

IAC-23-A6.9.10

Spacecraft Conjunction Assessment Optimisation Using Deep Learning algorithms applied to Conjunction Data Messages (CDMs)

Jose Javier Rosales Ruiz^{a*}, Nicola Garzaniti^b

^a *School of Aerospace, Transport and Manufacturing, Cranfield University, College Road, Cranfield, Bedfordshire, United Kingdom MK43 0AL, j.rosales-ruiz@cranfield.ac.uk*

^b *School of Aerospace, Transport and Manufacturing, Cranfield University, College Road, Cranfield, Bedfordshire, United Kingdom MK43 0AL, nicola.garzaniti@cranfield.ac.uk*

* Corresponding Author

Abstract

The lack of global regulations on space debris management during the early days of the space era until the last few decades of the 20th century resulted in a consistent increase in space debris. Spacecraft collisions in orbit and the industry's growing interest in launching constellations of satellites are now exacerbating the problem.

To address those concerns, multiple space organisations worldwide have implemented Situational Space Awareness programmes with integrated Conjunction Assessment systems that allow the detection of spacecraft conjunctions with an estimated collision risk probability. While this approach has proved effective in the last two decades, the foreseen increment of artificial space objects in orbit in the coming years will put any existing system under severe stress if the technology does not evolve to match the new demands.

The objective of this research is two-fold: it evaluates different architectures used in the field of Deep Learning to increase the accuracy of on-orbit Conjunction Events forecasting. It provides a multi-purpose modular, Machine Learning based Python library to support Conjunction Assessment activities.

The results of this study show that simpler cell architectures used in the Recurrent Neural Networks outperform the corresponding Vanilla versions in terms of accuracy for the problem at hand. It also demonstrates that the attention mechanism provides the best performance with up to 40% more accuracy.

Keywords: Recurrent Neural Network, Time Series Forecasting, Long Short-Term Memory, Gated Recurrent Unit, Conjunction Data Message, Spacecraft Conjunction Assessment

Nomenclature

Γ	Gate
i	Input
f	Forget
o	Output
t	Time t
x	Input
h	Hidden state
$\underline{h}$	Candidate hidden state
c	Cell state
$\underline{c}$	Candidate cell state
σ	Sigmoid activation function
W	Weight matrix
b	Bias
$\odot$	Hadamard product

Acronyms/Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
ASO	Artificial Space Object
CA	Conjunction Assessment
CCSDS	Consultative Committee for Space Data Systems
CDM	Conjunction Data Message
DL	Deep Learning
ESA	European Space Agency
GEO	Geostationary Orbit
GRU	Gated Recurrent Unit
LEO	Low Earth Orbit
LSTM	Long Short-Term Memory
RLV	Reusable Launch Vehicle
RNN	Recurrent Neural Network
SCA	Spacecraft Conjunction Assessment
SSA	Space Situational Awareness
TCA	Time of Closest Approach
TSF	Time Series Forecasting

1. Introduction

Since the beginning of the space era with the launch of the Russian satellite Sputnik on the 4th of October 1957, humankind has not ceased its activity in space to

exploit the full potential the near-Earth space environment offers. Communications and navigation, space exploration, weather monitoring, fire detection, and surveillance are just a few examples of the vast range of space applications in the civic, commercial and military domains. These activities play a vital role in satisfying our ever-increasing demands to be safer, more connected, and more knowledgeable about the universe surrounding us.

That continuous ambition to foster our civilisation's development has brought exponential progress in comfort and safety that have quickly been taken for granted. However, responsible sustainability regulations have not appropriately managed most of this activity since its beginning. Consequently, we are due to pay a high cost now.

1.1 The Kessler Syndrome

The first concerns about the potential impact of our continuous activity in space began to emerge back in the late 1970s when the former National Aeronautics Space Agency (NASA) scientists Donald J. Kessler and Burton G. Cour-Palais published the paper Collision frequency of artificial satellites: The creation of a debris belt. In this research paper, Kessler and Cour-Palais described a chain reaction caused by the continuous growth of the ASOs population in Low Earth Orbit (LEO) that increases the space density in orbit and the collision risk. If not managed in a controlled and timely manner, a higher collision risk would inherently lead to a greater ratio of collisions between objects every year, feeding back the space density with additional debris flux increasing the collision risk again in a vicious circle [1].

On the other side of the coin is the emerging trend in the Telecommunications industry to use constellations of satellites in an attempt to move them from GEO to LEO, drastically reducing launch costs and minimising space mission risks [2]. Only in the last three years SpaceX and OneWeb, two of the major private space companies, have launched more than two thousand new satellites with operational orbits in LEO, putting under severe stress the existing Space Traffic Management Systems (STMSs).

Consequently, the saturation that LEO is experiencing from our activity critically jeopardises the safety of space operations in all domains and creates global environmental damage with unforeseen negative consequences. Multiple actors that interact passively with the space environment are also hugely damaged from this issue: professional astronomers or stargazers, societies whose culture and religion have a profound foundation on the sky or the lives of many species in the animal kingdom that use the space environment instinctively are just a few examples [2]. The impact is too complex and progressing rapidly to be delimited and predicted accurately.

Numerous nations, organisations, and communities worldwide are working together to implement new regulations mostly concerned about disposal at the End-of-Life for new objects placed into orbit to contain the environmental issue and limit our impact. Unfortunately, many of these actions have incurred additional costs defrayed by all taxpayers and relevant stakeholders yet have proved insufficient [3].

