المؤتمر الأول للابتكار وريادة الأعمال في الجامعات السعودية(فكرة واختراع وأثر) ، م ١ ع١، ص: ٢٧١- ٢٨٥(٢٠٢٥م)
DOI:10.4197/Ieckau.1-1.13

Classifying Spacecraft Collision Risk: A Machine Learning Approach Using GRU Neural Networks

Amirah Manyur Almutairi[1] and Rana Abdulaziz Alzahrani[2]

[1]Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia; ameramun2002@gmail.com

[2]Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia; ranaalzahrani047@gmail.com

Abstract. As the density of space debris in low Earth orbit increases, the likelihood of or- bital collision risks in space rises significantly over time. This risk poses a critical and significant challenge for the global space sector. With the advancement of technology, particularly in artificial intelligence and machine learning, and their increasing application in solving real-world problems, our study focuses on utilizing these technologies, specifically by employing GRU (Gated Recurrent Units) networks. We utilized data from the” Collision Avoidance Challenge” released by the European Space Agency in 2019 to classify collision risks between spacecraft and other objects. By analyzing temporal patterns and the proximity of objects to satellites, we distinguished high- risk cases from low-risk ones. The data contains a large number of Conjunction Data Messages (CDMs), with over 162,000 messages included. Due to this large number, we needed to preprocess the data, which involved several steps, including imputing missing values using linear interpolation, standardizing features, and converting risk values into binary classifications. Several algorithms were applied to select the most influential features on the likelihood of a collision, combining statistical analysis (Pear- son and Spearman), mutual information, SHAP analysis, and permutation importance, ultimately resulting in the selection of the best 25 features using ensemble feature se- lection methods. The model was trained on sequential time-series data, and we used a Masking layer to address the issue of unequal lengths in the data sequences. Additionally, Dropout layers were applied, which significantly helped reduce the problem of overfitting. The model achieved high performance, with a verification accuracy of 97%, demonstrating its effectiveness in classifying collision risks. The results of this study highlight the potential of deep learning in solving orbital collision problems in space, enhancing collision prediction systems, and enabling proactive maneuver planning in satellite operations.

Keywords: Deep Learning; Satellite Collision; GRU; Risk Classification; Low Earth Orbit

1. Introduction

With the beginning of the space revolution and inter- national competition in this sector, Earth orbits, especially low-Earth orbit, have seen crowds of space objects from moons, debris, and unknown objects. As shown in Figure 1, there are many objects shown as white dots that increase in density at the closest orbit to the Earth and decrease outward to the farthest orbit.

They have been tracked by NASA's Orbital Debris Program Office (ODPO) as of 2019 (Office, 2019). According to Office (2019), about 95% of the objects in this illustration are orbital debris, and the number of particles larger than 1 mm in size in low Earth orbit has been estimated at more than 100 million. These objects are typically 10 centimeters or larger, but there are many other pieces of debris too small to be detected that pose a threat to spacecraft. With the continuous growth of space debris in orbit, collisions increase, which in turn leads to more

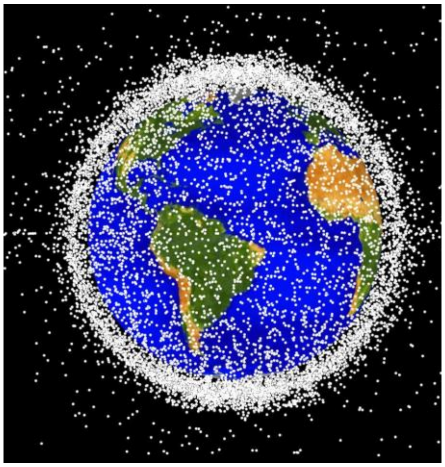

Figure 1: LEO objects tracked by NASA ODPO.

debris, and this is known as Kessler syndrome (Kessler and Cour-Palais, 1978). With the acceleration of industry in the space sector and the rise of companies in many countries, the risk of orbital collisions has become a pressing global challenge. Estimating the probability of collision risk and classifying it as high or low is an essential task to protect active space- craft from colliding with other space objects, as this allows scientists to make well-informed decisions about the need to perform potential avoidance maneuvers.

This study aims to apply advanced machine learning techniques specifically recurrent neural networks to classify collision risk more accurately. The study relied on classification as the crucial matter to take precautionary measures, and by knowing if the risk is high, maneuvers are performed. To achieve this goal, data from the Collision Avoidance Challenge launched by the European Space Agency (ESA) was studied to explore the possibility of using machine learning in space collision risk estimation (Team, 2019).

2. Materials and Methods

2.1 Data Source and Description

This study relied on orbital proximity data collected by the European Space Agency, specifically from the Space Debris Office in 2019, to analyze and monitor the risks of space collisions (Team, 2019). These data were published as part of the Collision Avoidance Challenge launched by the space agency. The agency relies on monitoring close approaches between satellites and space debris. When a potential close approach is detected, a Conjunction Data Message (CDM) is issued. The database contains 162,634 records distributed across 103 different features. This large amount of data has provided a rich basis for developing machine learning models to analyze and assess the risks of satellites approaching other space objects. The data consists of a set of events, each representing a potential close approach between a

satellite and another space object. Each event is identified by a unique identifier(event id). Each event consists of several CDMs (see Figure 3) representing time points that express the close approach between the two objects. Each CDM contains a set of data, including the satellite ID, the approaching object ID, the time of closest approach (TCA), the hazard level, and other relevant information. Three CDMs are typically published daily for each approach, covering a period of up to one week. The data is thus a time series of CDM messages. Each event is classified according to its calculated risk value. An event classified with a risk value greater than -6 at the last approach is considered a high-risk event, and a low-risk event if the value is less than or equal to -6 (Uriot et al., 2020).

