This section describes the data preparation process and ADS-B data features and explains the LSTM-based proposed model error correction and its implementation.

Figure 1 illustrates an ADS-B receiver in terms of functional blocks. It consists of a 1090 MHz RF front end, preamble detection, bit detection, parity check, and bit error correction functional blocks [18]. The output is an ADS-B message, which must be sent to a decoder to be transformed into real data. As mentioned earlier, the main goal is to improve the bit error correction block performance by applying the LSTM scheme.

The proposed solution is classified into the following three major categories:

- Data preparation;
- LSTM cell and LSTM network architecture;
- Implementation of LSTM algorithm for ADS-B error correction.

#### 3.1. Data Preparation

The database used here consists of over 400 flight data items for various types of airplanes from flights performed in different locations, including North America and Europe. The database contains over 2000 hours of flight data. **Figure 2** illustrates a sample flight in the USA.

In the initial stage, data cleaning is done to remove blank data as well as outliers. The flight data were collected from the GPS of each airplane. The flight data include positioning and velocity information. ADS-B data is generated from the flight data according to ADS-B airborne data format and transmission rate. ADS-B data is generated almost every 250 milliseconds. The ADS-B positioning message (Latitude, Longitude, and Altitude), the velocity message (speed, heading, and vertical rate), and the aircraft identity message are then generated. It is worth noting that the messages are generated based on the ADS-B airborne message transmission rate. Therefore, the positioning and velocity messages'



Figure 1. ADS-B Receiver functional blocks.



Figure 2. Flight route sample from the dataset.

generation rate is two messages per second, and the aircraft identity message generation rate is one message every 5 seconds [10]. In addition, the ICAO of each flight is generated randomly to maintain diversity. The ICAO data is created according to a 24-bit address format [10]. Other control information and CRC parity bits are generated according to their standard formats. After the generation of ADS-B messages, bit errors are applied to messages according to the following scenarios:

1) Bit errors 5: errors made randomly in five bits among the 112 bits of ADS-B messages;

2) Bit errors 10: errors made randomly in ten bits among the 112 bits of ADS-B messages;

3) Bit errors 12: errors made randomly in twelve bits among the 112 bits of ADS-B messages; and

4) Bit errors 15: errors made randomly in fifteen bits among the 112 bits of ADS-B messages.

The generated dataset is split randomly into two parts, training and test datasets, via the Scikit-learn library function [19]. The training dataset is used to train the proposed LSTM model and the test data set is deployed to evaluate the trained model performance.

#### 3.2. LSTM Cell and LSTM Network Architecture

An LSTM cell is the basic building block of an LSTM network. It consists of several components, including:

- An input gate: Determines what proportion of the input to allow into the cell;
- A forget gate: Defines how much of the previous cell state to forget;
- An output gate: Decides how much of the cell state to output;
- A cell state: Where the "memory" of the cell stores information over time;
- A hidden state: Where the output of the cell is passed to the next cell in the sequence.

Overall, an LSTM cell is designed to selectively store and retrieve information from the input and previous states, allowing it to capture long-term dependencies in the sequence data. An LSTM cell is considered a major element of an LSTM network.

As illustrated in **Figure 3**, a typical multilayer LSTM network architecture for serial data modeling is composed of numerous LSTM and fully connected (FC) layers. The input layer, two LSTM layers, two FC layers, and the output layer are all components of this LSTM architecture. Here, q1 and q2 represent the count of nodes for the two FC layers, whereas p1 and p2 represent the number of nodes (units) for the LSTM1 and LSTM2 layers, respectively.

To create the necessary amount of output features, the LSTM layers are linked to the target output layer using the FC layers. As their name would imply, FC layers have complete connections to all of the activation nodes in the previous layer. This structure allows the network to capture complex patterns in the data.

In general, LSTM models are a powerful tool for sequence prediction problems, as they can capture long-term dependencies in the data and handle variable-length sequences. However, training LSTMs can be computationally expensive and may require considerable computational resources, especially for large datasets and deep networks. Additionally, it can be challenging to tune the hyperparameters of an LSTM network to achieve good performance, and overfitting can be a common problem.