One of the most relevant segments nowadays in the industry are the SSA programmes led by major Space Agencies such as NASA and ESA. These programmes aim to provide Spacecraft Collision Avoidance services by adopting a Conjunction Assessment process.

1.2 The evolution of Space Artificial Objects population

The importance of promoting sustainable operations in space and developing capabilities to manage it becomes self-evident when looking at the evolution of the ASOs population in space.

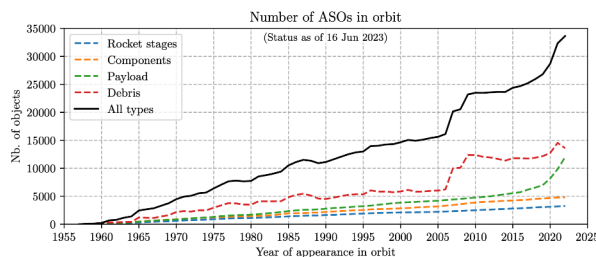


Figure 1. Number of artificial space objects in space.

A graphical representation of the evolution of the net number of all ASOs in space is shown in Figure 1. Using this view, it is clear that the net number of ASOs in the near-Earth orbits has never ceased to increase (inflow/outflow > 1), and the evolution of this growth is not constant, especially in the last two decades. The major contributors to this increment lately are Space Debris and Operational Payload objects launched (red and green lines, respectively).

1.2.1 Space Debris

Ignoring the ASOs population growth driven by conventional space activity, the most disruptive contribution to debris flux increase that abruptly broke with the historical trend was the two collisions in 2007 and 2009. Both events have put in evidence the vast potential for damage that collisions in orbit can have on the space environment:

- 2007 - The Chinese anti-satellite missile test destroys the weather satellite Fengyun-1C (750 Kg) at 863 Km of altitude, creating more than 3000 trackable debris.

- 2009 - Operational satellite Iridium-33 (560 Kg) and derelict Cosmos-2251 (950 Kg) collide at approximately 800 Km of altitude, creating another massive cluster of debris of more than 1000 trackable debris objects.

The reactivity of space agencies and communities worldwide following these events in developing and improving comprehensive CA systems has helped to prevent collisions from triggering the so-called *Kessler syndrome*, keeping the net number of debris objects roughly constant since 2010.

However, we are still far from reverting or containing the environmental impact with a sustainable space activity. Figure 1 shows how the evolution of the number of Payload objects has exponentially increased in the last few years. This steep growth breaking off the historical records is mainly driven by a new business concept emerging in the private sector: constellations of satellites.

1.2.2 Payload impact: Constellations of satellites

The growing number of constellations of satellites in LEO is the other side of the coin of the space environmental issue. Space activity derived from the communications domain has traditionally occurred in Geosynchronous Orbit. This conventional approach has mainly been driven by the limitations of the technology being used; better communications systems usually meant bigger satellites to be reliable and robust enough to provide good service during a noticeably long lifecycle of typically 10-15 years. This growth in size, in turn, increases the launch costs and the investment risk associated with a failed mission, which leads to fewer and more expensive launches in a vicious circle [4].

Nowadays, though, the development of profitable Reusable Launch Vehicles and technology that allows for the miniaturisation of satellites (minisatellites, microsatellites and nanosatellites) enabling the launch in batches of many units is shifting the communications domain interests towards LEO.

The logic behind this change is that it minimises the launch costs and risks of mission failure and has additional in-service operational benefits such as lower latency, higher resolution or a rapid decay of failed satellites, among others. However, the drawback is that many more satellites in LEO are required to achieve global coverage as they can cover a minimal region of the globe compared with satellites in Geosynchronous Orbit [2].

1.3 Spacecraft Collision Avoidance services

The development and implementation of SSA programmes based on global collaboration and cutting-edge technologies are critical to mitigate space activity's

impact on the space environment and guarantee safe operations in space through appropriate Space Traffic Management Systems.

Nowadays, many organisations worldwide have an active SSA programme in place. Supporting their respective SSA programmes, these organisations have dedicated Conjunction Assessment systems with architectures that automatically evaluate conjunction events among all known ASOs using data from global Space Surveillance and Tracking facilities. Additional data enhances the underlying assessment processes thanks to multiple agreements with other international organisations and spacecraft owners/operators.

1.4 Problem statement

Despite using different architectures and capabilities from one organisation to another, all CA systems follow a similar iterative approach when processing conjunction events at their coarsest levels. Conjunctions identified are evaluated by dedicated software to estimate the collision risk probability, and results are stored in a temporary database for a close follow-up in subsequent iterations [5].

The main communication channel for collision risk analysis and notifications uses the so-called Conjunction Data Messages (CDMs), a self-contained set of space data in a *key-value* configuration stored in an ASCII format text file. It contains relevant information about the conjunction event between OBJECT1 (O1) -the satellite or spacecraft of interest- and OBJECT2 (O2).

These dedicated messages are produced every time a conjunction takes place with an estimated collision risk probability at TCA above a certain threshold. Upon CDM receipt, dedicated flight dynamics teams assess whether or not an avoidance manoeuvre is required to mitigate the risk of collision.

The final decision comes from a trade-off between timing and estimated collision risk probability accuracy. The objective is to perform the avoidance manoeuvre as soon as possible to optimise costs and maximise operational lifetime.

Unfortunately, the estimations' accuracy only improves with the number of screenings (number of CDMs). Conversely, as the number of conjunctions increases, the lead time for the decision-making for the collision avoidance manoeuvre reduces. Consequently, the lack of visibility on the evolution of these conjunction events is one of the primary concerns for the spacecraft's owners/operators, whose main objective is to react accurately as soon as possible to incur the lowest operational cost while minimising the risk.