2.2 Exploratory Data Analysis

In this section, the nature and distribution of the data were examined to determine whether there were any anomalies affecting the machine learning approach. 2 shows the distribution of risk values, which is the target feature. We can see from the image that low-risk values are frequent in the data, while high-risk values are rare. This problem can cause bias during model training for classification.

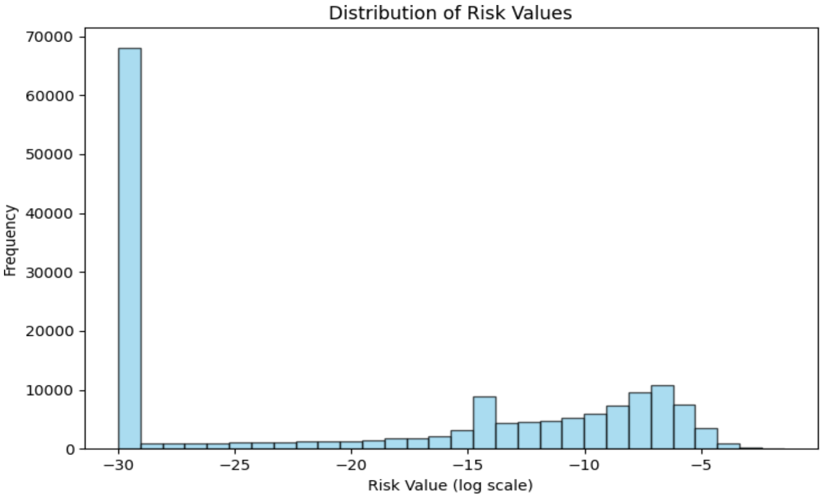

Figure 2: Distribution of risk values in the dataset.

Figure 3 shows the distribution of coupling data messages (CDMs) from each event in the dataset. It can be seen that most events are associated with either a small (1-3) or large (20-22) number of CDMs, while fewer events fall in the intermediate range. These insights are crucial for understanding how data density and the frequency of event observation impact model training and risk classification.

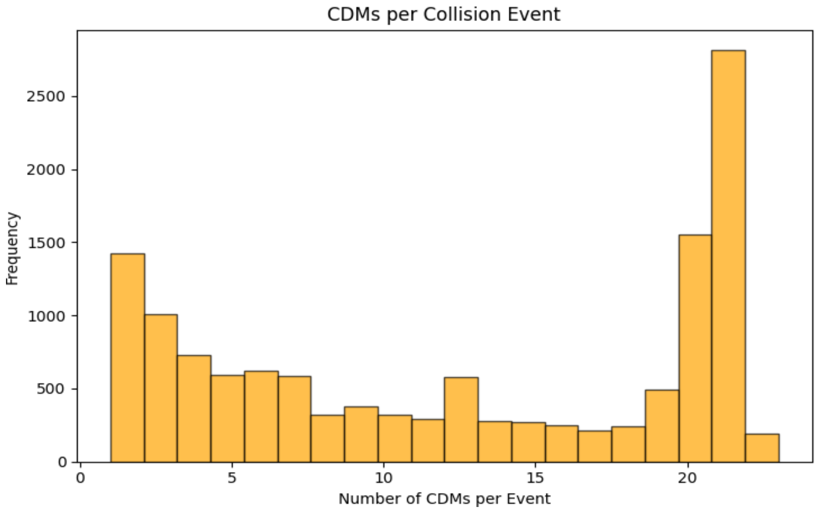

Figure 3: Distribution of the number of CDMs per collision event.

Figure 4 shows the distribution of objects. It can be seen that most of the data consists of space debris and unknown objects. This distribution shows that most collision risks originate from space debris.

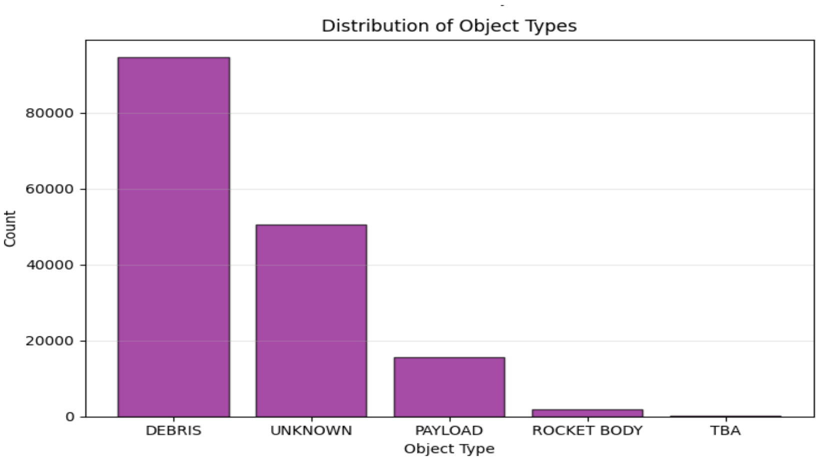

Figure 4: Distribution of object types in the dataset.

