aerospace

MDPI

Article

Bidirectional Long Short-Term Memory Development for Aircraft Trajectory Prediction Applications to the UAS-S4 Ehécatl

Seyed Mohammad Hashemi *, Ruxandra Mihaela Botez and Georges Ghazi

Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity LARCASE, École de Technologie Supérieure (ÉTS), Université de Québec, Montreal, QC H3C 1K3, Canada; ruxandra.botez@etsmtl.ca (R.M.B.); georges.ghazi@etsmtl.ca (G.G.)

* Correspondence: seyed-mohammad.hashemi@etsmtl.ca

Abstract: The rapid advancement of unmanned aerial systems in various civilian roles necessitates improved safety measures during their operation. A key aspect of enhancing safety is effective collision avoidance, which is based on conflict detection and is greatly aided by accurate trajectory prediction. This paper represents a novel data-driven trajectory prediction methodology based on applying the Long Short-Term Memory (LSTM) prediction algorithm to the UAS-S4 Ehécatl. An LSTM model was designed as the baseline and then developed into a Staked LSTM to better capture complex and hierarchical temporal trajectory patterns. Next, the Bidirectional LSTM was developed for a better understanding of the contextual trajectories from both its past and future data points, and to provide a more comprehensive temporal perspective that could enhance its accuracy. LSTM-based models were evaluated in terms of mean absolute percentage errors. The results reveal the superiority of the Bidirectional LSTM, as it could predict UAS-S4 trajectories more accurately than the Stacked LSTM. Moreover, the developed Bidirectional LSTM was compared with other state-of-the-art deep neural networks aimed at aircraft trajectory prediction. Promising results confirmed that Bidirectional LSTM exhibits the most stable MAPE across all prediction horizons.

Keywords: bidirectional; long short-term memory; trajectory prediction; unmanned aerial systems

Citation: Hashemi, S.M.; Botez, R.M.; Ghazi, G. Bidirectional Long Short-Term Memory Development for Aircraft Trajectory Prediction Applications to the UAS-S4 Ehécatl. *Aerospace* **2024**, *11*, 625. https://doi.org/10.3390/aerospace11080625

Academic Editor: Álvaro Rodríguez-Sanz

Received: 18 June 2024
Revised: 29 July 2024
Accepted: 30 July 2024
Published: 31 July 2024

1. Introduction

Unmanned aerial systems (UASs) have rapidly evolved from military to civil systems [1]. As technology has progressed, the cost, size, and complexity of UASs have diversified, making them accessible to a broader range of users [2]. These advancements have paved the way for a variety of applications, from aerial photography and surveying [3] to disaster relief and agricultural monitoring [4,5].

With the fast-growing application of UASs in various sectors, skies have become more congested with drones, and the risk of aerial collisions has intensified [6]. UAS collisions not only pose threats to property and to UASs themselves but, more critically, to human life [7,8]. Additionally, given the varying levels of operator experience and the diverse range of UAS sizes and capabilities, ensuring consistent safety standards becomes increasingly challenging [9,10]. Addressing these concerns requires reliable regulatory frameworks and advanced collision-avoidance technologies to ensure that the integration of UASs into airspace remains safe [11].

The precise prediction of UAS trajectories is a key element to designing an advanced collision avoidance algorithm and improving air traffic management (ATM) performance [12]. Hence, developing an algorithm to predict the path of a UAS becomes vital [13]. In knowing where a UAS is likely to be at any given time in the future, traffic controllers can better manage airspace, allocate flight corridors, and ensure safe distances between aircraft [14]. Moreover, in real-time flight operations, collision avoidance systems rely on these trajectory predictions to take proactive measures, such as rerouting or adjusting altitudes, to prevent potential collisions [15]. All in all, the accuracy of UAS trajectory prediction is infrastructure for ensuring safety and efficiency in congested skies [16].

Over recent years, UAS trajectory prediction has seen significant advancements underpinned by technological and algorithmic developments [17,18]. These methods range from physical models [19], which rely on fundamental physics principles, to statistical models that use historical data for forecasting [20].