Therefore, It is paramount to develop technical solutions that can provide forward visibility on the evolution of the conjunction events as accurately as

possible regardless of the number of screenings already performed.

1.5 Objectives, scope and rationale of the research

1.5.1 Objective

The primary purpose of this study is to increase the efficiency of the Conjunction Assessment process by providing more accurate Conjunction Event Forecasting capabilities.

Therefore, this research aims to provide a flexible **RNN model architecture** to maximise the forecast accuracy of the conjunction events evolution using associated CDM(s) received with the latest Deep Learning techniques. Additionally, an assessment of the different configurations is considered to propose a refined model.

1.5.2 Scope definition

It is part of the scope of the research:

- The evaluation and performance assessment of multiple RNN configurations using different variations of RNN-Cell architectures.
- The development of a multi-purpose modular Python library for on-orbit to complement and support Conjunction Assessment activities.
- The use of CDMs compulsory data as defined by the [Recommended Standards for Conjunction Data Messages CCSDS 508.0-B-1](#) document.

It is out of the scope of the research:

- Any performance assessment and evaluation of alternative methods to approach TSF problems other than those based on RNNs.
- The evaluation of the absolute accuracy of the final Forecasting model under the premise of using *Kessler's* performance as the baseline [6]. A higher accuracy vs. *Kessler* implies a higher absolute accuracy.
- The optimisation of the most optimal RNN configuration to maximise accuracy.

1.5.3 Rationale of the research

In 2020, a novel probabilistic generative model for synthetically generating conjunction data messages was proposed by Acciarini et al. [7] in order to help in making predictions and finding the parameters that explain the observed data in CDMs, and enabling the generation of physically accurate synthetic datasets.

Parallely, Pinto et al. [8] introduced an RNN architecture using LSTM Vanilla cells following a Bayesian deep-learning approach in the same year. This solution made a considerable step forward to support on-orbit conjunction assessments. It allowed the modelling of CDM data and provided forecasts with associated uncertainties on the event's evolution.

The culmination of both works came one year later with their integration into a new open-source Python package for machine learning (ML) applied to collision avoidance called *Kessler* [6].

Although *Kessler* is positioned as one of the most comprehensive machine learning solutions for collision avoidance to our awareness, its RNN model still has room for improvement on two main points: the internal architecture (based on LSTMs) and its minimal flexibility to implement cutting-edge deep learning techniques on its network. Both limitations not only impose a severe restriction on the capabilities to solve the problem it was designed for (i.e. maximising accuracy and minimising uncertainty) but also hinder its adaptation as the industry requirements evolve (i.e. scalability). Hence, there is a need to adapt its framework to enable such improvements and evaluate what other RNN model configurations could increase the performance of the Machine Learning (ML) library.

2. Material and methods

2.1 Research methodology

This study performs a preliminary quantitative analysis of the evolution of ASOs population in low/medium Earth orbits. The data source used for this part of the research is the General Catalog of Artificial Space Objects (GCAT), created and maintained by McDowell [9]. Since the multiple datasets composing this catalogue follow a relatively complex architecture with lengthy formatting standards, some data preprocessing is performed in Python using Jupyter Notebooks to extract relevant information directly from the raw data for this project. The analysis outputs highlight the root cause of the practical problem: debris and constellations of satellites (Section 1.2).

On the other hand, a qualitative analysis is performed on the Spacecraft Collision Avoidance services using NASA and ESA as reference organisations. The objective is to understand in general terms what tools and processes they have in place to avoid collisions on orbit and analyse the workflow of their CA segment from a qualitative perspective. The information used for this general assessment was taken from the *Spaceflight Safety Handbook for Satellite Operators* [10] available on the public website <https://www.space-track.org/> in the case of USSPACECOM and from papers published by Merz et al. in the European Conferences on Space Debris of

2017 and 2021 respectively for the case of ESA's SDO [5, 11].

The outputs of this qualitative analysis not only evince synergies in the way of working but also pave the way for the definition of the research problem, which is the optimisation of the lead-time on decision-making for required avoidance manoeuvres. From the multiple perspectives that the research problem can be addressed, this study focuses on enabling accurate Conjunction Event forecasting capabilities to provide some anticipation of hazardous conjunctions.

2.2 Development methodology

To set the baseline, a preliminary qualitative assessment is carried out of *Kessler*, the most recent solution developed in the domain of our awareness to tackle the same research problem.

This assessment of *Kessler*'s internal architecture enables the identification of the root cause of its bottleneck in terms of performance, which is its restrictive network configuration built upon the consideration that LSTM cells are the best performers for the problem at hand.

Having identified the opportunities for improvement in the RNN architecture used in *Kessler*, this study then takes two parallel technical workstreams using Python programming language:

1. *Kessler*'s internal architecture is taken as the baseline and adapted to use a broader range of custom RNN models. The new architecture is also enriched with additional modules, such as a Synthetic CDM Generator. The adapted architecture and the new modules implemented are part of a multi-purpose modular Python library named Scalib (Spacecraft Conjunction Assessment Library).
2. Additional RNN-Cell architectures and network configurations are explored with a particular focus on TSF tasks. In this workstream, the different cells and configurations are evaluated and compared with *Kessler*'s performance in terms of accuracy of prediction. Since different RNN-Cells and network configurations (i.e., attention mechanism) lead to different computational costs (number of learnable parameters to train), the settings for every model are adjusted accordingly to match *Kessler*'s computational cost as much as possible. The equivalence in computational cost guarantees fairness when ranking the best configurations for Conjunction Event Forecasting and proves that simpler models and models implementing the attention mechanism outperform *Kessler*'s model. Such assessment, ranking, and results evaluation are performed using

the Jupyter Notebooks environment and discussed in more detail in Section 4.