1.1 Data Preprocessing

Missing values in the data were handled using linear imputation to ensure that the records entered into the model were complete(pandas development team, 2024).Next, standardiza tion was performed on all variables using the StandardScaler tool from the scikit-learn library to reduce the impact of different measurement units and ensure the stability of the training process(scikit-learn developers, 2024b). The target variable, which is the risk value, was converted into a binary variable so that the event is classified as “high risk” if the risk value is greater than-6, and “low risk” if the value is less than or equal to-6. The data was randomly divided into training and testing sets at a ratio of 80% and 20%, with 130,107 samples for training and 32,527 samples for testing, respectively. This division was achieved while preserving class balance during the stratified split process (scikit-learn developers, 2024a).

1.2 Feature Selection

As mentioned earlier, the database contains a large number of features (103 features), but many of them may not be directly relevant to our study. Hence, it is crucial to apply feature selection techniques to reduce complexity, prevent overfitting, and, more importantly, enhance the model’s accuracy. The more the features are relevant to the study objective, the greater the accuracy and performance of the model Guyon and Elisseeff (2003). In the beginning, statistical correlation analysis was relied on as an input to understand the relationship between each attribute and the target variable as it is considered one of the basic statistical tools to understand the relationship between variables, and this helps us to understand the strength of the relationship between a particular attribute and the target variable. Thus it is a practical analysis to select the most influential attributes on the model. Two types of indicators have been applied to measure the relationship between variables, both of which are used within the framework of statistical correlation analysis. The first type is the Pearson correlation coefficient, which is used to measure the linear relationship between two numerical variables, and the distribution values of the data are between -1 and 1, where the value of 1 indicates a positive perfect linear relationship,-1 indicates a negative perfect linear relationship, and 0 indicates the absence of a linear relationship. The second type is Spearman Rank Correlation, which is used to measure the nonlinear monotonic relationship between variables and is more robust to outliers Pearson Johnson and Wichern (2007). In addition to the correlation indicators, the Mutual Information-MI scale is used to measure the extent of overlap or dependence between two variables, so if we want to know the extent to which the attribute X helps us in predicting the goal Y, MI tells us how much information does X provide about Y? If MI=0, it means that there is no relationship between the variable X and Y, and the higher the value of MI indicates that there is information in the attribute X that helps in predicting Y Intelligence (2023).Subsequently, more advanced machine learning tools were used to interpret the impact of features in greater depth. Among the most prominent of these tools is an advanced technique that is considered one of the most powerful interpretation tools in machine learning, known as SHAP Importance. This technique measures the individual impact of each attribute on the prediction resulting from the model, which helps in identifying the most influential attributes in the model’s output Lundberg and Lee (2017).

A technique has also been applied, Permutation Importance, which measures the amount of decrease in model performance by switching the values of each attribute to see how much the model’s performance decreases as a result of this switch, if the model’s performance is significantly affected after switching a particular attribute, this indicates that it is essential Learning and Group (nd). Due to the multiplicity of feature selection methods and their varying results, we applied the Ensemble Feature Selection technique, which is based on the hypothesis that combining the outputs of several models is better than relying on one model, this technique extracts the features that all the different algorithms that were previously applied agree on, and research has proven the importance of using this approach to reduce the reliance on the results of one algorithm, especially when there is a significant variation between features, and what sets it apart is its neutrality toward the model architecture, allowing it to be applied to different algorithms without bias Li et al. (2018). The details of the selected attributes and the results of each technique are described in the results section.

1.3 Deep Learning Algorithm

The deep learning approach was adopted in our study for several reasons, including the suitability of deep learning for complex data, and the data chosen for our study contained a sufficiently large number of events to ensure that the deep learning models functioned correctly. The data of each potential collision event is organized as a multivariate time series, with non-stationary intervals between points. In cases of this type, artificial neural networks, especially recurrent neural networks (RNNs), are among the most commonly used methods, as traditional methods for predicting time series are often limited to a single variable and assume a fixed time interval. The GRU algorithm, a type of recurrent neural network (RNN), was trained Chung et al. (2014).

1.3.1 Comparing models and selecting the most appropriate for time series data

Given the temporal nature of the data and the fact that it contains CDMs in sequential chronological order, some traditional models, such as Support Vector Machine (SVM) and Random Forest (RF), are unsuitable, as they do not take the chronological sequence of events into account. For example, the RF model treats each row independently without considering temporal order, ignoring the relationships between successive events, which contradicts the nature of temporal data STATWORX (nd). Similarly, the SVM model lacks an internal mechanism to capture long-term dependencies, making it unsuitable for sequence-based tasks Vapnik and Vashist (2006). Therefore, recurrent models such as the Gated Recurrent Unit (GRU) and the Long Short-Term Memory (LSTM) are crucial for achieving high performance in risk classification over time, due to their ability to handle sequential dependencies Chung et al. (2021). Although LSTM is one of the leading models for handling time series data, it was not the optimal choice in this study because it contains a larger number of operations, which can slow down training and increase the likelihood of overfitting. In contrast, GRU has a simpler structure because it uses only the hidden state to transfer information, eliminating the need for a separate memory cell. This reduces the number of operations and increases training efficiency while maintaining performance similar to that of LSTM Chung et al. (2021).

1.3.2 Model chosen for the study: GRU

GRU is a type of recurrent neural network (RNN) that has proven effective in processing sequential data such as speech, time series, and text. GRU emerged as a solution to the vanishing gradient problem in traditional neural networks, where models struggle to learn long-term relationships and lose information over time.

