Received: 2 July 2024 | Revised: 15 August 2024 | Accepted: 6 September 2024 | Published online: 12 December 2024

RESEARCH ARTICLE

Artificial Intelligence and Applications
2025, Vol. 3(1) 43–55
DOI: 10.47852/bonviewAIA42023741

BON VIEW PUBLISHING

Enhanced Space Debris Detection and Monitoring Using a Hybrid Bi-LSTM-CNN and Bayesian Optimization

Ishaani Priyadarshini[1,*]

[1]*School of Information, University of California, USA*

Abstract: Space debris detection is important to the integrity of space missions and satellites, especially with the increase in the number of satellites and spacecraft in orbit. This paper addresses this by a new concept innovative approach using hybrid Bi-LSTM-CNN architecture which is optimized using Bayesian optimization. The study presents an analysis approach based on the whole combination of machine learning and deep learning, a high-quality space debris detector, that can identify both the kind of object and the size of its RCS. The new method goes beyond what we have done so far and shows better results on a wide range of evaluation parameters, such as accuracy, precision, memory, and $F1$ score. Also, the study takes up the pragmatic issue of training time, thereby ensuring performance in real time. Esthetic trials on real datasets confirm the fit of the hybrid model, sensitivity, and efficacy with 99.16% and 99.98% detection and prediction of space debris types, respectively. In summary, this paper makes space debris tracking much more robust and mitigates threats associated with spaceflight and satellite operations, but they can offer a lot of information on threats and mitigation measures. The findings suggest that this hybrid model could be augmented with current space debris tracking systems, to increase their predictive power and operational effectiveness.

Keywords: Bayesian optimization, deep learning, monitoring system, space debris, Bi-LSTM-CNN

1. Introduction

Space debris—commonly known as space junk—refers to non-functional, human-made objects that orbit the Earth and are no longer operational. This classification includes a diverse array of items, such as discarded rocket stages, inactive satellites, and fragments that result from collisions or explosions in the space environment [1]. Because the number of satellites and spacecraft being launched into orbit continues to increase, the likelihood of collisions with this debris also rises. This situation presents significant risks to space missions, the functionality of satellites, and even human safety. Therefore, monitoring space debris is essential for mitigating these hazards and ensuring the long-term viability of space endeavors. The primary aim of such monitoring is to accurately track the position, trajectory, and properties of debris objects in Earth's orbit. These data empower space agencies, satellite operators, and other stakeholders to evaluate collision risks and implement necessary strategies to avert potential impacts. However, the complexity of the space environment makes this task particularly challenging, although technological advancements are continuously being made to address these issues. Such strategies may involve altering satellite orbits, executing collision avoidance maneuvers, or retiring satellites at the conclusion of their operational lifespan to prevent them from contributing further to the debris problem [2, 3]. The presence of space debris poses a significant threat to space infrastructure, which includes active satellites, space stations, and crewed spacecraft. Collisions involving debris may lead to significant damage, resulting in the loss of crucial assets and the interruption of vital services (such as communication, navigation, and Earth observation). Moreover, the buildup of debris in certain orbital zones—especially low Earth orbit (LEO)—increases the probability of cascading collisions; this phenomenon is referred to as Kessler syndrome. In such cases, one collision can trigger a sequence of further collisions, creating a dense cloud of debris that makes the impacted orbital area unviable for future space operations. However, efforts to mitigate this issue are ongoing, because the consequences could be dire for both current and upcoming missions. The monitoring of space debris has become increasingly vital for ensuring the safety and sustainability of space operations, positioning it as a key focus area for research and development in artificial intelligence (AI) aimed at social good. Researchers are employing AI methodologies, including machine learning (ML) and deep learning (DL), to improve the precision and effectiveness of space debris monitoring systems [4]. Recent developments in AI have revealed considerable promise in addressing intricate challenges, such as those associated with space debris surveillance. Current research trends highlight the application of AI-based strategies, including hybrid models and optimization algorithms, to process extensive datasets obtained from ground-based sensors, telescopes, and spaceborne instruments. These innovative techniques (which have emerged recently) demonstrate enhanced capabilities in detecting, classifying, and tracking debris; this development effectively addresses the shortcomings of traditional monitoring systems.

*Corresponding author: Ishaani Priyadarshini, School of Information, University of California, USA. Email: ishaani@berkeley.edu

© The Author(s) 2024. Published by BON VIEW PUBLISHING PTE. LTD. This is an open access article under the CC BY License (https://creativecommons.org/licenses/by/4.0/).

Moreover, AI-enhanced monitoring systems facilitate the efficient analysis of substantial amounts of observational data, allowing for more effective identification and tracking of debris objects. Additionally, these AI-driven systems improve operational efficiency and encourage global collaboration and accountability in space exploration. By delivering real-time risk assessments and predictive analytics, they enable satellite operators and space agencies to proactively manage collision risks, thereby reducing the likelihood of catastrophic incidents and safeguarding essential space assets. However, this forward-thinking strategy promotes international collaboration and fortifies frameworks for space governance (thus advocating for the peaceful and sustainable utilization of outer space) for the benefit of all humanity. Although the integration of AI in monitoring space debris significantly enhances space situational awareness, it also improves space traffic management, reflecting a crucial advancement in this field. This advancement aligns with the overarching objective of promoting long-term sustainability and peaceful coexistence among humanity in outer space. However, it also illustrates the transformative capacity of AI for societal benefit. Although there are challenges, this progress is essential because it paves the way for future endeavors. But we must remain vigilant, as the implications of such technology are profound [5]. The ability to predict the size of the Radar cross-section (RCS) plays a crucial role in identifying space debris by offering valuable insights into its physical attributes and the potential threats it may pose. Generally, a larger RCS suggests a greater physical size of the debris, which can increase the risk of collisions with operational spacecraft or satellites. By accurately forecasting RCS size, space agencies and satellite operators can more effectively evaluate collision risks, allowing for the implementation of suitable mitigation strategies, such as adjusting satellite trajectories to avoid potential impacts or organizing debris removal initiatives. Moreover, a thorough comprehension of RCS size distribution (1) can inform spacecraft design; this is especially true regarding the integration of advanced shielding to mitigate the impact of potential collisions with larger debris. The effective management of substantial challenges related to space debris monitoring requires the adoption of innovative strategies, which leverage the capabilities of AI and optimization algorithms. In this regard, a hybrid model referred to as the Bi-LSTM-CNN architecture, enhanced through Bayesian optimization, has been introduced. This model signifies a significant advancement in the realm of space debris monitoring, as it amalgamates the strengths of Long Short-Term Memory (LSTM) networks with convolutional neural networks (CNNs) and Bayesian optimization techniques. Although the LSTM layers excel at capturing temporal dependencies, they are particularly adept at handling the complexities inherent in sequential datasets. Within the domain of space debris monitoring, this sequential data can encompass time-series information acquired from a variety of sensors (including radar and infrared devices), which measure parameters like velocity, position, and other critical characteristics of debris in orbit. The LSTM component of this architecture is capable of analyzing these sequential data streams to discern patterns, trends, and anomalies that might indicate potential collision threats. Conversely, the CNN layers excel in extracting spatial features from multidimensional datasets. Although primarily utilized for image processing, CNNs can also be effectively employed on numerical data that possesses a spatial structure—such as grids or matrices that illustrate spatial relationships. However, the integration of these techniques remains complex, because each method has its own intricacies and challenges, yet the potential for improved monitoring is significant.

In space debris monitoring, this could involve data representing the spatial distribution of debris objects in different orbital regions or the layout of sensor networks for monitoring. By integrating LSTM and CNN layers into the architecture, our proposed Bi-LSTM-CNN model can effectively analyze the multidimensional numeric data encountered in space debris monitoring. Furthermore, incorporating Bayesian optimization allows automatic tuning of hyperparameters, resulting in improved model performance and efficiency. Bayesian optimization is particularly well-suited for optimizing complex, high-dimensional search spaces, making it an ideal choice for fine-tuning the parameters of the Bi-LSTM-CNN architecture [6, 7]. One of the key highlights of our approach lies in integrating a diverse set of ML and DL algorithms to assess the performance of the proposed model. Therefore, the approach is twofold: a classification is conducted to identify the type of debris and simultaneously classify the size of RCS using the same dataset. The model's capabilities across different data types and modalities have been validated by leveraging ML algorithms and DL architectures. To evaluate the performance of the Bi-LSTM-CNN model, a range of metrics, including accuracy, precision, recall, and $F1$ score, has been employed. These metrics comprehensively assess the model's ability to detect and classify space debris objects accurately. Additionally, training time has been considered a critical factor, ensuring the model is efficient and scalable for real-world deployment.