For the development of both workstreams, it has been used two anonymised datasets of conjunction events available to the public for the Collision Avoidance Challenge competition organised by ESA's Advanced Concepts Team (ACT) in partnership with ESA's Space Debris Office in 2019¹.

3. Theory and calculation

3.1 The problem of Time Series Forecasting

Among all data regression problems, Time Series Forecasting (TSF) is one of the most difficult due to its extensive applicability and complex nature in most cases.

Nowadays, TSF problems are approached using three methods: traditional, machine learning, or deep learning. Although simpler, traditional methods present critical limitations on their performance and applicability. They can only deal with stationary behaviours by extracting patterns from the closest historical data to the next prediction. Some algorithms attempt to overcome this limitation by transforming the time series into a stationary one by merely addressing drift, seasonality, autocorrelation, and heteroscedasticity. However, this is often not feasible on time series with complex behaviours and, hence, the need for more advanced models based on machine learning [12, 13].

With the advent of Artificial Intelligence, sequence modelling techniques in ML have continuously grown to handle Natural Language Processing (NLP). The need to develop neural networks capable of learning from past inputs led to the creation of RNNs, a particular type of neural network that considers previous data and computational parameters to predict the following input. Due to the implicit similarity NLP has in sequence prediction and behaviour complexity, RNNs are now considered promising for TSF, with some developments already providing excellent results [14].

3.2 Recurrent Neural Networks for Time Series

RNNs are the most used Machine Learning models to address TSF tasks because they can find tailored solutions to specific problems by altering their internal architecture. Typically, RNN models are built as a sequence of the same neuron/cell structure, which is the core of all computations that occur in the network. The problem with this approach is that it is often unclear what RNN-Cell architecture shall be used to address a specific task. In general, when facing a new problem, the most

popular cell architecture (LSTM Vanilla/*Kessler*) is used as the building block for the RNN model without any guarantee of its appropriateness to the current data and associated time series behaviour.

However, it is a must to perform a preliminary assessment of the different architectures developed in the industry to identify the best configurations to tackle the problem at hand.

3.2.1 Introduction to RNN Cells

The first RNN-Cell architectures developed for this new type of neural network were designed to take at each time step the previous hidden state² [15], the previous output state³ [16], or a combination of both, as shown in Figure 2. However, these configurations suffer from vanishing and exploding gradient problems that impede capturing long-term dependencies when doing back-propagation through time [17].

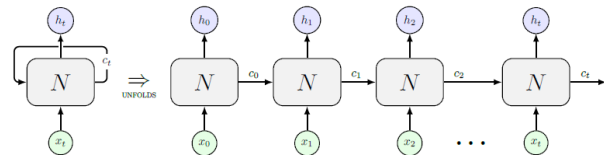


Fig. 2. Information flow at a typical RNN-Cell.

To overcome this limitation, different types of cell architectures based on the *gating mechanism* have been developed to regulate the flow of the information flux. These gates are hidden layers of neurons that act as filters to keep helpful information and selectively remove any irrelevant information from passing through. This information control process takes place by training the parameters underlying the neurons in the gate using the back-propagation through time algorithm [18], providing the RNN cell with an internal permanent memory able to store valuable information for long periods.

The three main cell architectures used in RNNs and associated variations are presented. Appendices A, B and C comprehensively describe the cell computations performed in every variation.

3.2.1 LSTM Vanilla cell

The architecture of *Kessler*. It is the most widely used architecture for TSF tasks [19]. It has one input gate, one forget gate and one output gate to filter the most relevant information and discard the irrelevant ones. Figure 3 shows the internal structure of the LSTM Vanilla cell.

¹ <https://kelvins.esa.int/collision-avoidance-challenge/home/>

² Cell working memory.

³ Cell long memory.

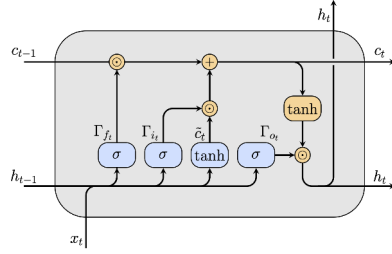


Fig. 3. LSTM Vanilla cell architecture.

The internal computations occurring within the cell are described with Equations 1-6.

$$\tilde{c}_t = \tanh(x_t W_{xc} + h_{t-1} W_{hc} + b_c) \quad (1)$$

$$\Gamma_{i(t)} = \sigma(x_t W_{xi} + h_{t-1} W_{hi} + b_i) \quad (2)$$

$$\Gamma_{f(t)} = \sigma(x_t W_{xf} + h_{t-1} W_{hf} + b_f) \quad (3)$$

$$\Gamma_{o(t)} = \sigma(x_t W_{xo} + h_{t-1} W_{ho} + b_o) \quad (4)$$

$$c_t = \Gamma_{f(t)} \odot c_{t-1} + \Gamma_{i(t)} \odot \tilde{c}_t \quad (5)$$

$$h_t = \Gamma_{f(t)} \odot h_{t-1} + \Gamma_{o(t)} \odot \tanh(c_t) \quad (6)$$

3.2.2 GRU Vanilla cell

A simplified variant of the LSTM Vanilla cell with only one update and relevance gates [20]. Designed to reduce the number of parameters to train. Figure 4 shows the internal structure of the GRU Vanilla cell.