Due to these characteristics, GRU has received widespread attention in recent research. For example, Li et al. (2021) described GRU as highly efficient in computation compared to other models, making it suitable for time-sensitive prediction tasks. Another study Anony- mous (nd) also showed that GRU outperforms LSTM in training speed, a decisive factor in safety systems and risk classification where rapid execution is crucial.

Based on the above, GRU was adopted as the most effective and appropriate option for achieving this study's objective, as it combines accuracy, time efficiency, and a simplified structure that reduces the likelihood of overfitting. These are crucial factors in collision risk classification.

1.3.3 Model Architecture

The model architecture is thoughtfully designed to process time series efficiently, while preserving the temporal gradient of information and improving model performance. Before building the model layers, a Masking layer is commonly used when dealing with time series of varying lengths that have been processed to be uniform. The goal is to prevent the model from learning patterns from unreal data resulting from the values added to fill in the blanks. The layer here is used to tell the model to ignore any value equal to 0 in the time series Team (ndc). This is followed by building four consecutive layers (100 → 80 → 50 → 50 → 30 units) with a dropout mechanism gradually applied between layers to reduce the risk of overgen- eralization Team (ndb). The return sequences parameter was also used, and the first three layers were set to return sequences=True. This means that it returns a complete sequence of hidden states, as we need to retain the complete temporal information for each step in the early stages, which is required for the following layers. In contrast, for the fourth and last layer, it was set to return sequences=False, that is, we only take the final representation of the entire temporal sequence and directly transfer it to the Dense Layer, which in turn makes the final classification decision Huang (2018). The first layer was set to (100 units) to start capturing temporal dependencies from the data with Dropout=0.3, meaning that we randomly deleted 30% of the units during training to minimize the risk of overfitting in this large layer, as for the second layer (80 units), the number of units was reduced slightly to force the model to focus on the data more and Dropout=0. The third layer (50 units) was more focused on temporal features, and sequence rewinding was still ongoing in this layer to prepare the data for the fourth and final layer, Dropout was reduced to 0.1 as the number of units became smaller, the last layer was responsible for capturing the final rep- resentation of the sequence and Dropout was raised to 0.2 to support the model before the final classification layer.

1.3.4 Final Representation and Classification Mechanism

After processing the final representation, it is sent to the Dense layer, which is responsible for outputting the final classification decision. In this layer, the sigmoid function was used because it is suitable for classification tasks and is mathematically defined as $f(x) = \frac{1}{1+e^{-x}}$ where x is the resulting value from the previous layer and e^{-x} is the mathematical basis for the exponential function, which in turn helps to reduce large values, the function converts the resulting values to a range (0,1) to determine whether the case represents a risk (1) or not (0), according to our definition of the target variable Tuzsuz (nd)..

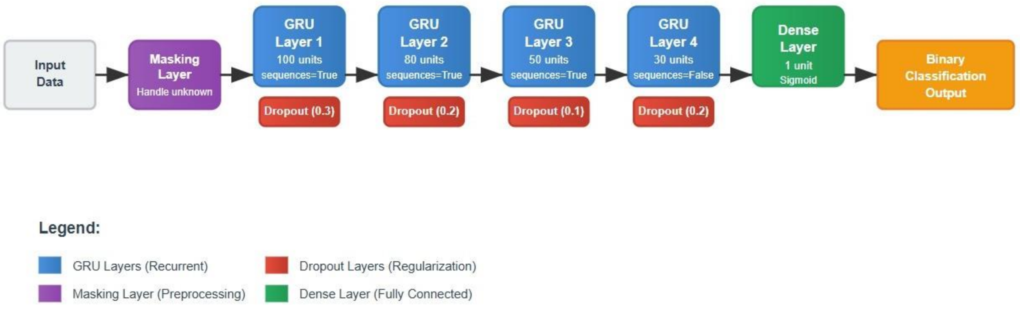

Figure 5: Architecture of the proposed model.

Figure 5 shows the architecture of the proposed model, starting from the input layer, through the GRU layers, to the final classification layer.

1.3.5 Training Settings and Model Evaluation

A type of loss function, the binary regression loss function, was used because it is suitable for binary classification tasks and is the best choice to minimize the loss between the actual value and the predicted value of the model Team (nda). To increase the model's efficiency, the Adam optimizer, a renowned optimization algorithm, is employed, which is particularly effective in handling large or noisy datasets Van Otten (2023). To reduce the issue of over- fitting, EarlyStopping is used when the observed metric (such as val loss) stops improving; after that, the best weights obtained during training are retrieved in order to obtain the best performance of the model TechwithJulles (2024). To evaluate the model's performance, three key metrics were used: Accuracy, Precision, Recall and ROC AUC. These metrics provide a comprehensive and accurate assessment of the performance of the model.

2. Results

2.1 Feature Selection Result

Several feature selection techniques were applied, including Pearson and Spearman correla- tions, Mutual Information, SHAP values, and Permutation Importance. While each method highlighted slightly different aspects, there was a strong convergence on a consistent subset of features.

Across methods, temporal features (e.g., time to tca), risk-related measures (max risk scaling, max risk estimate), and statistical descriptors (mahalanobis distance, c sigma t) appeared repeatedly as influential. This convergence indicates that both time proximity, risk indices, and distributional uncertainty play central roles in predicting collision risk.