The rest of the paper is organized as follows. Section 2 encompasses materials and methods wherein some related works, the methodology, and the proposed architecture have been discussed. The main emphasis is given to the Bi-LSTM-CNN technique and the Bayesian optimization. Section 3 highlights the overall results and observations based on the dataset and experimental analysis. A comparative analysis depicts how the proposed work contributes to the space debris monitoring research, followed by overall observations. Section 4 concludes the study.

2. Materials and Methods

In this section, there is an emphasis on some of the recent related works concerning space debris monitoring. The study also highlights the limitations of the existing approaches for identifying space debris. This is followed by a methodology section focusing on the proposed architecture and its components. Consequently, the algorithm and the pseudocode have been analyzed.

2.1. Related works

A study by Jordan et al. [8] introduced a method based on Particle Swarm Optimization (PSO) to estimate the inertia parameters of uncooperative satellites involved in space debris removal. This methodology emphasizes the estimation of the inertia of a rotating target within a torque-free environment, utilizing quaternion data derived from attitude observations. The authors derive the symmetric inertia tensor of the target by conceptualizing the PSO solution space as a multidimensional vector that corresponds to the inertia tensor. To generate estimated measurements, they apply Euler's equations to propagate the attitude motion, achieving validation against experimental data with an error margin of less than 1%. Despite its merits, the approach is limited by the assumption of a torque-free environment (a condition that may not accurately depict the realities faced during debris removal operations). It relies solely on experimental measurements for validation; however, this neglects potential inaccuracies in those measurements. Further validation across various conditions is crucial to assess the robustness of this method. In a separate study, Ryan et al. [9] explored the application of ML techniques to enhance spacecraft protection against impacts from micrometeoroids and orbital debris. Their research highlights the capability of ML to tackle complexities that surpass those handled by traditional semi-empirical models. By utilizing artificial neural networks (ANNs), support vector machines (SVMs), and extreme gradient boosting (XGBoost), the authors illustrate the effectiveness of ML in this domain, with XGBoost emerging as the most effective model. However, the study's limitations are evident, as it primarily relies on fundamental ML techniques. Future research, employing advanced ML methods and comprehensive datasets, could yield deeper insights into spacecraft risk assessment and protection strategies. Researchers [10] introduced an unsupervised learning approach—DBSCAN—to identify clusters of orbital debris. This method utilizes proper element data obtained from two-line element (TLE) sets; however, its effectiveness is contingent upon the quality of the input data. Although promising, the results must be interpreted cautiously because they represent only a portion of the broader context in space safety. Proper elements for debris fragments in LEO are computed using a numerical scheme similar to the Fourier-series-based method for asteroids. To enhance the classical DBSCAN's heuristic nature, neural networks trained on known families are explored. However, it is important to acknowledge the study's limitations, including potential challenges in accurately representing complex orbital debris distributions with neural networks. Another study [11] conducted an analysis of active debris removal (ADR) mission planning, aiming to generate optimal debris removal plans. They established a two-layer time-dependent traveling salesman problem mathematical model to address debris removal sequence and transfer trajectory planning. Novel ML-based methods were proposed for ADR mission planning, including a deep neural network (DNN)-based estimation method for optimal velocity increments and a reinforcement learning-based method for optimizing debris removal sequence and rendezvous time. Simulation results demonstrate higher estimation accuracy compared to analytical methods. However, the proposed methods may face challenges in scaling to more complex mission scenarios and require extensive computational resources for training neural networks in real-time applications. Some researchers [12] introduced a ML-driven regression method for estimating the ballistic coefficient in LEO, covering a broad range of orbital parameters. They evaluated various ML techniques using synthetic space catalog data and conducted sensitivity analyses on training size and measurement frequency factors. Despite the neural network achieving an 84% success rate, challenges arise in extrapolating the approach to real-world scenarios due to potential biases in the synthetic dataset and uncertainties in real data. The effectiveness of the method may be constrained by its dependence on precise and extensive training data, particularly in the context of dynamic and changing space environments. A separate research team [13] developed a DL model aimed at predicting the re-entry of uncontrolled objects in Low LEO, utilizing a modified Sequence-to-Sequence architecture. This model was trained on average altitude profiles derived from TLE data of more than 400 objects and incorporated innovative input features such as a drag-like coefficient (B^*), average solar index, and area-to-mass ratio. Performance evaluations conducted on objects from the Inter-Agency Space Debris Coordination Committee campaign indicated that the model performed optimally for objects exhibiting similar drag-like coefficients and eccentricity distributions as those in the training dataset. However, its effectiveness diminished when applied to objects with markedly different characteristics, underscoring the challenges associated with generalizing findings to a broader range of space debris. In addition, a recent study conducted by Guo et al. [14] introduced a novel methodology for clustering spectral polarization data derived from space debris. This approach employs a hybrid fuzzy C-means (FCM) algorithm, which integrates hierarchical agglomerative clustering (HAC). The validation of this algorithm's effectiveness was based on the Kosko subset measure formula and characteristic parameters acquired from laboratory tests were utilized to develop a clustering matrix. The parameters for the algorithm were determined by randomly selecting points within the external field. However, although this method achieved an impressive classification accuracy of 96.92% across six sample types in spectral polarization images, its applicability to more diverse or complex debris datasets may be constrained. Additionally, the reliance on laboratory test data may not adequately reflect the variability found in actual debris environments, which could adversely affect the algorithm's performance in real-world scenarios. Another research group [15] introduced a Physics Informed Neural Network (PINN) approach for estimating space debris trajectory post-collision events with active satellites. The simulation involved 8565 inelastic collisions using TLE data for 1647 Starlink and 66 LEO Multi-Use Receiver satellites. Despite comprehensive simulation and proposed velocity sampling methods, the classical optimization method, the Lagrange multiplier approach, yielded unsatisfactory state estimation due to under-determination. Alternative DNN and PINN-based methods were developed, with PINN-based approaches demonstrating superior performance in estimating position, velocity, mass, and coefficient of restitution of space debris. However, limitations may arise when attempting to generalize findings to intricate and rapidly changing collision situations that are not adequately captured in the simulation data. Furthermore, the effectiveness of the suggested methodologies may fluctuate based on the accessibility and quality of input data derived from actual collision occurrences. In their review, researchers [16] examined DL techniques for identifying space targets and their components, highlighting their significance in enhancing space missions. Although the successful detection and identification of space targets through electro-optical sensors are vital for the management of spacecraft, current studies reveal certain limitations. The review systematically covers the principles and characteristics of these sensors and common synthetic methods for space target datasets. Despite summarizing recent research and addressing major issues in space target detection and segmentation, the applicability of DL methods may be constrained by challenges such as limited training data and variable environmental conditions in space. Another study [17] explored the application of a commercial global flash Light Detection and Ranging (LiDAR) sensor in ADR operations, emphasizing the need for precise target positioning and orientation. While relative navigation devices like cameras or LiDAR sensors are commonly used for such missions, this study simulated data acquisition and processing from a commercial LiDAR sensor. The novelty lies in using multilayer perceptron neural networks to process LiDAR depth images for estimating the target's pose. However, limitations may arise from the complexity of accurately modeling real-world debris environments and the potential variability in LiDAR sensor performance under different conditions.