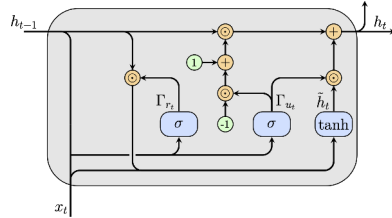


Fig. 4. GRU Vanilla cell architecture.

$$\Gamma_{u(t)} = \sigma(x_t W_{xu} + h_{t-1} W_{hu} + b_u) \quad (7)$$

$$\Gamma_{r(t)} = \sigma(x_t W_{xr} + h_{t-1} W_{hr} + b_r) \quad (8)$$

$$\tilde{h}_t = \tanh(x_t W_{xh} + (\Gamma_{r(t)} \odot h_{t-1}) W_{rh} + b_h) \quad (9)$$

$$h_t = \Gamma_{u(t)} \odot h_{t-1} + (1 - \Gamma_{u(t)}) \odot \tilde{h}_t \quad (10)$$

3.2.3 MGU Vanilla cell

It has the simplest architecture compared to LSTM and GRU cells, as it only has one forget gate [21]. As with GRU, it is designed to reduce the number of parameters further to train. Figure 5 shows the internal structure of the MGU Vanilla cell.

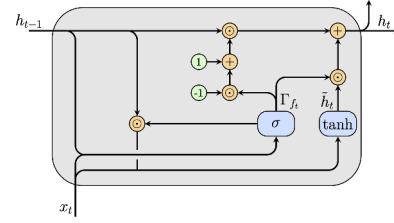


Fig. 5. MGU Vanilla cell architecture.

$$\Gamma_{f(t)} = \sigma(x_t W_{xf} + h_{t-1} W_{hf} + b_f) \quad (11)$$

$$\tilde{h}_t = \tanh(x_t W_{xh} + (\Gamma_{f(t)} \odot h_{t-1}) W_{fh} + b_h) \quad (12)$$

$$h_t = \Gamma_{f(t)} \odot h_{t-1} + (1 - \Gamma_{f(t)}) \odot \tilde{h}_t \quad (13)$$

3.2.4 SLIM - A common method for model parameters reduction

Apart from the cells mentioned above and associated variations, in 2017, a new method for parameter reduction common for the Vanilla versions of LSTM, GRU and MGU was proposed in an attempt to lighten their structure and speed up the learning process while maintaining the same number of gates to regulate the information flux [22, 23, 24]. This method led to the creation of the following three additional variations:

- **SLIM1** - Removes the dependency on the input tensor and its associated weight matrix from the input, forget and/or output gates.
- **SLIM2** - Removes the dependency on the input state, its associated weight matrix and any bias term bg from the input, forget and/or output gates.
- **SLIM3** - Removes the dependency on the input tensor, previous hidden states and their associated weight matrices from the input, forget and/or output gates.

3.2.5 The Attention Mechanism

Although LSTM, GRU and MGU models were specifically designed to overcome the problem of vanishing and exploding gradients, their performance in resolving TSF tasks still has considerable room for improvement. Different adaptations specifically developed to tackle Time Series and sequence-to-sequence problems, such as stacked or bi-directional LSTMs, have improved their accuracy [25]. However, they are still a very conservative approach, which may lead to configurations with a modest performance and computational cost ratio.

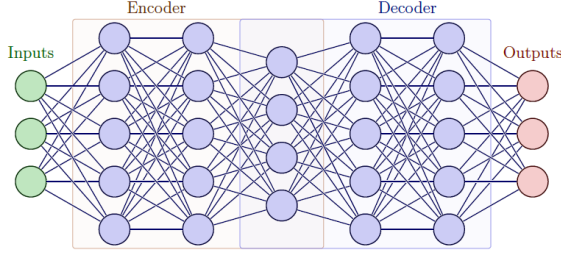


Fig. 6. RNN with Encoder-Decoder architecture.

Another approach that has yielded good results is the sequence-to-sequence modelling architecture in traditional neural models for NLP problems. These models aim at implementing a deep neural network consisting of an encoder-decoder architecture, as shown in Figure 5, that is adjusted to maximise performance. In this architecture, the encoder transforms an initial sequence into a fixed-length vector that the decoder takes to produce the predicted sequence.

However, in 2014, Bahdanau et al. [26] foresaw that using a fixed-length vector was a potential bottleneck in improving the performance of this basic encoder-decoder architecture as it only passed to the decoder the last hidden state of the sequence. This inherently limits the model's accuracy since the cells' memory is insufficient to tackle the complexity of some TSF problems. It is the case, for instance, of Time Series such as Conjunction Events CDMs where the latest items in a given sequence might have different relevance compared to others. In Conjunction Events forecasting, the more observations available (number of CDMs), the more accurate the data values are and, therefore, more relevant to produce the next prediction.

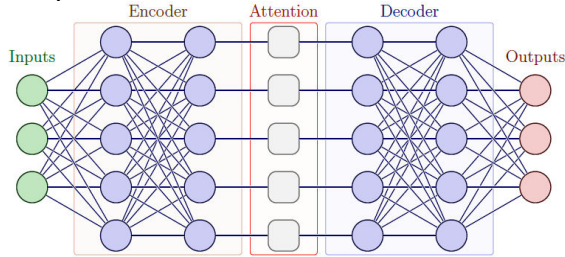


Fig 7. RNN with attention mechanism.