To avoid redundancy, we report only the final ensemble fusion in the main text (Figure 6), which integrates all techniques and provides a stable and reliable set of top-ranked features. The detailed per-method plots are provided in Appendix 5 for reference.

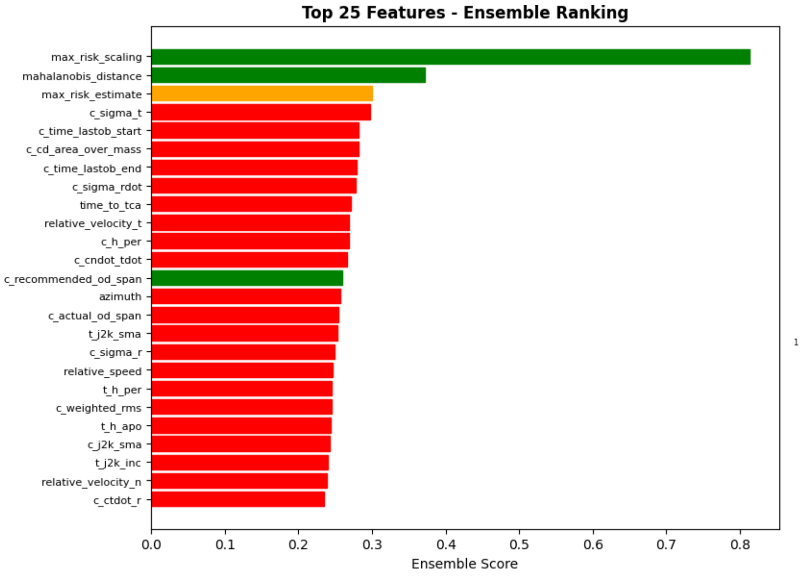

Figure 6: Top 25 features ranked by the degree of agreement across all feature selection techniques (ensemble fusion).

2.2 Model Result

The performance of the GRU classification model was evaluated using four metrics: **Accuracy, Loss, Precision, and Recall.**

Table 2 summarizes the results obtained from the training and validation sets.

The GRU classifier demonstrated strong overall performance across training, validation, and test sets. Accuracy remained consistently high (97%), with tight confidence intervals on the test set (95% CI: [0.9691, 0.9734]), confirming robust generalization. The low loss values (< 0.01) further indicate stable optimization and convergence.

The model achieved a balanced trade-off between recall and precision. Recall values (0.79–0.85) show that the classifier successfully detected the majority of high-risk cases, which is critical in collision risk prediction where missing a dangerous conjunction has severe consequences. Precision scores (0.74–0.81) reveal that some non-risky cases were misclassified as risky, implying that false alarms remain an operational challenge. This outcome highlights the safety-oriented bias of the model: it prioritizes minimizing false negatives, even at the cost of additional false positives.

Discrimination ability was further confirmed by the ROC-AUC (≈0.99) and PR-AUC (≈0.86), both of which indicate excellent separability despite class imbalance. The high ROC-AUC reflects strong overall classification capability, while the PR-AUC provides a more realistic view under skewed class distributions, demonstrating that the model maintains high recall without collapsing precision.

Overall, these findings validate the suitability of GRUs for sequential CDM data, offering reliable performance in collision risk prediction. Nonetheless, the observed precision–recall trade-off suggests that further work is needed to improve precision, for example by threshold tuning, class-weight adjustments, or ensemble learning to reduce false alarms while main- taining high recall.

Table 1: Summary Table of Model Performance

Metric	**Training (Best)**	**Validation (Best)**	**Test (Point)**	**95% CI (Test)**
Accuracy	0.9779	0.9711	0.9713	[0.9691, 0.9734]
Loss	0.0057	0.0068	–	–
Precision	0.8094	0.7457	0.7599	–
Recall	0.8465	0.8160	0.7912	–
ROC-AUC	0.9927	0.9880	0.9873	[0.9858, 0.9888]
PR-AUC	0.9223	0.8731	0.8649	[0.8492, 0.8786]

Table 2: Performance metrics of the GRU models.

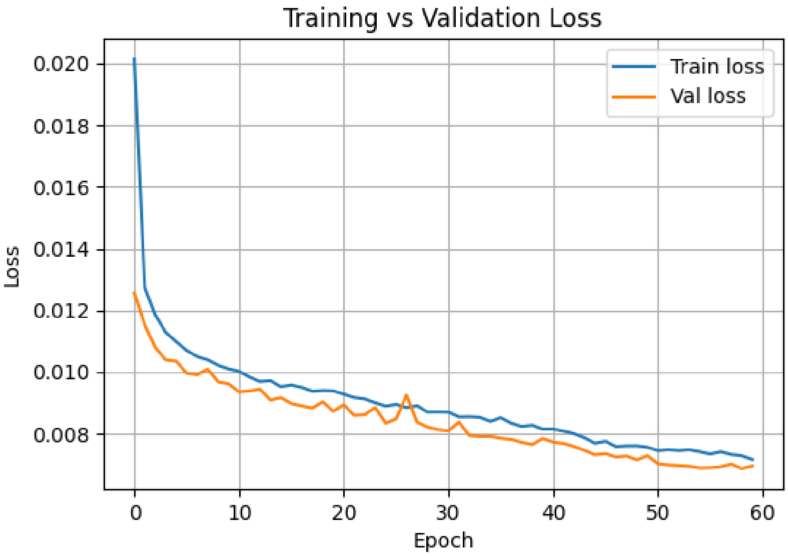

Figure 7: Training and validation loss curves across 60 epochs. Both curves decrease steadily and stabilize below 0.01, with validation loss closely tracking training loss. The small gap between the two indicates strong generalization and confirms that the GRU model converged stably without overfitting.