Historically, a variety of techniques have been employed to monitor and identify space debris. The literature delves into numerous methodologies for monitoring and removing this debris. These approaches include sophisticated algorithms, such as PSO and ML techniques that aim to estimate debris, assess impact risks, and facilitate mission planning. Research emphasizes the use of neural networks and DL frameworks to enhance the accuracy of debris classification and trajectory prediction. However, although these methodologies yield promising results, they encounter persistent challenges, such as difficulties in managing complex debris distributions, scalability to real-world applications, and the need for comprehensive validation. This collective research underscores the urgent necessity for integrating advanced techniques while addressing practical constraints to enhance space debris management. Nevertheless, the potential of AI—particularly in ML and DL—remains significant significantly underutilized (in various contexts). By harnessing advanced ML and DL frameworks, this study seeks to rectify the deficiencies of prior methodologies: consequently, it advocates for the implementation of the Bi-LSTM-CNN model. Optimized through Bayesian techniques, this model is anticipated to provide robust solutions; however, there may be challenges to overcome. Although the potential exists, the study will also address the limitations identified in earlier research, because it is essential for progress in this field.

2.2. Methodology

The initial phase involves a thorough preparation of data intended to accurately classify space debris and RCS sizes through various ML methodologies. This stage encompasses the collection and preprocessing of the dataset; it is crucial for ensuring completeness and consistency by addressing issues such as missing values, outliers, and discrepancies. However, feature engineering is conducted meticulously to extract relevant information. Following this, a selection process is implemented to optimize model efficacy. The dataset is then divided into training and testing subsets, with careful attention to maintaining balanced class distributions to mitigate bias. Although this process is complex, it is essential for achieving reliable results. Once the data are adequately prepared, a variety of ML algorithms, along with DL and ensemble methods, are implemented. Following this deployment, the models' performance is rigorously assessed. Each algorithm is trained on the training dataset and subsequently evaluated on the test dataset using a range of performance metrics, including accuracy, precision, recall, and $F1$-score. Notably, the proposed methodology incorporates a Bi-LSTM-CNN framework enhanced by Bayesian optimization, which synergizes the advantages of DL with optimization strategies, thereby delivering robust and efficient classification outcomes. Figure 1 depicts the overall methodology of the process.

1) Bi-LSTM-CNN architecture

This study presents a hybrid methodology that amalgamates Bi-LSTM and CNN techniques for the classification of space debris and RCS sizes. The architecture seamlessly incorporates Bi-LSTM and CNN components, culminating in a definitive output layer. The Bi-LSTM layer processes input sequences in both forward and backward directions; this enables the model to effectively capture temporal dependencies from past and future time steps. Each LSTM unit is endowed with a memory cell that retains information over time, thereby preserving long-term dependencies. The LSTM units employ gate mechanisms—such as input, forget, and output gates—to regulate the flow of information throughout the network. However, some challenges remain in optimizing these interactions, although the potential for enhanced classification accuracy is significant, because the integration of these techniques can yield promising results, which enhances gradient flow and mitigates issues related to vanishing or exploding gradients. The hidden state of each LSTM unit reflects the current state of the sequence, which is then transmitted to subsequent time steps and layers for additional processing. Meanwhile, the CNN layer employs convolutional filters on the input numerical data to identify local patterns or features. These filters traverse the input sequence, executing element-wise multiplications and summations to generate feature maps. Following the convolution process, an activation function is applied to introduce non-linearity to the extracted features, thereby improving the model's ability to capture intricate patterns. Max Pooling can down sample the feature maps, reducing computational complexity and spatial dimensions while retaining important features. The output of the CNN layer serves as input to

Figure 1
Space debris classification methodology

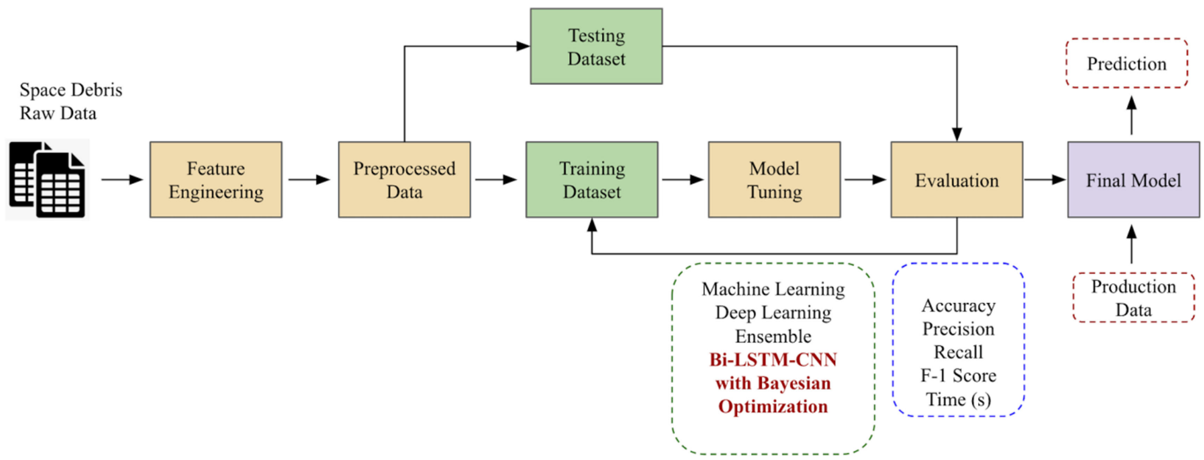
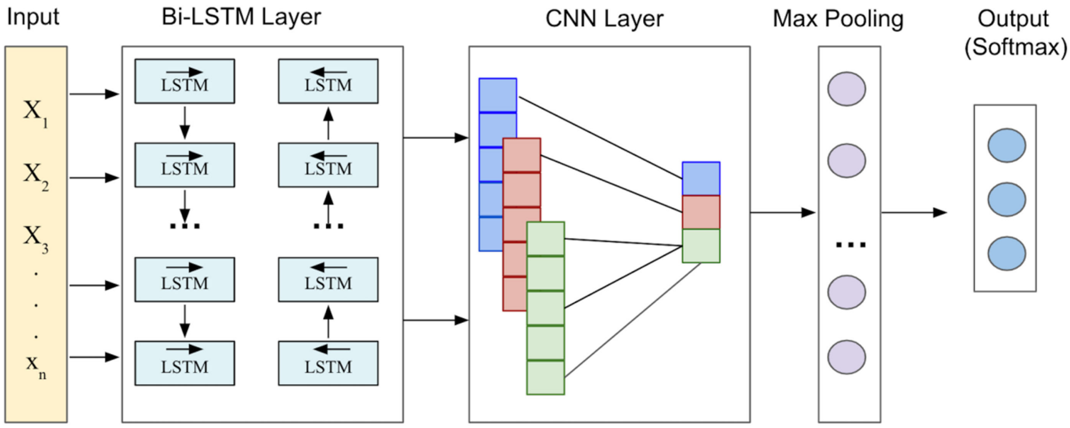
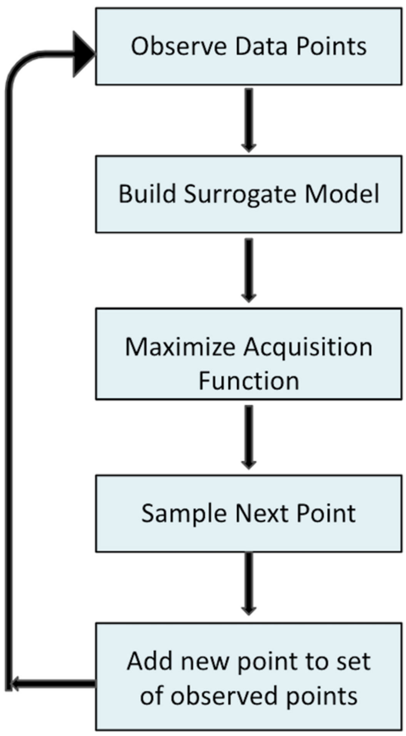


Figure 2
Proposed Bi-LSTM-CNN architecture



the Bi-LSTM layer, providing it with spatial features extracted from the numeric data. These features capture spatial patterns and relationships within the input sequence. The Bi-LSTM layer processes the spatial features temporally, capturing sequential patterns and dependencies within the numeric sequences over time. This integration allows the model to learn both spatial and temporal representations simultaneously. The output layer typically consists of one or more fully connected layers, which map the learned features to the desired output format. An activation function (softmax) is applied to the output of the fully connected layers to produce the final predictions or outputs. During training, a loss function measures the disparity between predicted and true labels/values, guiding the optimization process. Figure 2 depicts the overall architecture of the Bi-LSTM-CNN model.