To overcome this limitation, Bahdanau et al. [26] proposed a new method in which the different parts of the source sequence are soft-weighted according to their relevance to predicting the following sequence, the so-called attention mechanism. In this mechanism, the encoder passes the last and all hidden states from all time steps to an attention layer, as shown in Figure 7. This layer then compares the hidden states from all time steps coming out from the encoder with all hidden states in the initial layer of the decoder (target values) and computes

a weight vector to put more "attention" on the required time steps.

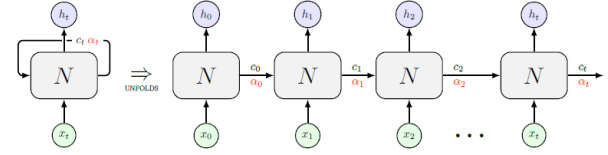


Fig. 8. Attention mechanism at the neuron level.

The process that takes place at the neuron level is shown in Figure 8. The last layer of the encoder is passed to the attention layer, which weights hidden states with an attention function α . Since the decoder provides the hidden states from all time steps, the workload of the architecture increases; however, it has more information to fine-tune the inputs the decoder receives, which considerably improves the accuracy [27].

3.3 Performance scores used

The following performance scores are used for the RNNs evaluation:

- Mean Squared Error (MSE)

$$MSE = \frac{1}{n} \sum_{t=1}^n (Y_a^t - Y_p^t)^2 \quad (17)$$

- Mean Absolute Error (MAE)

$$MAE = \frac{1}{n} \sum_{t=1}^n (Y_a^t - Y_p^t) \quad (18)$$

- Mean Absolute Percentage Error (MAE)

$$MAE = \frac{1}{n} \sum_{t=1}^n \frac{(Y_a^t - Y_p^t)}{Y_a^t} \quad (19)$$

- Prediction Of Change In Direction (POCID)

$$POCID = \frac{1}{n} \sum_{i=1}^n D^t \quad (20)$$

$$D^t = \begin{cases} 1 & \text{if } (Y_p^{t+1} - Y_p^t)(Y_a^{t+1} - Y_a^t) \geq 0 \\ 0 & \text{if } (Y_p^{t+1} - Y_p^t)(Y_a^{t+1} - Y_a^t) < 0 \end{cases}$$

- Bayesian Information Criterion (BIC)

$$BIC = n \log(MSE) + k \log(n) \quad (21)$$

- Multi-Criteria Index Measure (MCIM)

$$MCIM = \frac{MSE_n + MAE_n + MAPE_n + (1-POCID_n) + BIC_n}{5} \quad (22)$$

Where n is the input size, Y_a is the target values, Y_p is the predicted values, and k is the number of learnable parameters of the model.

4. ScaPy library

A new Machine Learning Python library called Spacecraft Collision Assessment for Python (*ScaPy*) is proposed in the frame of the present research to cover the gaps identified in *Kessler*.

- Flexible architecture: the RNN network setting is defined by the user and the data available instead of being pre-defined by the architecture.
- Includes a hybrid model for collision risk evaluation that can either:
 - Estimate the collision probability risk through regression.
 - Identify on-orbit Conjunctions with high risk of collision through classification.
- Includes a module for Synthetic CDM data generation to enable future developments.



Fig. 9. ScaPy library logo

The tool is publicly available in GitHub using the link <https://github.com/JJavierRosales/scapy.git>⁴.

5. Results and discussion

Implementing all the different configurations of LSTM, GRU and MGU cells along with the attention mechanism in *ScaPy*, this study evaluated 23 different RNN networks vs. *Kessler* architecture (using LSTM Vanilla).

All configurations have been adjusted to have the same number of learnable parameters as *Kessler* as a proxy of computational cost. For this evaluation, *Kessler* is configured with an RNN network model with an input and output size of 66 features, a stacked LSTM Vanilla layer with a depth of 2 and 264 hidden neurons (4 per feature), a dropout layer with a dropout probability of 0.2 and a fully connected layer at the end of the network, embedding a total of 927,762 learnable parameters. Models have been tested on a test dataset with 100 Conjunction Events using a batch size of 10 events per batch. Results provided show the average of MSE and MAE scores.

⁴ The location of the repository may change. For any technical issues accessing the repository please contact the authors.

For the evaluation of the relative performance, the performance scores introduced in Section 3.3 have been used to compute MCIM, a metric to take into account all scores results by using their normalised values [14]. Table 1 shows the results of such performance scores per RNN configuration.

It is straightforward to conclude that the LSTM Vanilla cell architecture is not the most efficient configuration for Conjunction Events forecasting since 21 other configurations outperform its accuracy. Among the 21 variants, six stand out: LSTM, GRU and MGU using SLIM3 parameter reduction method and three networks using the attention mechanism with LSTM, GRU and MGU Vanilla cell architectures as encoder-decoders, respectively.

The slight accuracy improvement of the other 15 variants above LSTM Vanilla (ranging from 7 to 21 in Table 1) is primarily due to the considerably increased number of hidden neurons introduced in the model to pair the model complexity of *Kessler* configuration (BIC score). In this regard, more hidden neurons per feature can balance the performance of simpler architectures and even allow for small accuracy improvements compared to the LSTM Vanilla architecture (up to 5% improvement).