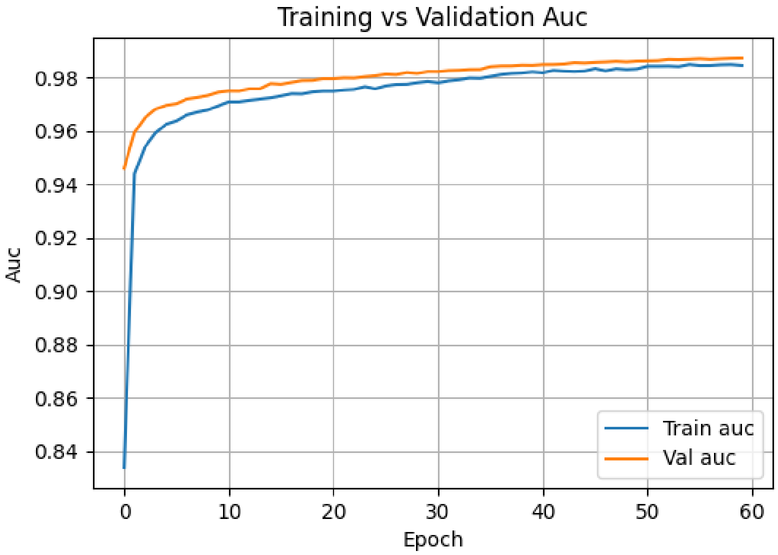

Figure 8: Training and validation AUC curves across 60 epochs. Both curves increase rapidly during early epochs and stabilize around 0.98. The close alignment between training and validation AUC, with validation slightly higher at several points, indicates strong general- ization and confirms the robustness of the GRU model without overfitting.

3. Discussion

To test and simulate the model for use in practical environments, a simple interactive in- terface was developed that allows users to enter the event ID.The system can also be pro- grammed to allow another input. In our study, the event ID was entered and the sys- tem classified the level of risk associated with the event. If the event is classified as high risk, the interface displays an automatic warning message, facilitating immediate decision- making and risk management.The model provides a strong foundation that will enable fu- ture researchers to develop more accurate interpretations of the results or utilize interpre- tive artificial intelligence.The interactive interface was utilized to create a clear picture of the application of machine learning in space agencies and to support informed decision- making. Previous research addressing the same objective as our study and utilizing the same data was unable to provide a clear picture or simulation for building a deep ma- chine learning approach to risk management. The source code and implementation details are available publicly on GitHub to encourage further development and collaboration: See: github.com/avmera/GRU-Neural-Networks

Acknowledgments

We would like to thank Rahab Al-Makrami, a physics student, for helping us understand the issue and the data, as we have no experience in the nature of astronomical data. She also helped us research the algorithms used to achieve the research goal. We would also like to thank Dr. Eman Al-Harbi from our department, who specializes in the field of artificial intelligence, for using her expertise to supervise and guide our research from the very beginning

4. Appendix

4.1 Detailed Feature Selection Results

For completeness, we include the full plots of the individual feature selection techniques. These complement the ensemble fusion result shown in the main text.