Figure 3
Bayesian optimization steps



2) Bayesian optimization

Bayesian optimization begins with an initial set of hyperparameters for the model. These hyperparameters define the model's configuration, such as learning rate, batch size, dropout rate, etc. Bayesian optimization relies on a surrogate probabilistic model (often a Gaussian Process) to approximate the objective function, which in our case is the performance metric of the Bi-LSTM-CNN model (e.g., accuracy, $F1$-score). The surrogate model estimates the model's performance for different combinations of hyperparameters based on the evaluations conducted so far. The acquisition function guides the selection of the next set of hyperparameters to evaluate. It balances exploration (trying new hyperparameter configurations) and exploitation (evaluating promising configurations). Common acquisition functions include Expected Improvement, Probability of Improvement, and Upper Confidence Bound. The selected set of hyperparameters is used to train and evaluate the Bi-LSTM-CNN model on a subset of the training data (validation set). The model's performance metric (e.g., accuracy) is computed based on its performance on the validation set. The performance metric obtained from evaluating the objective function updates the surrogate model. The surrogate model is refined based on the new data point to better approximate the objective function. Building a surrogate function and acquisition function are repeated iteratively for a predefined number of iterations or until convergence criteria are met. At each iteration, the acquisition function is used to select the next set of hyperparameters to evaluate, and the process continues until the optimal set of hyperparameters is found. Figure 3 shows the steps involved in Bayesian optimization.

2.3. Algorithmic analysis and pseudocode

In this section, the overall algorithm of the classification process has been analyzed along with the pseudocode. The following algorithm depicts the pseudocode for deploying Bi-LSTM-CNN with Bayesian optimization for identifying Space debris (Figure 4).

The algorithm described here elucidates the methodology for optimizing the hyperparameters of a Bi-LSTM-CNN model through the application of Bayesian optimization [18]. The process begins with the initialization of Bayesian optimization;

Figure 4
Algorithm for Bi-LSTM-CNN with Bayesian optimization

```
Algorithm 1: Bayesian Optimization for Hyperparameter Tuning of
Bi-LSTM-CNN Model
Input  : Max iterations (max_iterations)
Output: Optimal hyperparameters, Final model performance
initialize_bayesian_optimization();
define_search_space();
define_objective_function();
for iteration = 1 to max_iterations do
    next_hyperparameters ← select_next_hyperparameters();
    train_model_with_hyperparameters(next_hyperparameters);
    model_performance ← evaluate_model_performance();
    update_bayesian_optimization(next_hyperparameters, model_performance);

end
optimal_hyperparameters ← get_optimal_hyperparameters();
train_final_model(optimal_hyperparameters);
final_model_performance ← evaluate_final_model_performance();
output_results(final_model_performance);
```

however, it is subsequently followed by the establishment of the hyperparameter search space. An objective function evaluates the model's performance (based on predetermined metrics). The algorithm advances through a specific number of iterations, during which it selects hyperparameters, trains the model, assesses its performance, and updates the Bayesian optimization framework as necessary. The optimal hyperparameters are identified upon the conclusion of the iterations and the final model is trained to employ these parameters. Following this, the performance of the trained model is evaluated and the results are reported. This methodology streamlines the hyperparameter tuning process, facilitating an efficient exploration of the hyperparameter space to identify configurations that enhance the model's performance. Although the overall process can be articulated in the following steps, it is important to note that these steps are not strictly linear.

Step 1: Initializing the Bayesian optimization algorithm.
Step 2: Defining the range of hyperparameters to be optimized.
Step 3: Defining the performance metric (e.g., accuracy, $F1$-score) used to evaluate the model.
Step 4: Bayesian Optimization Loop

1) Using the Bayesian optimization algorithm to select the next set of hyperparameters for evaluation.
2) Training the Bi-LSTM-CNN model using the selected hyperparameters.
3) Evaluating the model's performance on a validation set using the defined objective function.
4) Updating the Bayesian optimization algorithm with the hyperparameters and corresponding model performance.

Step 5: Once the optimization loop is complete, obtaining the optimal set of hyperparameters.
Step 6: Training the final Bi-LSTM-CNN model using the optimal hyperparameters obtained from Bayesian optimization.
Step 7: Evaluate the performance of the final model on a separate test set.
Step 8: Output the final model performance and any other relevant results.

3. Results and Observations

This section examines the dataset utilized in the study as well as the criteria employed to assess the performance of the model. Subsequently, a comprehensive analysis is conducted employing a variety of ML and DL algorithms. The overall results are then compared with findings from prior research to draw meaningful conclusions.

3.1. Dataset

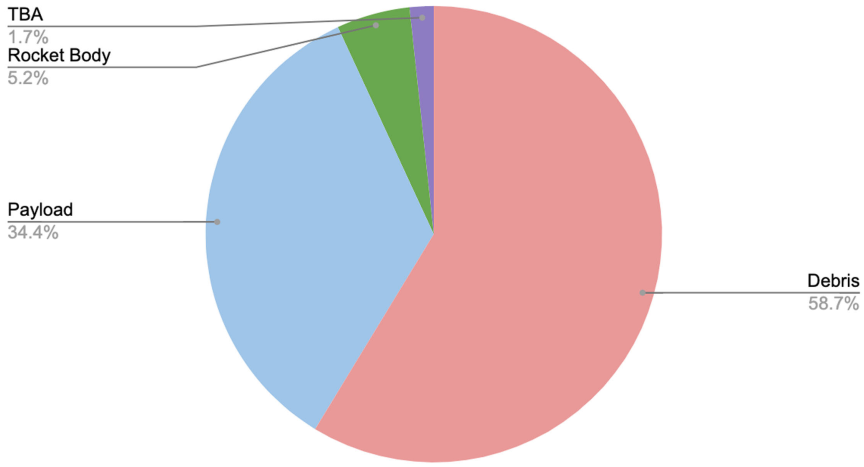
The dataset "Satellites and Debris in Earth's Orbit", sourced from Kaggle, was obtained through an API provided by space-track.org. Comprising 14,372 rows and 40 columns, it encompasses attributes such as object name, ID, country code, and launch date. The target variable, object type, categorizes entities into payload, debris, rocket body, and TBA (to be announced). The data will be partitioned into 80% training and 20% test datasets for our experimental analysis. A significant imbalance in the dataset was noticed during the initial exploratory analysis. Figure 5 illustrates the overall distribution of the data. Notably, 58.7% of the data is labeled as debris, while 34.4% corresponds to the payload. Rocket body data accounts for 5.2%, with the remaining unnamed data. To mitigate bias, the study applies sampling techniques to balance the data before conducting a comprehensive analysis. Similarly, the RCS size can be classified as small, large, and medium.

3.2. Evaluation parameters

The subsequent evaluation criteria (parameters) have been taken into account for assessing the performance of the model.

1) Accuracy: This metric quantifies the ratio of instances that are correctly classified to the total number of instances, thus offering a thorough overview of the model's efficacy across various categories. A greater accuracy indicates that the model is making meaningful progress; however, it is essential to consider other factors as well. Although higher accuracy is desirable, it does not always reflect the model's true performance because nuances in data can lead to misleading interpretations fewer erroneous predictions.