Table 1. Performance scores per RNN configuration

RNN Configuration	MSE	MAE	MAPE	POCID	BIC	MCIM
1 GRU ATTN-VANILLA	0.040	0.046	0.018	0.028	0.000	0.026
2 MGU ATTN-VANILLA	0.000	0.143	0.000	0.000	0.423	0.113
3 LSTM ATTN-VANILLA	0.323	0.234	0.156	0.334	0.721	0.354
4 MGU SLIM3	0.429	0.279	0.938	0.804	0.406	0.571
5 LSTM SLIM3	0.422	0.099	0.972	0.786	0.676	0.591
6 GRU SLIM3	0.359	0.000	0.857	0.908	0.832	0.591
7 GRU SLIM2	0.771	0.543	0.657	0.880	0.382	0.647
8 GRU SLIM1	0.720	0.534	0.680	0.857	0.595	0.677
9 GRU VANILLA	0.661	0.594	0.743	0.928	0.487	0.683
10 LSTM PC	0.829	0.760	0.729	0.878	0.283	0.696
11 LSTM NOG	0.682	0.570	0.746	0.855	0.763	0.723
12 MGU SLIM2	0.791	0.666	0.825	0.873	0.484	0.728
13 LSTM NIG	0.709	0.594	0.697	0.893	0.763	0.731
14 LSTM SLIM2	1.000	0.672	0.855	0.926	0.234	0.737
15 GRU MUT3	0.681	0.612	0.730	0.923	0.763	0.742
16 MGU VANILLA	0.785	0.769	0.906	0.944	0.316	0.744
17 MGU SLIM1	0.768	0.691	0.938	0.736	0.609	0.748
18 LSTM NOGAF	0.668	0.634	0.824	0.820	0.841	0.757
19 LSTM NIGAF	0.663	0.643	0.824	0.881	0.841	0.770
20 LSTM SLIM1	0.913	0.637	0.800	1.000	0.517	0.773
21 LSTM NFGAF	0.676	0.632	0.810	0.934	0.841	0.779
22 LSTM VANILLA	0.749	0.638	0.747	0.949	0.841	0.785
23 LSTM FB1	0.816	0.697	0.706	0.848	1.000	0.813
24 LSTM NFG	0.969	1.000	1.000	0.931	0.763	0.933

Figure 10 translates to a visual representation the results provided in Table 1. The box plot indicates how reliable the different models are when predicting new values, that is, how similar the MSE loss is between different predicted values for the ten batches of the test dataset. Narrower boxes indicate more reliable models.

Figure 11 shows the validation loss evolution during model training per epoch for the top 6 best-performing models vs. LSTM Vanilla. They all outperform the LSTM Vanilla architecture with ten training epochs and reach their minimum MSE validation loss at around 25 epochs.

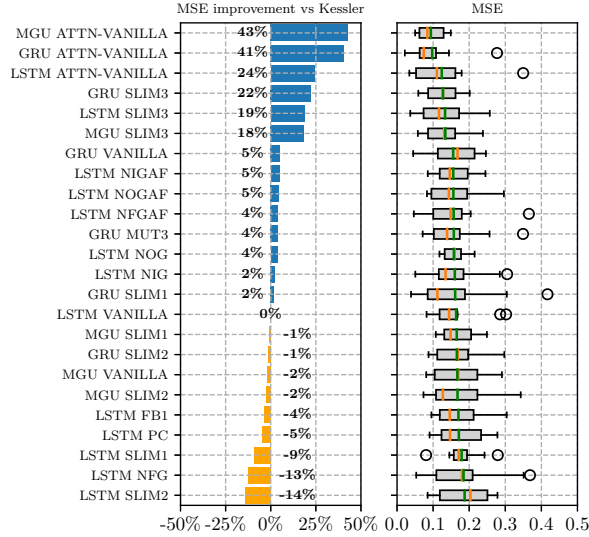


Fig. 10. MSE loss per model vs *Kessler* (LSTM Vanilla). (Box plot: mean in green, median in orange).

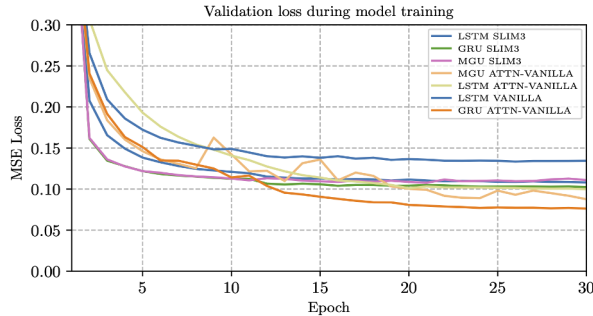


Fig. 11. Validation loss of the top 5 models vs. LSTM Vanilla during model training.

6. Conclusions

In this study, we demonstrate that the most used cell architecture in RNN models, such as LSTM Vanilla, does not necessarily perform the best on specific Time Series Forecasting tasks. Therefore, a pre-assessment of different Recurrent Neural Network configurations and cell structures must be made to ensure the appropriateness of the model to address a specific task.

For the problem approached in this research (Spacecrafts on-orbit Conjunction Events Forecasting), we have found a series of RNN networks that can improve between 18% up to 43% the accuracy obtained in the tool *Kessler*, based on a conventional LSTM Vanilla architecture.

These candidates are RNN networks with LSTM, GRU and MGU cell architectures using SLIM3 parameter reduction method (Section 3.2.4) and three networks using the attention mechanism with LSTM, GRU and MGU Vanilla cells in the encoder and decoder (Section 3.2.5).

The best network configuration found in our study includes the attention mechanism with a Gated Recurrent Unit Vanilla cell structure in its encoder and decoder.

Acknowledgements

Thanks to Dr Nicola Garzaniti for the guidance, feedback and recommendations provided throughout the research.