#### 3.3. LSTM Model Implementation for ADS-B Error Correction

This section describes the design and implementation of the LSTM model for ADS-B In error correction and offers a solution for how to utilize the model in ADS-B In technology. An ADS-B reception error correction system based on an LSTM model could involve using LSTM neural networks to analyze and correct errors in the data received from ADS-B transmitters. The LSTM model can be trained on a dataset of known, correct ADS-B transmissions and corresponding receiver errors, and then used to predict and correct errors in real time as new



Figure 3. A typical multilayer LSTM architecture.

ADS-B data is received. This approach would improve the accuracy and reliability of the ADS-B system.

There are three types of messages in ADS-B airborne. The characteristics and features of each message as below:

• Positioning message: the message consists of control bits, latitude, longitude, and altitude information. It broadcast every 0.5 second. With taking into consideration that the position information changes in real-time smoothly, these series of messages are highly correlated with each other. Therefore, positioning messages logs could provide prediction for possible next messages which can be useful for error correction.

- Velocity message: The message includes the aircraft's ground speed, Heading and vertical rate, which provides information about the aircraft's direction and speed of travel. It broadcasts every 0.5 second. In real operational environment, the velocity information changes smoothly. They are considered as time series data which is highly correlated with each other. Thus, with having sequential data of velocity, it is possible to predict next data. As a result, these features can leverage the prediction of possible errors.
- Aircraft identification message: The message includes a unique identifier for the aircraft, which allows for accurate tracking and identification of the aircraft. The same message is emitted by ADS-B every 5 second.

Overall, ADS-B messages provide series of data which broadcast according to transmission rate. Time series features and high correlation among the messages make ADS-B message to be predictable by LSTM model.

**Figure 4** illustrates ADS-B's airborne major messages structure and related features. As indicated, MSG1 gives the positioning information, which consists of the downlink format and capability field (8 bits), the International Civil Aviation Organization (ICAO) address (24 bits), the payload (56 bits), which contains altitude, latitude, and longitude data, and a 24-bit cyclic redundancy check (CRC). The periodicity of the "even" and "odd" compact position reporting (CPR) messages is 0.5 seconds. MSG1's format and ID fields are the same for all messages, while the payload and CRC fields change according to the positioning information and maintain a considerable correlation with previous messages. MSG2 offers velocity information, which contains the downlink format and capability field (8 bits), the ICAO address (24 bits), the payload (56 bits), which includes the speed, heading and vertical rate, and a 24-bit CRC. MSG2 periodicity is two messages per second.

MSG2's first 32 bits of data are repeated for every message, whereas the payload and CRC bits are gradually modified based on the velocity data. Therefore, every message is highly correlated with previous messages. MSG3 indicates aircraft identification information; it contains the downlink format and capability field (8 bits), the ICAO address (24 bits), the payload (56 bits) which includes the aircraft category and identification (8 characters), and a 24-bit CRC. The





MSG3 message data is always consistent and is repeated every 5 seconds.

ADS-B messages are considered as time series data. Given that the complexity of a time series makes time series prediction challenging, it is not possible to formulate ADS-B data prediction. Considering all the above-mentioned features and specifications of ADS-B messages, neural network models, specifically LSTM models, can help an ADS-B receiver system to correct a possible error.

**Figure 5** illustrates the LSTM-based ADS-B message error correction structure and signal processing procedure.

In the initial stage, ADS-B reception detects the ADS-B bust of a signal without applying any error detection and correction. ADS-B's bust of the messages are categorized based on ICAO matching and correlation, and then placed in buffers whose size should be a multiplication of 5 seconds duration messages. Therefore, the buffer size should be a multiplication of 21 messages that consist of 10 successive positioning messages, 10 sequential velocity messages, and one aircraft identification message. It is worth noting that the longer the buffer size (or window size), the better the performance of the LSTM model. However, such a longer size leads to an increase in the processing time as well as CPU load. Upon completion of each buffer, the newest message's CRC error will be evaluated. If there is no CRC error, then the oldest message will be sent to the ADS-B decoding block for further decoding and displaying. If an error is detected, the whole buffer data including the error data are injected into the LSTM-trained model. The function of the LSTM-trained model is to estimate and correct possible errors. The output of the LSTM model is evaluated in terms



Figure 5. LSTM-based ADS-B message error correction architecture.

of the CRC error check. If an error is corrected, then the bust of the message will be sent to ADS-B decoding block for further processing. It is also expected that the trained model will reduce the number of error bits in comparison to the input data. If there is an error in the message after the LSTM output, the brute force or conservative error correction algorithms can be applied make the message free of errors.