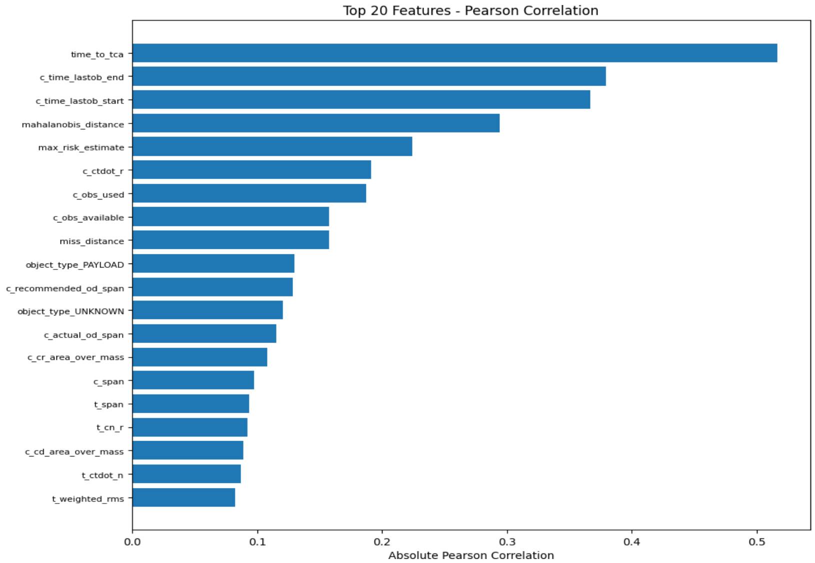

Figure 9: Top 20 features ranked by Pearson correlation with the target variable

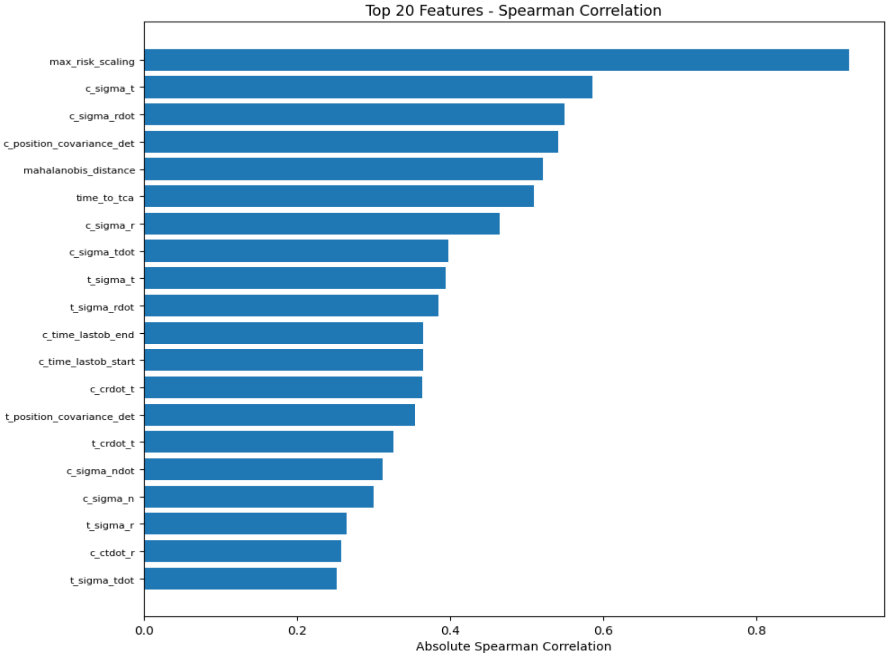

Figure 10: Top 20 features ranked by Spearman correlation with the target variable.

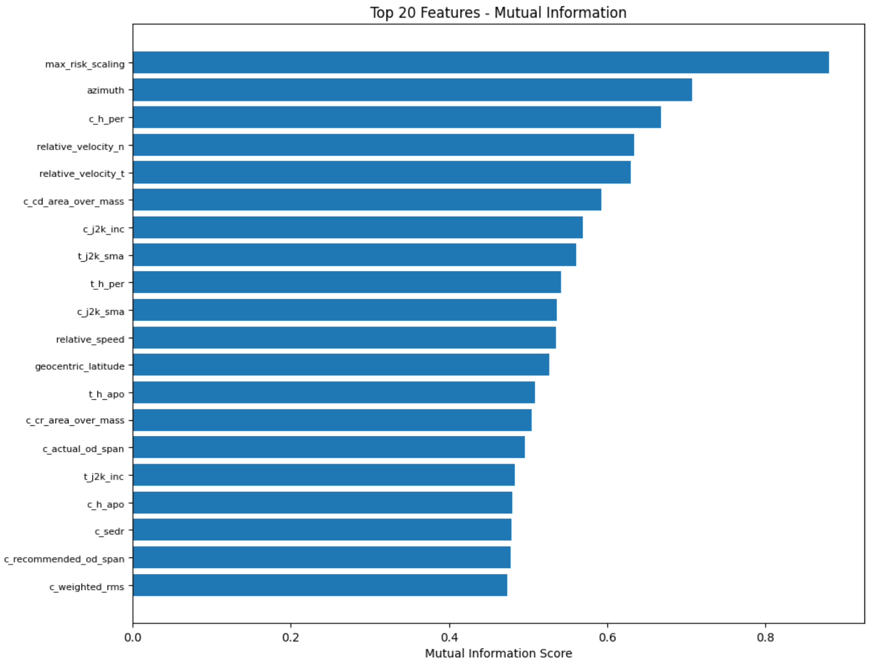

Figure 11: Top 20 features by Mutual Information with the target variable.

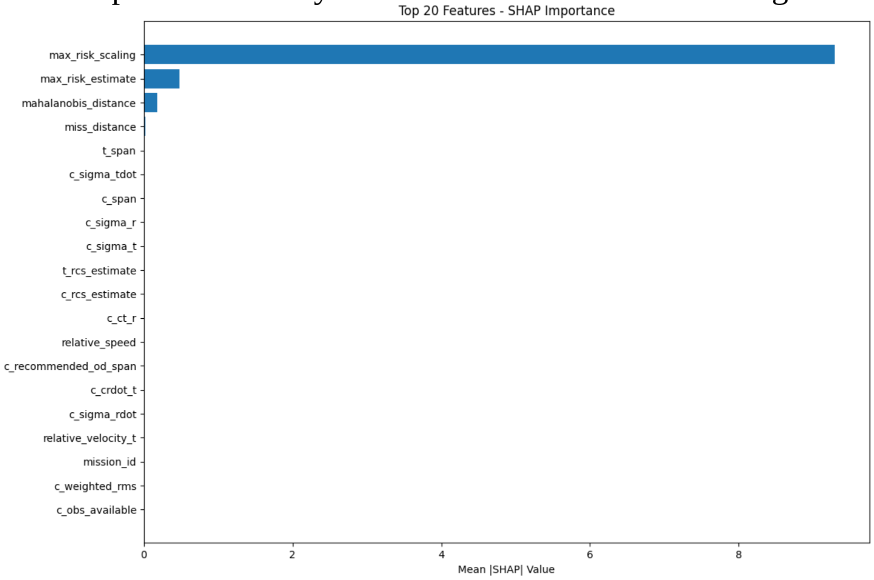

Figure 12: Top 20 features ranked by SHAP values.