Modern trajectory prediction tools now incorporate sophisticated artificial intelligence techniques [21], allowing for real-time predictions based on dynamic environmental conditions [22]. By integrating big data, these systems can learn from vast amounts of previous flight data and refine their predictions over time [23,24]. In this way, machine learning algorithms [25], especially deep learning models [26], are being successfully employed. These models can learn complex patterns from large datasets and often provide more accurate predictions, especially when the system encounters uncertainties that might not be explicitly programmed into traditional models.

Deep learning algorithms, particularly those underpinned by multi-layered neural networks, have revolutionized trajectory prediction [27]. Convolutional Neural Networks (CNNs) [28], Generative Adversarial Networks (GANs) [29], autoencoders [30], and Random Forest [31] contribute, in particular, to trajectory feature identification and data generation. Each of these methodologies brings unique capabilities to the challenge of predicting motion paths. CNNs can extract and utilize spatial hierarchies from trajectory data where trajectories are heavily influenced by GPS features. However, it has limited capabilities in handling temporal dependencies [28]. GAN is utilized for its ability to generate diverse and plausible future trajectories, proving invaluable in dynamic air corridors. The training of GANs, however, is marred by stability issues, such as mode collapse, where the diversity of the data is not adequately captured by the model [29].

An autoencoder is employed for its ability to condense trajectory data into a more manageable, lower-dimensional space, thus improving the computational efficiency of prediction models. Yet, the accuracy of data reconstruction by the autoencoder is problematic, as some crucial information has been lost in decoded outputs [30]. Random Forest is employed for its robust performance across varied conditions, benefiting from an ensemble nature that helps reduce variance and prevent overfitting. However, its effectiveness in capturing the sequential dependencies essential to trajectory data and crucial for accurate predictions degraded [31].

LSTMs, a type of recurrent neural network (RNN) [32], have shown a very good performance in processing and predicting time-series data [33]. Among these algorithms, Long Short-Term Memory (LSTM) networks stand out for their unparalleled ability to handle sequential data and capture temporal dependencies, making them well-suited for predicting future motions based on historical patterns. Its intrinsic strength positions LSTM as a leading tool in the domain of trajectory prediction, ensuring more accurate and reliable predictions compared to non-recurrent algorithms [34]. Hence, this study focuses on the development of an LSTM architecture to better understand contextual trajectories, aiming to achieve more accurate trajectory predictions.

In this way, the contribution of this article can be stated as follows. First, the UAS-S4 trajectory prediction is formulated as a time-series regression problem. The Stacked LSTM architecture is then developed to predict future trajectories, outperforming the LSTM in capturing trajectory patterns more precisely. Eventually, a Bidirectional LSTM is designed such that it empowers the capture of temporal dependencies and early detection of sharp trajectory changes. Therefore, it can be more resilient to imperfections in trajectory data and perform better predictions compared to the Stacked LSTM.

The organization of the paper is as follows. In Section 2, the related works on LSTM-based contextual trajectory prediction problems are investigated. The proposed methodologies using the Stacked LSTM and Bidirectional LSTM are described in Section 3. Trajectory prediction performance is numerically analyzed and discussed in Section 4. A comprehensive conclusion is given in Section 5.

2. Related Works

LSTM algorithms have been significantly used for solving various trajectory-based problems. These types of algorithms have been successfully employed for autonomous driving, maritime traffic prediction, robot navigation, human activity recognition, crowd management, and air traffic management. A long-term interactive trajectory prediction method was designed [35], and it utilized a hierarchical multi-sequence learning network to capture dependencies between multiple interacting vehicles, and it could automatically learn high-level dependencies. Its innovation lies in the use of a structural LSTM network. The method assigned an LSTM for each interacting vehicle. These LSTMs then share their cells and hidden states with neighboring LSTMs in a spatial manner through radial connections. This process allows the network to analyze its own output state as well as the states of other LSTMs in their deeper layers. Based on these output states, the network is able to develop trajectory predictions for the surrounding vehicles [35].

With the aim of marine vessel trajectory prediction, an unsupervised trajectory prediction methodology with prediction regions at arbitrary probabilities was introduced. This approach leverages two methods: LSTM prediction region learning and wild bootstrapping [36]. The study demonstrated that both the autoencoder-based and wild bootstrapping region prediction algorithms could effectively predict vessel trajectories. These predictions could be applied to detect abnormal marine traffic in an unsupervised manner by evaluating the predicted values at desired prediction probabilities [37].