The proposed LSTM model is not dependent on the confidence level factor. However, legacy algorithms such as the conservative and brute force approaches are dependent on the confidence declaration of each message's bits during the bit detection phase.

#### 4. Experimental Results and Discussion

After outliers and inconsistences were removed, the dataset contained over 2000 hours of clean flight data. LSTM networks are mainly used for the prediction of data series, which follow patterns. In ADS-B messages, ADS-B positioning messages are a series of aircraft coordinate information that indicate an aircraft's trajectory. Similarly, ADS-B velocity messages contain sequences of data that reveal horizontal and vertical speeds and heading information. Therefore, serial correlation is extracted from a complete dataset to measure the correlation between consecutive messages. **Table 1** gives the serial correlation mean values for three major airborne messages for the whole database, which show that the messages are highly correlated with each other.

**Figure 6** illustrates the flow of actions for the implementation of ADS-B error correction based on the LSTM model. The following five steps (in blocks on the left side of **Figure 6**) must be executed in sequence:

1) Target data preparation: The input data has to be transformed into ADS-B Out format according to each type of message transmission rate. These data are considered as the LSTM model target values, as they are free of error. Since the LSTM model works with time-series data, the sequences of these data need to be maintained.

2) Input data preparation: Various types of error must be applied randomly for the whole dataset that is generated in step 1. Therefore, the input data of the LSTM model contains various types of errors.

3) Training and test datasets' classification: Target and input data is divided randomly into training and test sets in a ratio of 70 and 30 percent, respectively.

4) Train the model: The proposed model is trained according to the training dataset's input and target data. The Adam optimizer and the root mean squared error (RMSE) loss function are deployed during the training phase. Using the

Table 1. Serial Correlation of airborne messas	ges.
--	------

Serial Correlation				
Message Type	Positioning message	Velocity message	Aircraft Identification message	
mean value	0.666	0.845	1	



Figure 6. ADS-B error correction with LSTM implementation flow graph.

training data, the model is trained by calling the fit method and providing the training data and target values. During training, the model adjusts its weights to minimize the loss function.

5) Performance evaluations: After training, the model's performance must be assessed on both training and test data. This is done by calling the evaluate method and providing the input data and the target values. The outcome provides an estimate of the model's generalization performance, which determines whether a model is overfitted or underfitted.

The loss value produced by the LSTM model during training represents the difference between the predicted outputs and the true outputs. The smaller the loss, the closer the predictions are to the true values. The trained dataset is split into 90 and 10 percent for training and validation sets to avoid the overfitting phenomena. **Figure 7** reveals the losses of training and validation data sets for 18 epochs. As the trends demonstrate, both the losses' values converge as more epochs are run. Additionally, the deployment of an early stopping function prevents the trained model from the overfitting phenomenon.

In addition to the loss metric, the packet error correction rate (PECR) is also used to measure the model performance. A measure of the ability of the trained LSTM model to correct errors in packets of data, it represents the proportion of packets that have been corrected by the LSTM model compared to the total number of packets that have errors. **Figure 8** shows the model performance in training and test datasets in various scenarios. To be more specific, **Figure 8(a)** indicates the PECR for packets that have 5 bits of error. The PECR values at axis



Figure 7. Trend of trained model loss and validation loss.



Figure 8. Packet error correction rate of training and test datasets for various test cases.

"0" reveal that about 94 and 85 percent of the corrupted packets in the training and test datasets are corrected without any bit error, respectively. Similarly, the PECR value at axis "1" demonstrates that about 4 and 10 percent of the training and test datasets bit errors, respectively, are reduced from 5-bit errors to 1-bit errors. **Figures 8(b)-(d)** demonstrate the model performance for 10-, 12-, and 15-bit error messages respectively. **Figure 8(d)**, the worst case, with a high number of bit errors and testing sets, shows that the trained model is able to correct over 60 percent of the packets with errors. In addition, over 92 percent of the test dataset packets fall below 5 bits of error, which makes it possible to correct them with the brute force algorithm. Overall, the trained model performance is quite promising. It can improve ADS-B In performance and sensitivity in high interference situations as well as low SNR conditions.