Figure 5
Data distribution for object type



2) Precision: Precision is a metric that evaluates the ratio of true positive predictions (those accurately identified as positive) to all instances deemed positive. This measure is particularly important (because) the consequences of false positives can be quite significant. Elevated precision suggests that the model is adept at minimizing the occurrences of false positives.
3) Recall: Recall, often called sensitivity or the true positive rate, examines the ratio of true positive predictions to all actual positive instances. This metric reflects the model's ability to recognize all relevant instances and is crucial when the stakes of false negatives are high. A high recall implies that the model effectively identifies positive instances.
4) *F*-1 Score: The *F*-1 score, on the other hand, represents the harmonic mean of precision and recall, generating a unified metric that harmonizes both components. This score is especially useful in contexts with imbalanced class distributions or when the costs of false positives and negatives vary. A higher *F*-1 score typically indicates enhanced overall performance, effectively balancing precision and recall. However, it is essential to consider these metrics collectively, as each serves its purpose depending on the specific context and requirements of the analysis recall.
5) Time to Run: The time to run (in seconds) is an important parameter: this indicates the duration necessary for the model to train on the given dataset and produce predictions. It is crucial to consider this aspect—especially for extensive datasets or intricate models—because it significantly influences the model's scalability and efficiency. Striving to reduce the time to run is often advantageous; however, it is essential to maintain a balance, although achieving this can be challenging with satisfactory performance metrics.

3.3. Experimental analysis

A diverse set of algorithms, including classical ML techniques, DL techniques, and ensembles, has been considered for the experimental analysis. The study deploys logistic regression (LR), K-Nearest Neighbors (KNN), Gaussian Naive Bayes (GNB), Decision Trees (DT), Random Forests (RF), Gradient Boosting (GB), XGBoost, Light Gradient Boosting (LGB), SVM, ANN, Feed Forward Neural Networks (FFNN), LSTM, CNN, and CNN-LSTM models on the dataset for classification. The performance of the proposed method, bi-LSTM-CNN with Bayesian optimization, is compared against the diverse set of algorithms using evaluation metrics. To ensure a fair comparison across all models, each algorithm was trained with a consistent number of epochs, set to 100. This uniform training duration was chosen to standardize the experimental conditions, allowing for an equitable evaluation of model performance. The consistency in epochs helps in isolating the effects of algorithmic differences from those arising due to variations in training duration. The first phase of our study involves deploying classification models for identifying object types in the dataset, while the second phase involves identifying the RCS size (small, medium, large). The use of 100 epochs provides sufficient training time for each model to converge and demonstrate its performance characteristics effectively.

Moreover, the hyperparameters for all models are specified as follows:

1) LR—Regularization strength $C = 1.0$ C = 1.0, solver = 'liblinear'.
2) KNN—Number of neighbors $\kappa = 5$ k = 5, distance metric = 'minkowski'.
3) GNB —Default parameters. DT: Maximum depth = 10, criterion = 'gini'.
4) RF—Number of trees = 100, maximum depth = 10.
5) GB—Learning rate = 0.1, number of estimators = 100.
6) XGB—Learning rate = 0.1, number of estimators = 100, max depth = 6.
7) LGB—Learning rate = 0.1, number of estimators = 100, max depth = 6.
8) SVM—C = 1.0, kernel = 'rbf'.
9) ANN—Hidden layers = [32,64], activation function = 'relu', optimizer = 'adam'.
10) FFNN—Hidden layers = [32,64], activation function = 'relu', optimizer = 'adam'.
11) LSTM—Units = 50, dropout rate = 0.2, optimizer = 'adam'.
12) CNN—Convolutional layers = [32,64], kernel size = (3,3), activation function = 'relu', optimizer = 'adam'.
13) CNN-LST-: CNN layers = [32,64], LSTM units = 50, dropout rate = 0.2, optimizer = 'adam'.

Table 1
Performance of machine learning models for identifying object type

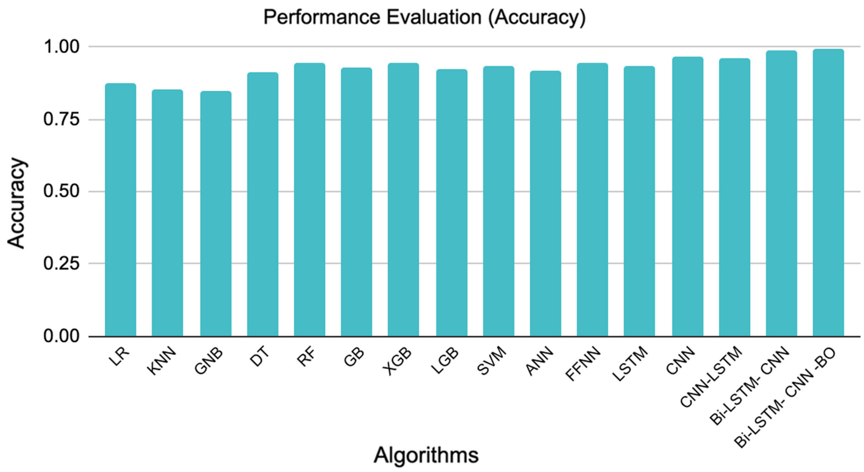
Algorithm	Accuracy	Precision	Recall	***F*-1 score**	Time (s)
LR	0.8744	0.8109	0.8246	0.81	1.332
KNN	0.8515	0.7568	0.8278	0.78	0.870
GNB	0.8492	0.8002	0.8384	0.81	1.001
DT	0.9132	0.8422	0.8625	0.85	1.114
RF	0.9439	0.8567	0.8897	0.86	1.782
GB	0.9287	0.8018	0.9208	0.86	3.654
XGB	0.9436	0.8824	0.9118	0.89	1.161
LGB	0.9222	0.8505	0.9046	0.87	3.112
SVM	0.9328	0.8646	0.9108	0.88	8.002
ANN	0.9208	0.8104	0.8755	0.84	6.038
FFNN	0.9451	0.9036	0.9228	0.91	14.732
LSTM	0.9344	0.8877	0.9105	0.89	12.343
CNN	0.9678	0.9412	0.9478	0.94	20.861
CNN-LSTM	0.9604	0.9238	0.9558	0.93	28.446
Bi-LSTM-CNN	0.9864	0.9123	0.9766	0.94	24.889
Bi-LSTM-CNN with Bayesian optimization	0.9916	0.9604	0.9824	0.97	47.662
LR	0.8744	0.8109	0.8246	0.81	1.332
KNN	0.8515	0.7568	0.8278	0.78	0.870
GNB	0.8492	0.8002	0.8384	0.81	1.001
DT	0.9132	0.8422	0.8625	0.85	1.114
RF	0.9439	0.8567	0.8897	0.86	1.782
GB	0.9287	0.8018	0.9208	0.86	3.654
XGB	0.9436	0.8824	0.9118	0.89	1.161
LGB	0.9222	0.8505	0.9046	0.87	3.112
SVM	0.9328	0.8646	0.9108	0.88	8.002
ANN	0.9208	0.8104	0.8755	0.84	6.038
FFNN	0.9451	0.9036	0.9228	0.91	14.732
LSTM	0.9344	0.8877	0.9105	0.89	12.343
CNN	0.9678	0.9412	0.9478	0.94	20.861
CNN-LSTM	0.9604	0.9238	0.9558	0.93	28.446
Bi-LSTM-CNN	0.9864	0.9123	0.9766	0.94	24.889
Bi-LSTM-CNN with Bayesian optimization	0.9916	0.9604	0.9824	0.97	47.662

14) Bi-LSTM-CNN with Bayesian optimization (Bi-LSTM-CNN-BO)—The model was optimized using Bayesian optimization with hyperparameters such as Number of CNN filters = 32, 64; Filter sizes = (3,3), (5,5); LSTM units = 50, 100; Dropout rate = 0.2, 0.3; Batch size = 32; Learning rate = 0.001; Optimization algorithm = Adam

Table 1 illustrates the performance evaluation of a comprehensive array of algorithms utilized in the analysis. The evaluation criteria include accuracy, precision, recall, $F1$ score, and the duration taken for model deployment. Across most algorithms, satisfactory performance is observed for multiclass classification. LR, KNN, and GNB exhibit relatively lower accuracy levels at 87%, 85%, and 84%, respectively. However, they boast shorter training times of 1.332 s, 0.87 s, and 1.001 s, respectively. Conversely, neural networks like CNN and CNN-LSTM demonstrate excellent performance with accuracy rates of 96%, accompanied by longer training times of 20.86 s and 28.446 s, respectively. Bi-LSTM-CNN surpasses others in accuracy with 98% but requires a longer training duration of 24.889 s. Notably, our proposed model, Bi-LSTM-CNN with Bayesian optimization (Bi-LSTM-CNN-BO), achieves the highest accuracy of 99.16%, albeit with a training time of 47.66 s. The rise in training time may be attributed to several reasons. Bi-LSTM-CNN models combine the complexities of both Bidirectional Long Short-Term Memory (Bi-LSTM) and CNNs, requiring substantial computational resources to train. Bayesian optimization explores a wide range of hyperparameters to find the optimal configuration for the model, which involves training and evaluating the model multiple times, increasing the overall training time. Moreover, Bayesian optimization may require many iterations before converging to the optimal set of hyperparameters, prolonging the training process as it iteratively refines the model configuration. We observe that the proposed model (Bi-LSTM-CNN-BO), achieves the highest accuracy of 99.16%, but has a training time of 47.66 s. The extended training duration is primarily due to the complexity of the model and the optimization process, and can be attributed to the following reasons.