In the context of robot navigation, an LSTM network was introduced as an online search agent to address the challenges of path planning for mobile robots in unfamiliar environments. This approach relies solely on local map awareness, obtained through a Laser Range Finder (LRF) sensor, and relative information between the robot's position and the destination. The study thoroughly examined the final structure of the LSTM network and assessed its performance in comparison to that of the A* algorithm, which is a well-established method that employs a best-first search approach for path planning [38].

An innovative approach was presented for predicting the future trajectory of pedestrians based on a limited history of their past actions, as well as those of their neighboring pedestrians. This work was developed using an LSTM-based attention model, which incorporated both "soft" and "hard" attention mechanisms. This approach effectively mapped trajectory information from the local neighborhood to predict future positions of the pedestrian of interest. The obtained results demonstrated how a straightforward approximation of hard attention weights could be integrated with soft attention weights, making the model suitable for complex scenarios involving numerous neighbors [39].

In the context of crowd management, an innovative LSTM model was designed to collectively analyze the behaviors of multiple individuals within a given scene. Unlike traditional LSTMs, this model incorporated a new pooling layer that facilitated information sharing between multiple LSTMs. This pooling layer aggregates hidden representations from LSTMs associated with neighboring trajectories, effectively capturing interactions and dependencies among individuals within the same neighborhood [40].

For ATM systems, a new aircraft trajectory prediction (ATP) model was introduced based on a constrained LSTM. This model was specifically designed to account for the dynamic characteristics of an aircraft flight, with particular attention given to the climbing, cruising, and descending/approaching phases. A notable feature of this model is its capability to maintain long-term dependencies while incorporating such dynamic physical constraints. Data segmentation and preprocessing were performed using a combination of density-based spatial clustering applications with noise [41]. The LSTM model was then developed to capture long-term trajectory dependencies to improve the accuracy of trajectory predictions. The sliding windows technique [42] was utilized within the LSTM to maintain data continuity and preserve dynamic dependencies between adjacent states in long sequences.

The LSTM architecture was developed into "Deep Long Short-Term Memory" (D-LSTM) for the ATP. The proposed D-LSTM model enhanced the accuracy of aircraft trajectory predictions, particularly in complex flight scenarios. It effectively integrated the multi-dimensional features of aircraft trajectories into the LSTM framework and was empirically validated using real-world ADS-B flight data [43].

With respect to the advancement of designed LSTMs, a security issue arose from data-driven ATP problems. Hence, an LSTM model was developed for robustness against adversarial attacks [44]. The sensitivity of the model was investigated, and then the model was retrained using adversarial samples generated through the adaptive fast gradient sign method. The model was retrained using a 4-D trajectory of a UAS-S4 and was able to predict future trajectories accurately despite approaching adversarial samples [27].

In technical terms, using adversarial retraining to improve robustness can compromise prediction accuracy and degrade LSTM's efficiency. Therefore, the LSTM must be enhanced to achieve its maximum potential by implementing advanced techniques, such as Bidirectional LSTMs [45], Stacked LSTMs [46], Gated Recurrent Units [47], and attention mechanisms [48]. In the following section, we delve deeper into these methodologies and adapt them specifically for the ATP task.

3. Methodology

In order to utilize the LSTM model for the ATP, we need to formulate it as a time-series regression problem [49]. In this way, firstly, an aircraft should be arranged in an air corridor considering "timestamp". Then, it is assumed that an aircraft is navigating within its designated pathway, as illustrated in Figure 1 [50].

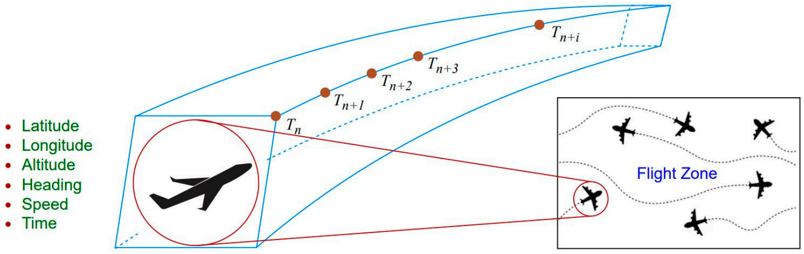

Figure 1. Time-series ATP problem in an air corridor [50].