Moreover, a detailed investigation was performed on bit error distributions and their correction. Figure 9 shows the error counts for each bit over a 112-bit span before and after correction. It reveals that errors are distributed randomly over the 112-bit span. As explained earlier in the section on serial correlation and ADS-B data specification, the error correction performs better on bits of over 1 to 60 and 72 to 81 segments which contain control and message type data as well as ICAO information, but the error correction is not as good as the first bits in the payload field. This is because the payload field, including the CRC data, is varying in relation to the real data. However, the bit error correction rate is still over 73% for the worst cases, which is quite promising.

Due to the nature of radio signal transmission, errors often occur in the data transmitted via ADS-B. These errors can affect the accuracy of the aircraft's position and other information, such as altitude and velocity, and are caused by a variety of factors, such as multipath, interference, and atmospheric conditions. The CRC technique adds a check value to the transmitted data that can be used to detect errors at the receiver. The primary goal of this research is to propose a novel mechanism to improve error correction and enhance ADS-B In sensitivity.



ADS-B messages are considered time series data due to their statistical features like serial correlation. Time series prediction is a challenging problem in

Figure 9. Error distribution and error correction rate before and after correction over the 112-bit span.

the field of machine learning due to its temporal dependencies and non-stationary nature. LSTM networks have been shown to be effective in addressing some of these challenges. LSTMs are able to capture and retain important information from the past and use it to make predictions or error corrections, making them well-suited for time series prediction. Additionally, LSTMs can handle multi-step forecasting by making predictions one time step at a time, and they can also handle multivariate time series data by using multiple parallel memory cells. However, careful feature extraction and parameter tuning are still necessary to achieve good performance on time series prediction tasks with LSTMs.

In the context of ADS-B error correction, the Brute force and Conservative approaches refer to two different strategies for correcting errors in the received data. In the Brute force approach, the receiver uses all available information to correct errors in the received data, regardless of the amount of processing required. This approach is more computationally intensive but can provide higher error correction performance by exploiting all the available information. In the Conservative approach, the receiver only uses a subset of the available information to correct errors in the received data, thereby reducing the processing requirements and the computational complexity. While this approach may result in lower error correction performance, nonetheless, it is typically more suitable for real-time applications where computational resources are limited. The choice between a brute force and a conservative approach depends on the specific requirements of the ADS-B system, such as the error rate, the transmission rate, and the available computational resources. In some cases, a hybrid approach that uses elements of both approaches may be used to achieve a balance between error correction performance and computational complexity. In addition, both approaches are dependent upon confidence bit declaration, the number of error bits, and the error distribution window. Thus, if these conditions are not met, the error correction will fail and the message will be discarded.

In contrast, the proposed LSTM-based error correction solution is not restricted to the number of error bits or an error distribution window, nor does it depend upon a confidence bit declaration. After the training phase, the model can be utilized in real-time applications, as it can process and correct thousands of packet errors in less than one second which is quite sufficient for real-time usage. In addition, computational load and the complexity are far less than those of the brute force approach. Thus, implementation of the LSTM-based error correction model not only improves error correction performance, it also reduces computational load and processing time.

Root Mean Squared Error (RMSE) criteria are used for the measurement of loss during the training phase. The smaller the loss, the better the model is trained. However, it is important to note that a low loss value does not necessarily mean that the model has good performance. The model may be overfitting to the training data, meaning that it has memorized the training examples but cannot generalize well to new examples. In this case, the model would perform well on the training data but poorly on the test data. In addition, to determine whether a model is overfitted to the data, the model's performance is evaluated on a separate validation set, which is a portion of the data that the model has not seen during training. The loss value on the validation set can provide an estimate of the model's generalization performance. Furthermore, early stopping is a technique to prevent overfitting in machine learning. In the case of LSTMs, early stopping is implemented by monitoring the performance of the model on a validation set during training. If the performance on the validation set stops improving or starts to degrade, this is taken as an indication that the model has started to overfit and so the training should be stopped [20].

ADS-B In sensitivity refers to the ability of a receiver to detect ADS-B out signals with high accuracy [10]. The sensitivity of an ADS-B In receiver is directly impacted by low SNR, particularly at long ranges or in challenging environments, such as when there are high levels of interference or overlapping. Error correction is an important aspect of ADS-B In systems, as it helps to ensure the accuracy of the data received from aircraft. Overall, the performance and accuracy of ADS-B receivers are dependent on both the sensitivity of the receiver and the effectiveness of the error correction techniques used. Taking advantage of the proposed LSTM model can help to achieve sensitivity requirements and improve effectiveness.