1) Complex Model Architecture—The Bi-LSTM-CNN-BO model is defined by the synthesis of Bi-LSTM and CNN elements. The Bi-LSTM layer excels at processing sequential data (because it leverages bidirectional context), whereas the CNN layer focuses on the extraction of spatial features. However, the fusion of these two architectures leads to a notable increase in computational demands when compared to simpler models.

Figure 6
Performance evaluation for object type (Accuracy)



2) Bayesian optimization—This methodology involves the investigation of a wide range of hyperparameters to determine the most effective configuration for the model. Bayesian optimization employs probabilistic models to forecast the performance of diverse hyperparameter combinations. This iterative process requires numerous cycles of training and evaluation to fine-tune the hyperparameters, thus extending the overall training time.
3) Computational Resources—The training of a Bi-LSTM-CNN-BO model requires substantial computational resources—this is largely due to its complex architecture and the involved optimization processes. The necessity for significant computational power (to handle the increased number of parameters and iterations) is a crucial factor that contributes to the prolonged training duration.
4) Hyperparameter Search—Bayesian optimization includes the evaluation of multiple hyperparameter sets to attain convergence on the optimal configuration. Each iteration during this search process entails training the model, which cumulatively extends the overall training time. Figures 6 and 7 indicate that, however, the Bi-LSTM-CNN-BO model necessitates more computational time, and it concurrently achieves improved accuracy; this underscores the inherent trade-off between model performance and computational efficiency.

Table 2 illustrates the performance evaluation of a comprehensive array of algorithms utilized in the analysis. The evaluation criteria include accuracy, precision, recall, *F*1 score, and the duration taken for model deployment. Across most algorithms, satisfactory performance is observed for multiclass classification. LR, KNN, and GNB exhibit relatively lower accuracy levels at 93%, 91%, and 89%, respectively. However, they boast shorter training times of 0.414 s, 0.668 s, and 1.003 s, respectively. Conversely, neural networks like CNN and CNN-LSTM demonstrate excellent performance with accuracy rates of 97.99% and 98.04%, accompanied by longer training times of 23.889 s and 38.775 s, respectively. Bi-LSTM-CNN surpasses others in accuracy with 98.99% but requires a longer training

Figure 7
Performance evaluation for object type (Time)

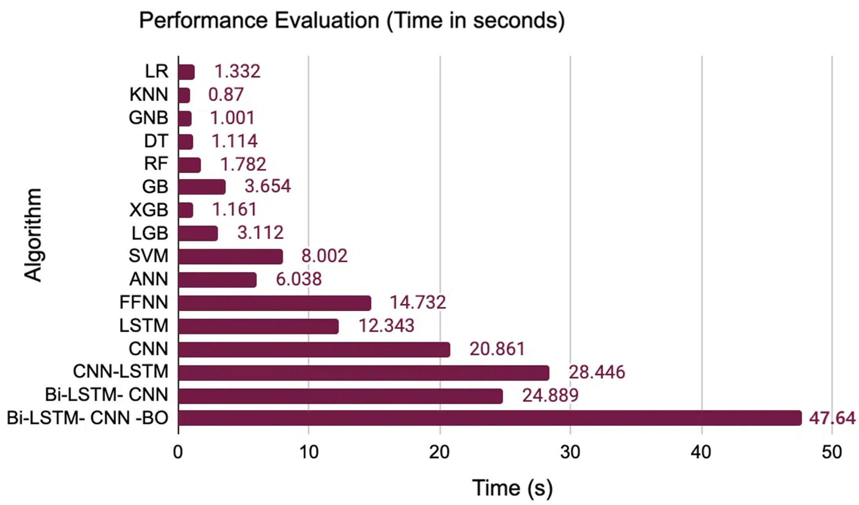
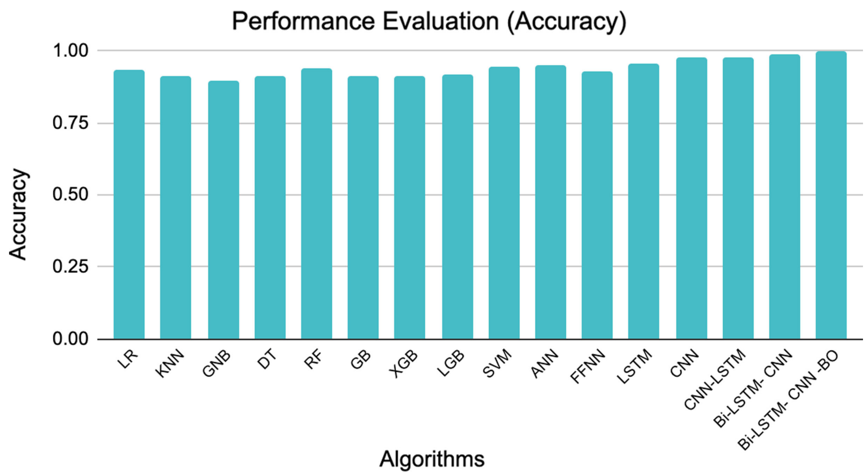


Table 2
Performance of machine learning models for identifying RCS size

Algorithm	Accuracy	Precision	Recall	***F*-1 score**	Time (s)
LR	0.9333	0.9104	0.9040	0.90	0.414
KNN	0.9122	0.8566	0.8778	0.86	0.668
GNB	0.8989	0.8678	0.8944	0.87	1.003
DT	0.9111	0.8222	0.8899	9.85	1.182
RF	0.9389	0.7584	0.7784	0.76	2.324
GB	0.9144	0.8333	0.8541	0.84	2.882
XGB	0.9123	0.8312	0.8717	0.85	2.006
LGB	0.9189	0.8678	0.8802	0.87	2.267
SVM	0.9465	0.8686	0.8993	0.87	7.067
ANN	0.9499	0.9129	0.9118	0.91	5.114
FFNN	0.9312	0.8504	0.8894	0.86	11.998
LSTM	0.9552	0.8436	0.8902	0.86	17.614
CNN	0.9799	0.8211	0.8894	0.85	23.889
CNN-LSTM	0.9804	0.9004	0.9122	0.90	38.775
Bi-LSTM-CNN	0.9899	0.9234	0.9403	0.93	45.222
Bi-LSTM-CNN with Bayesian optimization	0.9998	0.9666	0.9438	0.95	53.023

Figure 8
Performance evaluation for RCS size (Accuracy)



duration of 45.222 s. Notably, our proposed model, Bi-LSTM-CNN with Bayesian optimization (Bi-LSTM-CNN-BO), achieves the highest accuracy of 99.98%, albeit with a training time of 53.023 s. Figures 8 and 9 depict the performance evaluation concerning accuracy and time (seconds) for all the deployed algorithms.

3.4. Comparative analysis

In this segment, a comparative examination between the proposed approach and several existing methods for identifying space debris has been conducted. The comparative analysis has been performed using the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) guidelines, for reporting systematic reviews and meta-analyses) in a transparent and standardized way. Table 3 outlines the results of this comparative analysis.