The goal is to forecast the aircraft's future path on its upcoming m steps (prediction horizon) using GPS data at any given moment (T_n). The GPS data encompasses parameters including latitude, longitude, altitude, heading, speed, and time.

Then, an LSTM model needs to be developed for the ATP task. Improving the performance of LSTM networks for time-series problems involves a combination of architectural improvements and training strategies [51]. The ATP problem inherently requires understanding complex sequential patterns, predicting future movements based on historical data, and sometimes even integrating upcoming contextual clues. Given the nature of this task, the enhancements brought by the Stacked LSTMs and Bidirectional LSTMs (BiLSTMs) can be particularly beneficial.

Stacked Long Short-Term Memory (SLSTM) networks represent an advanced configuration of the LSTM architecture. The distinctive feature of Stacked LSTMs is their layered structure. Instead of relying on a single LSTM layer, this model comprises multiple LSTM layers stacked sequentially. Each layer processes the output sequence of its predecessor, creating a cascade of information through the layers [52]. Figure 2 illustrates our Stacked LSTM designed for the ATP problem.

The principle behind this architecture is not just adding more neurons to a single layer but adding depth (more layers) to the network. The idea is that as trajectory data progress through these multiple layers, they capture intricate patterns and dependencies, building on the simpler patterns detected by the initial layers.

Let us consider multiple LSTM layers stacked on top of each other. UAS-S4 trajectories (as input samples X) and actual target trajectory Y are applied to the model, and, after processing using LSTM blocks, activation functions ($h = tanh$) introduce non-linearity into the output of a neuron, which provides predicted trajectories $\widetilde{Y}$. This deep architecture can capture more complex patterns and representations in the data, potentially leading to enhanced model performance, especially in datasets that possess hierarchical or multi-tiered characteristics.

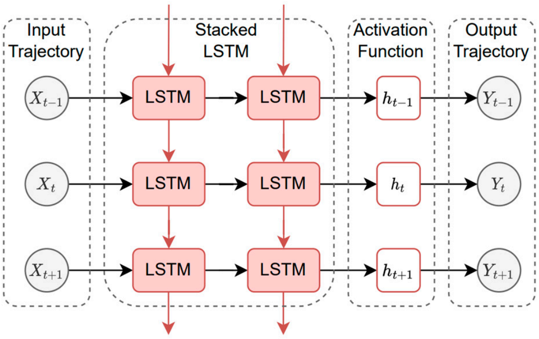

Figure 2. The proposed Stacked LSTM architecture for the ATP time-series problem.

Algorithm 1 represents the Stacked LSTM algorithm that trained a model for the UAS-S4 trajectory prediction.

Algorithm 1. Stacked LSTM Algorithm

```
    Initialize Parameters:
1.      Define the number of layers $l$ in the LSTM.
2.      Define the number of hidden units in each layer $H$.
3.      Initialize weights $W$, and biases $b$ for each layer.
    Input Preparation:
4.      Prepare input sequence $X = \{x_1, x_2, \ldots, x_T\}$ and standardize it.
    Procedure:
5.      For each time step, $t = 1 : T$.
6.          For each layer, $l = 1 : L$.
7.              Input gate : $i_t^l = \sigma\left(W_{ii}^l \cdot \left[h_{t-1}^l, x_t^l\right] + b_{ii}^l\right)$.
8.              Forget gate : $f_t^l = \sigma\left(W_{if}^l \cdot \left[h_{t-1}^l, x_t^l\right] + b_{if}^l\right)$.
9.              Cell candidate : $\widetilde{C}_t^l = \tanh\left(W_{ig}^l \cdot \left[h_{t-1}^l, x_t^l\right] + b_{ig}^l\right)$.
10.             Output gate : $o_t^l = \sigma\left(W_{io}^l \cdot \left[h_{t-1}^l, x_t^l\right] + b_{io}^l\right)$.
11.             Cell State update : $o_t^l = f_t^l * C_{t-1}^l + i_t^l * \widetilde{C}_t^l$.
12.             Hidden state update : $h_t^l = o_t^l * \tanh\left(C_t^l\right)$.
13.             Set output of current layer as input to next layer : $x_t^{l+1} = h_t^l$ .
    Backpropagation:
14.     Compute gradients of the loss function with respect to all parameters.
15.     Update model parameters using Stochastic Gradient Descent optimizer.
    Iteration/Epoch Control:
16.     Repeat steps 5–15 for each batch of data, and for each epoch, until convergence or the
        maximum number of epochs is reached.
17. Output = $h_t^l$.
```