## **5.** Conclusions

Error correction plays a pivotal role in ADS-B receiver performance and is key to meeting sensitivity requirements. Although a few algorithms, including the brute force and conservative approaches, have been proposed for performing error correction, they have the limitations of requiring a bit error distribution window, a limited number of error bits, and a confidence level declaration. The novel proposed error correction model, the LSTM model, performs without any restrictions in real-time cases and with less computational load. In contrast to legacy solutions, LSTM-based model accuracy for packets with over 12 bits of error is over 60%, while these sorts of packages are simply discarded in legacy algorithms. Additionally, the LSTM-based error correction model is quite compatible with SDR-based ADS-B receivers and it can enhance the performance of ADS-B In, enabling it to detect more sensitive data [9].

The proposed model is designed based on ADS-B's various messages' features and their transmission periods. In addition, data cleaning and the removing of outliers are done during the data preparation period. The model is trained based on the training dataset and cross-validation and early stopping techniques are deployed to avoid the overfitting phenomenon. The model has been evaluated for various test datasets with different numbers of error bits, and the results are quite promising.

It is essential to have a sufficient amount of training data to adequately learn the patterns in data. Although the dataset contains over 2000 hours of flight data, providing more data and flight information from every part of the world and for different types of aircraft will add more patterns to the model for training.

Whole datasets including Training and test sets are limited to 2000 hours flight data in north America. Exposing the system to new data from other area of the world may lead to degradation of the system performance. Feeding the system with variety of data can cause to have more robust trained system.

Bi-LSTM (Bidirectional Long Short-Term Memory) and LSTM are both types of recurrent neural networks that are commonly used for modeling sequential data. While they share many similarities, there are also some key differences between Bi-LSTMs and LSTMs. It's important to note that the effectiveness of the Bi-LSTM for error correction depends on the quality and size of the training data, as well as the specific error correction task being performed. The main difference between the two is that Bi-LSTM processes the input sequence in both forward and backward directions, whereas LSTM processes the input sequence only in the forward direction. This means that Bi-LSTM can capture both past and future dependencies in the input sequence, whereas LSTM can only capture past dependencies.

Applying Bi-LSTM model offers error correction based on past and future data. It is expected to improve the model performance; however the Bi-LSTM model implementation is more complex than LSTM model.

The proposed model offers a solution for high interference conditions and low SNR situations, such as high-range aircraft signal detection, as well as a space-based ADS-B signal identification which requires more sensitivity.

# Acknowledgements

The work in this paper was done under the NextGen Project of the LASSENA Lab, École de Technologie Supérieure (ÉTS). NSERC, CRIAQ with Thales, ACSS/L3Harris, SII Canada and Bombardier all provided funding for this study.

# **Conflicts of Interest**

The authors declare no conflicts of interest regarding the publication of this paper.

#### References

- Strohmeier, M., Martinovic, I. and Lenders, V. (2018) A k-NN-Based Localization Approach for Crowdsourced Air Traffic Communication Networks. *IEEE Transactions on Aerospace and Electronic Systems*, 54, 1519-1529. https://doi.org/10.1109/TAES.2018.2797760
- [2] Mangali, N.K. and Bagmare, V.S. (2017) Development of a Power over Ethernet (PoE) Enabled ADS-B Receiver System. 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 22-24 March 2017, 1888-1891. <u>https://doi.org/10.1109/WiSPNET.2017.8300089</u>
- [3] Hableel, E., Baek, J., Byon, Y.-J. and Wong, D.S. (2015) How to Protect ADS-B: Confidentiality Framework for Future Air Traffic Communication. 2015 *IEEE Confe*-

*rence on Computer Communications Workshops (INFOCOM WKSHPS*), Hong Kong, 26 April-1 May 2015, 155-160. https://doi.org/10.1109/INFCOMW.2015.7179377