Thank you to my parents, Carmen and José, for their love, passion and courage to devote their lives to give me the wings I needed to live my dream. To my sister and her family for giving me the two brightest stars of my universe.

References

- [1] Kessler, D. J. and Cour-Palais, B. G. (1978). Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research*, 83:2637–2646.
- [2] Lawrence, A., Rawls, M. L., Jah, M., Boley, A., Di Vruno, F., Garrington, S., Kramer, M., Lawler, S., Lowenthal, J., McDowell, J., and McCaughrean, M. (2022). The case for space environmentalism. *Nature Astronomy*, 6(4):428–435.
- [3] Long, G. (2020). The impacts of large constellations of satellites. Technical Report JSR-20-2H, JASON Program Office: The Mitre Corporation, 7515 Colshire Drive McLean, Virginia. Updated: January 21, 2021.
- [4] Wertz, J., Everett, D., and Puschell, J. (2011). *Space Mission Engineering: The New SMAD*. Space technology library. Microcosm Press.
- [5] Merz, K., Siminski, J., Virgili, B. B., Braun, V., Flegel, S., Flohrer, T., Funke, Q., Horstmann, A., Lemmens, S., Letizia, F., Mclean, F., Sanvido, S., and Schaus, V. (2021). ESA's space debris office collision avoidance service: Current status and special cases. In 8th European Conference on Space Debris. ESA Space Debris Office.
- [6] Acciarini, G., Pinto, F., Letizia, F., Martinez-Heras, J. A., Merz, K., Bridges, C., and Baydin, A. G. (2021). *Kessler*: a machine learning library for spacecraft collision avoidance. In the 8th European Conference on Space Debris. ESA Space Debris Office.
- [7] Acciarini, G., Pinto, F., Metz, S., Boufelja, S., Kaczmarek, S., Merz, K., Martinez-Heras, J. A., Letizia, F., Bridges, C., and Baydin, A. G. (2020). Spacecraft collision risk assessment with probabilistic programming. In the third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.
- [8] Pinto, F., Acciarini, G., Metz, S., Boufelja, S., Kaczmarek, S., Merz, K., Martinez-Heras, J. A., Letizia, F., Bridges, C., and Baydin, A. G. (2020). Towards automated satellite conjunction management with Bayesian deep learning. In AI for Earth Sciences Workshop at NeurIPS 2020, Vancouver, Canada.
- [9] McDowell, J. C. (2020). General catalog of artificial space objects (GCAT). <https://planet4589.org/space/gcat/>. Release 1.2.12 - (accessed 16.07.2023).
- [10] 18th & 19th Space Defence Squadron (2023). Spaceflight Safety Handbook for satellite operators. https://www.space-track.org/documents/SFS_Handbook_For_Operators_V1.7.pdf. Version 1.7. (accessed 12.04.2023).
- [11] Merz, K., Bastida Virgili, B., Braun, V., Flohrer, T., Funke, Q., Krag, H., and Lemmens, S. (2017). Current collision avoidance service by ESA's space debris office. In T. Flohrer, F. S., editor, 7th European Conference on Space Debris. ESA Space Debris Office.
- [12] Adhikari, R. and Agrawal, R. K. (2013). An introductory study on time series modelling and forecasting.
- [13] Shi, J., Jain, M., and Narasimhan, G. (2022). Time series forecasting (TSF) using various deep learning models.
- [14] Khaldi, R., Afia, A. E., Chiheb, R., and Tabik, S. (2022). What is the best RNN-cell structure for forecasting each time series behaviour?
- [15] Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.
- [16] Jordan, M. I. (1997). Chapter 25 - serial order: A parallel distributed processing approach. In Donahoe, J. W. and Packard Dorsel, V., editors, *Neural-Network Models of Cognition*, volume 121 of *Advances in Psychology*, pages 471–495. North-Holland.
- [17] Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks.
- [18] Werbos, P. (1990). Back-propagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.
- [19] Gers, F., Schmidhuber, J., and Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation*, 12:2451–71.
- [20] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation.
- [21] Zhou, G.-B., Wu, J., Zhang, C.-L., and Zhou, Z.-H. (2016). Minimal gated unit for recurrent neural networks.
- [22] Lu, Y. and Salem, F. M. (2017). Simplified gating in long short-term memory (LSTM) recurrent neural networks.
- [23] Dey, R. and Salem, F. M. (2017). Gate-variants of gated recurrent unit (GRU) neural networks.
- [24] Heck, J. and Salem, F. M. (2017). Simplified minimal gated unit variations for recurrent neural networks.
- [25] Chandra, R., Goyal, S., and Gupta, R. (2021). Evaluation of deep learning models for multi-step ahead time series prediction. *IEEE Access*, 9:83105–83123.
- [26] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate.
- [27] Wen, X. and Li, W. (2023). Time series prediction based on LSTM-attention-LSTM model. *IEEE Access*, 11:48322–48331.



DSSPACE

<https://dspace.org/>

Spacecraft conjunction assessment optimisation using deep learning algorithms applied to conjunction data messages (CDMs)

Rosales Ruiz, Jose Javier; Garzaniti, Nicola

2023-10-06

International Astronautical Federation (IAF)

<https://dl.iafastro.directory/event/IAC-2023/paper/76773/>

Rosales Ruiz JJ, Garzaniti N. (2023) Spacecraft conjunction assessment optimisation using deep learning algorithms applied to conjunction data messages (CDMs). In: 74th International Astronautical Congress (IAC) 2023, 2-6 October 2023, Baku, Azerbaijan, Paper number 76773