The first step in this algorithm involves initializing the parameters, which sets the stage for the network's structure and its learning capacity. This includes specifying the number of layers (l) and the number of hidden units (H) in each layer. Additionally, weights and biases for each LSTM unit across all layers are initialized, which is crucial for the gate operations within each LSTM cell. These operations include the input gate, forget gate, output gate, and cell state adjustments that regulate the flow of information through the network, determining what to retain or forget as the data progress through the model.

During the forward pass, the Stacked LSTM processes the input sequence step-by-step. At each time step t, the input x_t^l is fed into the first layer, which processes the data and outputs a hidden state h_t^l. This output then serves as the input to the second layer, continuing in this fashion through all layers. Each layer's LSTM cells independently execute their gate operations and state updates, with the output of the last layer in the sequence representing the final output for that time step.

Trajectory patterns, especially in dynamic environments, are not solely governed by historical flight trajectories. Instead, they often intertwine with anticipatory reactions to upcoming events. Let us consider the trajectory of an aircraft navigating along a corridor. While its previous route provides insights into its immediate direction, forthcoming challenges, such as encountering deviations due to uncertainties or unpredictable failures, could play an influential role in its future path. A model that relies predominantly on historical data, such as Stacked LSTM, might inadvertently overlook these critical dynamic cases.

This is where the strength of BiLSTMs comes into play. By processing data from both forward and backward directions, BiLSTMs create a comprehensive contextual understanding of the trajectory for every point in the data sequence [45]. In the realm of trajectory prediction, this dual-direction processing offers several distinct advantages:

- "Understanding Trajectories": BiLSTMs make each trajectory prediction within a dual context derived from both past actions and potential future occurrences. This approach aligns predictions closer to real-world trajectories, which are often adjusted based on retrospective and prospective cues.
- "Early Detection of Sharp Trajectory Changes": Trajectories can sometimes exhibit sudden resolution advisories. BiLSTMs, through their backward pass, are strong at recognizing these shifts, leading to more accurate predictions.
- Resilience in Data Imperfections: In practical scenarios, trajectory data might not be perfect due to missing data or inherent noise. BiLSTMs' bidirectional processing offers a form of data redundancy, enhancing the model's resilience against such imperfections and ensuring more stable predictions.

Compared to the Stacked LSTM, BiLSTM networks are an evolved form of recurrent neural network (RNN) specifically designed to enhance sequence modeling. Their unique capability is rooted in their ability to observe both the preceding and the succeeding contextual trajectories within a sequence. Unlike standard LSTMs that progress linearly from the beginning to the end of a sequence, BiLSTMs adopt a dual-directional approach. In other words, BiLSTMs process the input sequence both forward and backward, allowing the model to capture trajectory dependencies in both directions. Figure 3 shows the architecture of the designed BiLSTM.

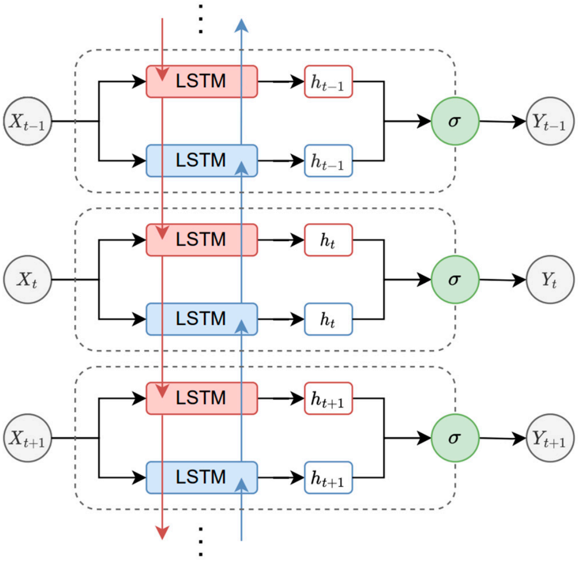

Figure 3. Proposed BiLSTM architecture for solving the ATP time-series problem.