- [4] Barsheshat, A.A. (2011) Implementation of ADS-B Systems—Benefits and Considerations. 2011 Tyrrhenian International Workshop on Digital Communications— Enhanced Surveillance of Aircraft and Vehicles, Capri, 12-14 September 2011, 197-201.
- [5] Schafer, M., Strohmeier, M., Lenders, V., Martinovic, I. and Wilhelm, M. (2014) Demonstration Abstract: OpenSky: A Large-Scale ADS-B Sensor Network for Research. *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, Berlin, April 2014, 313-314.
- [6] McCallie, D., Butts, J. and Mills, R. (2011) Security Analysis of the ADS-B Implementation in the Next Generation Air Transportation System. *International Journal* of Critical Infrastructure Protection, 4, 78-87. https://doi.org/10.1016/j.ijcip.2011.06.001
- [7] Rekkas, C. and Rees, M. (2008) Towards ADS-B Implementation in Europe. 2008 *Tyrrhenian International Workshop on Digital Communications—Enhanced Sur veillance of Aircraft and Vehicles*, Capri, 3-5 September 2008, 1-4. <u>https://doi.org/10.1109/TIWDC.2008.4649019</u>
- [8] Zhang, C.L., Wang, Y. and Zhang, C.Z. (2014) Enhanced ADS-B Reception Techniques Research. 10th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2014), Beijing, 26-28 September 2014, 657-661.
- [9] Habibi Markani, J., Amrhar, A., Gagné, J.-M. and Landry, R.J. (2022) NextGen ADS-B Software-Defined Reception with Enhanced Techniques. *Applied Sciences*, 13, Article 493. <u>https://doi.org/10.3390/app13010493</u>
- [10] RTCA (DO-260B) (2009) Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance-Broadcast (ADS-B) and Traffic Information Services-Broadcast (TIS-B). RTCA, Inc., Washington DC.
- [11] Chen, S.-Y. and Li, Y.-B. (2006) Error Correcting Cyclic Redundancy Checks based on Confidence Declaration. 2006 6th International Conference on ITS Telecommunications, Chengdu, 21-23 June 2006, 511-514. https://doi.org/10.1109/ITST.2006.288954
- [12] Ren, P., Wang, J. and Zhang, P. (2019) Novel Error Correction Algorithms for ADS-B Signals with Matched Filter Based Decoding. *Physical Communication*, 36, Article 100788. <u>https://doi.org/10.1016/j.phycom.2019.100788</u>
- [13] Delovski, T., Bredemeyer, J. and Werner, K. (2016) ADS-B over Satellite Coherent Detection of Weak Mode-S Signals from Low Earth Orbit. https://elib.dlr.de/106316/1/188Delovski.pdf
- [14] O'Shea, T. and Hoydis, J. (2017) An Introduction to Deep Learning for the Physical Layer. *IEEE Transactions on Cognitive Communications and Networking*, 3, 563-575. <u>https://doi.org/10.1109/TCCN.2017.2758370</u>
- [15] Kim, H., Jiang, Y., Kannan, S., Oh, S. and Viswanath, P. (2020) Deepcode: Feedback Codes via Deep Learning. *IEEE Journal on Selected Areas in Information Theory*, 1, 194-206. <u>https://doi.org/10.1109/JSAIT.2020.2986752</u>
- [16] Shafienya, H. and Regan, A. (2022) 4D Flight Trajectory Prediction Based on ADS-B data: A Comparison of CNN-GRU Models. 2022 *IEEE Aerospace Conference (AERO)*, Big Sky, MT, 5-12 March 2022, 1-12. https://doi.org/10.1109/AERO53065.2022.9843822
- [17] Adesina, D., Adagunodo, O., Dong, X. and Qian, L. (2019) Aircraft Location Pre-

diction using Deep Learning. *MILCOM* 2019-2019 *IEEE Military Communications Conference (MILCOM)*, Norfolk, VA, 12-14 November 2019, 127-132. https://doi.org/10.1109/MILCOM47813.2019.9020888

- [18] Habibi Markani, J., Amrhar, A., Gagné, J.-M. and Landry, R.J. (2023) Security Establishment in ADS-B by Format-Preserving Encryption and Blockchain Schemes. *Applied Sciences*, **13**, Article 3105. <u>https://doi.org/10.3390/app13053105</u>
- [19] sklearn.model\_selection.train\_test\_split. *scikit-learn*. <u>https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.train\_test</u> \_split.html
- [20] K. Team. Keras Documentation: EarlyStopping. https://keras.io/api/callbacks/early\_stopping/