1) **Title**
   - Comparative Analysis of Proposed Work with Relevant Works
2) **Eligibility Criteria**
   - Studies that focus on space debris monitoring, classification, and removal.
   - Studies employing ML and DL techniques.
   - Studies provide performance metrics for accuracy, error rate, or computational efficiency.
3) **Information Sources**
   - Academic databases such as IEEE Xplore, SpringerLink, and ScienceDirect.
   - Keyword searches include terms such as “space debris”, “machine learning”, “deep learning”, “PSO”, “SVM”, “LSTM”, “ANN”, and “clustering”.

Figure 9
Performance evaluation for RCS size (Time)

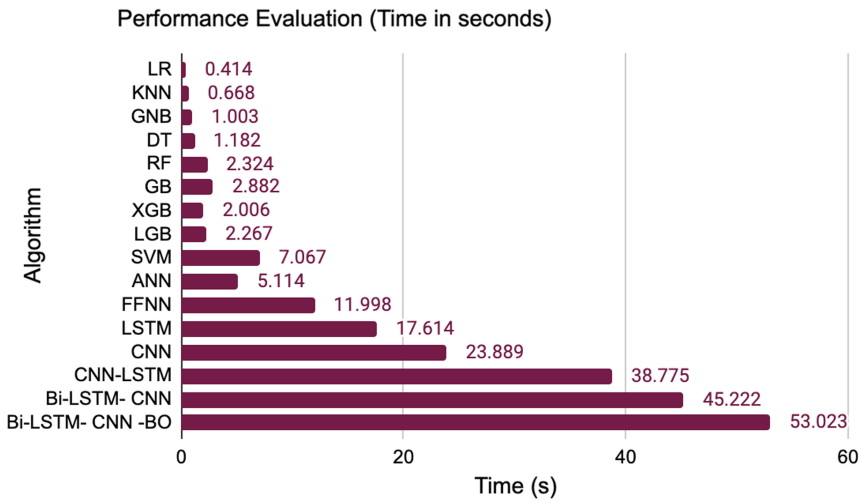


Table 3
Comparative analysis of proposed work with relevant works

Author and Year	Proposed work	Methodology/ Parameters	Results
Jordan et al. [8]	Space Debris Removal	PSO-based method to estimate inertia parameters for uncooperative satellites	The proposed method exhibits less than 1% error
Ryan et al. [9]	Orbital debris impact risk assessments	ML algorithms like ANN, SVM, XGBoost, etc.,	XGBoost shows the best performance with an accuracy 97.4%
Guo et al. [14]	Clustering spectral polarization data from space debris	Hybrid fuzzy C-means (FCM) algorithm model incorporating hierarchical agglomerative clustering (HAC)	Accuracy of 96.92%
Qashoa and Lee [19]	Classifying low-orbit space objects	Algorithms like SVM, LSTM	LSTM exhibits an accuracy of 92%
Zhao et al. [20]	Task Allocation for Space Debris Removal	Improved Particle Swarm optimization (PSO) Algorithm	Improved PSO is 22.8% faster than traditional PSO
Our Proposed Work	**Monitoring and Classifying Space Debris**	**Extensive ML algorithms, proposed Bi-LSTM-CNN, and Bayesian optimization**	**The proposed method archives accuracy of 99.16% and 99.98%**

4) **Search Strategy**
  - Searches were conducted using combinations of keywords such as “space debris”, “machine learning”, “deep learning”, “PSO”, “SVM”, “LSTM”, “ANN”, and “clustering”.
  - Filters applied for publication year 2023.

5) **Study Selection Process**
  - The initial search yielded a total of ten records. None of the duplicates were removed. Titles and abstracts of five records were screened for relevance. Full texts of ten articles were assessed for eligibility. Five studies were included in the final analysis.

The comparative analysis highlights the strengths and weaknesses of various methodologies employed for space debris monitoring and removal. Jordan et al. [8] demonstrated that a PSO-based method for estimating inertia parameters of uncooperative satellites shows less than 1% error, indicating high precision. Ryan et al. [9] found that XGBoost outperforms ANN and SVM with 97.4% accuracy in orbital debris impact risk assessments. Guo et al. [14] achieved 96.92% accuracy using a hybrid FCM-HAC model for clustering spectral polarization data. Qashoa and Lee [19] reported that LSTM achieves 92% accuracy in classifying low-orbit space objects. Zhao et al. [20] improved the PSO algorithm, increasing its speed by 22.8% for task allocation in debris removal. Our proposed Bi-LSTM-CNN model, optimized with Bayesian optimization, outperforms these methods with 99.16% accuracy in space debris classification and 99.98% in RCS size prediction, indicating a substantial improvement in detection and classification capabilities. It is observed that

1) The Bi-LSTM-CNN model presented in this study demonstrates a notable improvement in performance metrics when juxtaposed with current methodologies. This finding underscores the effectiveness of hybrid models that fuse LSTM and CNN architectures, enhanced by Bayesian optimization, in addressing the complex challenges associated with monitoring space debris. However, the implications of this research are

far-reaching, because they suggest innovative approaches to a pressing issue. Although the results are promising, further investigation is needed to fully understand the potential of these techniques.
2) In addition to attaining elevated levels of accuracy, the suggested method effectively tackles the significant concern of training duration—thus ensuring operational efficiency in real-time scenarios. This positions it as a resilient solution for implementation in (various contexts); however, challenges may arise. Although it is promising, the effectiveness might be influenced by external factors, but the overall framework remains sound because it incorporates essential principles space missions.
3) By employing advanced AI techniques, (this) approach tackles the research shortcomings that are often found in traditional strategies. The integration of Bi-LSTM and CNN architectures, combined with Bayesian optimization, creates an innovative framework that enhances both accuracy and operational efficiency in systems designed for space debris detection.
4) The improved performance of this method carries significant implications for managing space debris. It enhances space situational awareness, optimizes space traffic management, and supports the long-term sustainability of activities in outer space. This progressive strategy not only mitigates the risks of collisions; however, it also fosters international cooperation and accountability in the domain of space exploration.

3.5. Observations

Several key points have been observed based on the experimental and comparative analysis.

1) Extensive prior research within the realm of space exploration (utilizing AI) has predominantly focused on specific methodologies. This study undertakes a comprehensive examination that incorporates a variety of algorithms; thus, it advances the field of space debris analysis.
2) The evaluation of ML algorithms—encompassing classical methods, ensemble techniques, and neural networks—is conducted based on essential performance indicators: precision, accuracy, recall, $F1$ score, and deployment time.
3) A highly effective Bi-LSTM-CNN model, refined through Bayesian optimization, stands out as the leading performer across various analyses, demonstrating its capability in identifying and classifying space debris.
4) The initial phase of the analysis categorizes debris according to object type (which includes debris, payloads, and rocket bodies). However, the subsequent phase emphasizes the identification of RCS size, a vital element for comprehending both the characteristics of debris and the associated risks. Although this study presents significant findings, more research is necessary to explore additional methodologies.
5) The comparative analysis suggests that this methodology achieves commendable accuracy in relation to earlier studies; however, it underscores its proficiency in identifying and monitoring space debris.
6) Although the results are promising, there are areas that require further investigation. This is particularly important because the implications of space debris are significant for future missions.

3.6. Limitations

Some limitations of the study are as follows:

1) The research utilized a specific dataset for its analysis, which may not fully encompass the entire spectrum of characteristics and behaviors linked to space debris. This limitation could significantly affect the degree to which the findings are generalizable.
2) The study assumes the dataset's quality and reliability for its analysis; nevertheless, inconsistencies, missing data, or inaccuracies within the dataset could introduce biases and undermine the robustness of the results.
3) The complex nature of the Bi-LSTM-CNN model, combined with Bayesian optimization, may pose challenges in terms of interpretability and scalability. Moreover, the computational requirements needed for training and implementing this model might limit its practical application in certain situations.
4) External factors, such as changes in space policy, technological progress, or unforeseen events, could modify the dynamics of space debris, potentially making the study's conclusions obsolete or less relevant over time.