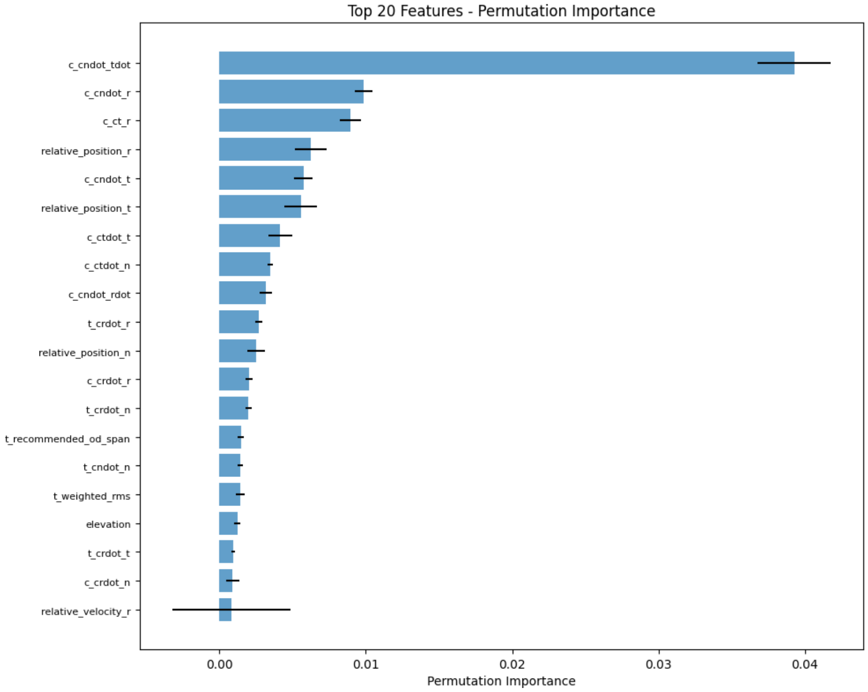

Figure 13: Top 20 features ranked by Permutation Importance.

References

Anonymous (n.d.). Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. Preprint / PDF reference.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint*, arXiv:1412.3555.

Chung, M., Park, S., and Lee, J. (2021). A review of recurrent neural network architecture for sequence learning: Comparison between lstm and gru. *ResearchGate*.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182. Retrieved July 25, 2025.

Huang, T. (2018). How to use return state or return sequences in keras. DLology.

Intelligence, S. (2023). Feature selection: An illustrated guide to selecting the right features for your model. Spot Intelligence. Retrieved July 19, 2025.

Johnson, R. A. and Wichern, D. W. (2007). *Applied Multivariate Statistical Analysis*. Pearson Education, 6th edition.

Kessler, D. J. and Cour-Palais, B. G. (1978). Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research: Space Physics*, 83(A6):2637– 2646.

Learning, S. and Group, D. S. (n.d.). Permutation feature importance. Interpretable Machine Learning –

Limitations and Extensions.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., and Liu, H. (2018). A survey on feature selection methods. *Information Fusion*, 50:115–144.

Li, Y., Zhang, H., Wang, J., and Chen, P. (2021). A comparative analysis of lstm, gru, and transformer models for construction cost prediction with multidimensional feature integration. *Information*, 12(11):442.

Lundberg, S. M. and Lee, S. I. (2017). A unified approach to interpreting model predictions.
In *Advances in Neural Information Processing Systems*, volume 30. n.d.

Office, N. O. D. P. (2019). It's always sunny in space. that's a problem for satellite teams.
NASA Earthdata.

pandas development team (2024). pandas.dataframe.interpolate. Accessed: 2025-07-30. scikit-learn

developers (2024a). sklearn.model$_s$*election.train test split. Accessed* :$_-$ 2025 −
07 − 30.

scikit-learn developers (2024b). sklearn.preprocessing.standardscaler. Accessed: 2025-07-30.

STATWORX (n.d.). Time series forecasting with random forest. STATWORX Blog. Retrieved
August 6, 2025.

Team, E. A. C. (2019). Kelvins collision avoidance challenge. European Space Agency. Team, K.

(n.d.a). Binarycrossentropy class. Keras. Retrieved July 23, 2025.

Team, K. (n.d.b). Keras documentation: Dropout layer. Keras. Retrieved July 25, 2025. Team, K.

(n.d.c). Masking layer. Keras. Retrieved July 20, 2025.

TechwithJulles (2024). Model validation and optimization — early stopping and model check- points. Medium. Retrieved July 23, 2025.

Tuzsuz, D. (n.d.). Sigmoid function – learndatasci. LearnDataSci. Retrieved July 29, 2025. Uriot, T.,

Izzo, D., Simões, L. F., Abay, R., Einecke, N., Rebhan, S., Martinez-Heras, J.,
Letizia, F., Siminski, J., and Merz, K. (2020). Spacecraft collision avoidance challenge: design and results of a machine learning competition. *arXiv preprint*, arXiv:2008.03069. [Submitted7Aug2020; last revised12Oct2020].

Van Otten, N. (2023). Adam optimizer explained & how to use in python [keras, pytorch & tensorflow]. Spot Intelligence. Retrieved July 20, 2025.

Vapnik, V. and Vashist, A. (2006). A new learning paradigm: Learning using privileged information. *arXiv preprint cs/0512062*.