4. Conclusion

This study presents a comprehensive analysis of the intricate processes involved in the identification and classification of space debris, offering substantial insights and proposing potential avenues for future research. By utilizing a range of ML algorithms—such as LR, KNN, GNB and advanced neural networks, including CNN and CNN-LSTM—the study assesses the efficacy of various models in categorizing types of space debris and forecasting their RCS sizes. Notably, the introduction of a sophisticated Bi-LSTM-CNN model, refined through Bayesian optimization techniques, exhibited remarkable accuracy in classification tasks; however, this achievement came at the expense of prolonged training times due to the model's inherent complexity. The findings underscore the urgent necessity for precise identification and monitoring of space debris, as it poses considerable threats to space missions and satellite operations.

Although the proposed model demonstrated commendable performance, it is crucial to acknowledge the limitations that are inherent in this study. The reliance on a specific dataset may, in fact, restrict the applicability of the findings; furthermore, potential issues related to data quality could lead to biases. Future research should aim to address these limitations by utilizing larger and more diverse datasets, ensuring the integrity and reliability of the data, and exploring advanced methodologies to enhance model performance and efficiency. This is important because there should also be a focus on integrating real-time monitoring systems and predictive analytics to mitigate the escalating risks associated with space debris and to safeguard space infrastructure. In summary, while this study signifies a substantial advancement in the realm of space debris analysis, it also highlights the necessity for ongoing innovation and collaboration within the scientific community; however, to effectively tackle the challenges posed by space debris and to promote the sustainability of space exploration initiatives, these efforts are essential.

Acknowledgement

The author would like to thank the anonymous referees for their useful suggestions.

Conflicts of Interest

The author declares that she has no conflicts of interest to this work.

Data Availability Statement

The datasets cited in this manuscript are publicly available and can be accessed from the original sources referenced in the text. The data that support the findings of this study are openly available in Kaggle at https://www.kaggle.com/datasets/kandhalkhandeka/satellites-and-debris-in-earths-orbit/data. No additional data were generated or analyzed during the current study.

Author Contribution Statement

Ishaani Priyadarshini: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition.

References

[1] Svotina, V. V., & Cherkasova, M. V. (2023). Space debris removal – Review of technologies and techniques. Flexible or virtual connection between space debris and service spacecraft. *Acta Astronautica*, *204*, 840–853. https://doi.org/10.1016/j.actaastro.2022.09.027

[2] Ledkov, A. S., & Aslanov, V. S. (2023). Active space debris removal by ion multi-beam shepherd spacecraft. *Acta Astronautica*, *205*, 247–257. https://doi.org/10.1016/j.actaastro.2023.02.003

[3] Wang, C., Chen, D., Liao, W., & Liang, Z. (2023). Autonomous obstacle avoidance strategies in the mission of large space debris removal using potential function. *Advances in Space Research*, *72*(7), 2860–2873. https://doi.org/10.1016/j.asr.2022.04.013

[4] Viavattene, G., Devereux, E., Snelling, D., Payne, N., Wokes, S., & Ceriotti, M. (2022). Design of multiple space debris removal missions using machine learning. *Acta Astronautica*, *193*, 277–286. https://doi.org/10.1016/j.actaastro.2021.12.051

[5] Wu, Y., Peng, K. L., Yao, Y., & Guo, Y. (2024). Sustainable space travel: What can we do in education from economic and environmental perspectives? *Sustainability*, *16*(2), 684. https://doi.org/10.3390/su16020684

[6] Deng, S., Su, J., Zhu, Y., Yu, Y., & Xiao, C. (2024). Forecasting carbon price trends based on an interpretable light gradient boosting machine and Bayesian optimization. *Expert Systems with Applications*, *242*, 122502. https://doi.org/10.1016/j.eswa.2023.122502

[7] Khan, M. A., Arshad, H., Khan, W. Z., Alhaisoni, M., Tariq, U., Hussein, H. S., . . . , & Elashry, A. (2023). HGRBOL2: Human gait recognition for biometric application using Bayesian optimization and extreme learning machine. *Future Generation Computer Systems*, *143*, 337–348. https://doi.org/10.1016/j.future.2023.02.005

[8] Jordan, J., Posada, D., Zuehlke, D., Nocerino, A., Fontdegloria, P., John, S., . . . , & Henderson, T. (2023). Estimation of uncooperative satellite inertia parameters for space debris removal using particle swarm optimization. In *2023 IEEE Aerospace Conference*, 1–13. https://doi.org/10.1109/AERO55745.2023.10115606

[9] Ryan, S., Sushma, N. M., Le, H., Kumar, A. V. A., Berk, J., Nguyen, T. M., . . . , & Venkatesh, S. (2023). The application of machine learning in micrometeoroid and orbital debris impact protection and risk assessment for spacecraft. *International Journal of Impact Engineering*, *181*, 104727. https://doi.org/10.1016/j.ijimpeng.2023.104727

[10] Wu, D., & Rosengren, A. J. (2023). An investigation on space debris of unknown origin using proper elements and neural networks. *Celestial Mechanics and Dynamical Astronomy*, *135*, 1–20. https://doi.org/10.1007/s10569-023-10157-0

[11] Xu, Y., Liu, X., He, R., Zhu, Y., Zuo, Y., & He, L. (2023). Active debris removal mission planning method based on machine learning. *Mathematics*, *11*(6), 1419. https://doi.org/10.3390/math11061419

[12] Cimmino, N., Opromolla, R., & Fasano, G. (2023). Machine learning-based approach for ballistic coefficient estimation of resident space objects in LEO. *Advances in Space Research*, *71*(12), 5007–5025. https://doi.org/10.1016/j.asr.2023.02.007

[13] Salmaso, F., Trisolini, M., & Colombo, C. (2023). A machine learning and feature engineering approach for the prediction of the uncontrolled re-entry of space objects. *Aerospace*, *10*(3), 297. https://doi.org/10.3390/aerospace10030297

[14] Guo, F., Zhu, J., Huang, L., Li, H., Deng, J., Jiang, H., & Hou, X. (2023). Enhancing spatial debris material classifying through a hierarchical clustering-fuzzy C-means integration approach. *Applied Sciences*, *13*(8), 4754. https://doi.org/10.3390/app13084754

[15] Munivenkatappa, H., Singh, G., Kumar, V., Buduru, A. B., & Biswas, S. K. (2024). Tracking an untracked space debris after an inelastic collision using physics informed neural network. *Scientific Reports*, *14*, 3350. https://doi.org/10.1038/s41598-024-51897-9

[16] Zhang, H., Zhang, Y., Feng, Q., & Zhang, K. (2024). Review of machine-learning approaches for object and component detection in space electro-optical satellites. *International Journal of Aeronautical and Space Sciences*, *25*, 277–292. https://doi.org/10.1007/s42405-023-00653-w

[17] Pensado, E. A., de Santos, L. M. G., Jorge, H. G., & Sanjurjo-Rivo, M. (2023). Deep learning-based target pose estimation using LiDAR measurements in active debris removal operations. *IEEE Transactions on Aerospace and Electronic System*, *59*(5), 5658–5670. https://doi.org/10.1109/TAES.2023.3262505

[18] Yang, K., Liu, L., & Wen, Y. (2024). The impact of Bayesian optimization on feature selection. *Scientific Reports*, *14*, 3948. https://doi.org/10.1038/s41598-024-54515-w

[19] Qashoa, R., & Lee, R. (2023). Classification of low earth orbit (LEO) resident space objects' (RSO) light curves using a support vector machine (SVM) and long short-term memory (LSTM). *Sensors*, *23*(14), 6539. https://doi.org/10.3390/s23146539

[20] Zhao, Y., Wu, P., Yu, H., & Tong, Y. (2023). Task allocation for space debris removal based on improved particle swarm optimization algorithm. In *2023 International Conference on Cyber-Physical Social Intelligence*, 399–404. https://doi.org/10.1109/ICCSI58851.2023.10303952

How to Cite: Priyadarshini, I. (2025). Enhanced Space Debris Detection and Monitoring Using a Hybrid Bi-LSTM-CNN and Bayesian Optimization. *Artificial Intelligence and Applications*, *3*(1), 43–55. https://doi.org/10.47852/bonviewAIA42023741