As seen in Figure 3, within this structure, the first layer interprets the sequence from the initial to the final element (the forward direction is shown in red), while a second layer traverses in the opposite direction, from the final to the initial element (the backward direction is shown in blue). In the model, the trajectories of UAS-S4 (represented as input samples X) are fed into the LSTM layers. After being processed through the LSTM blocks and the activation functions (*h* = tanh), the model generates predicted trajectories (denoted as output samples Y). By strategically merging the outputs from both LSTM layers (using the function σ) at every time step, BiLSTMs create a more comprehensive representation of trajectories.

This synthesized representation benefits from insights drawn from both previous and future contextual trajectory sequences of the aircraft. Algorithm 2 represents the Bidirectional LSTM algorithm that was developed using the Stacked LSTM algorithm.

Algorithm 2. Bidirectional LSTM Algorithm

```
    Initialize Parameters:
1.      Define the number of layers $l$ in the LSTM.
2.      Define the number of hidden units in each layer $H$.
3.      Initialize forward and backward weights ($W_f, W_b$) and biases ($b_f, b_b$), for each layer.
    Input Preparation:
4.      Prepare input sequence $X = \{x_1,\ x_2,\ \ldots,\ x_T\}$ and standardize it.
    Procedure:
    Forward pass:
5.      For each time step, $t = 1 : T$.
6.          For each layer, $l = 1 : L$.
7.              Input gate: $i_t^{l,f} = \sigma\left(W_{ii}^{l,f} \cdot \left[h_{t-1}^{l,f}, x_t^{l,f}\right] + b_{ii}^{l,f}\right)$.
8.              Forget gate: $f_t^{l,f} = \sigma\left(W_{if}^{l,f} \cdot \left[h_{t-1}^{l,f}, x_t^{l,f}\right] + b_{if}^{l,f}\right)$.
9.              Cell candidate: $\widetilde{C}_t^{l,f} = \tanh\left(W_{ig}^{l,f} \cdot \left[h_{t-1}^{l,f}, x_t^{l,f}\right] + b_{ig}^{l,f}\right)$.
10.             Output gate: $o_t^{l,f} = \sigma\left(W_{io}^{l,f} \cdot \left[h_{t-1}^{l,f}, x_t^{l,f}\right] + b_{io}^{l,f}\right)$.
11.             Cell state update: $o_t^{l,f} = f_t^{l,f} * C_{t-1}^{l,f} + i_t^{l,f} * \widetilde{C}_t^{l,f}$.
12.             Hidden state update: $h_t^{l,f} = o_t^{l,f} * \tanh\left(C_t^{l,f}\right)$.
    Backward pass:
13.     For each time step, $t = 1 : T$.
14.         For each layer, $l = 1 : L$.
15.             Input Gate: $i_t^{l,b} = \sigma\left(W_{ii}^{l,b} \cdot \left[h_{t-1}^{l,b}, x_t^{l,b}\right] + b_{ii}^{l,b}\right)$.
16.             Forget Gate: $f_t^{l,b} = \sigma\left(W_{if}^{l,b} \cdot \left[h_{t-1}^{l,b}, x_t^{l,b}\right] + b_{if}^{l,b}\right)$.
17.             Cell Candidate: $\widetilde{C}_t^{l,b} = \tanh\left(W_{ig}^{l,b} \cdot \left[h_{t-1}^{l,b}, x_t^{l,b}\right] + b_{ig}^{l,b}\right)$.
18.             Output Gate: $o_t^{l,b} = \sigma\left(W_{io}^{l,b} \cdot \left[h_{t-1}^{l,b}, x_t^{l,b}\right] + b_{io}^{l,b}\right)$.
19.             Cell State Update: $o_t^{l,b} = f_t^{l,b} * C_{t-1}^{l,b} + i_t^{l,b} * \widetilde{C}_t^{l,b}$.
20.             Hidden State Update: $h_t^{l,b} = o_t^{l,b} * \tanh\left(C_t^{l,b}\right)$.
21.     At each time step, $t$, concatenate, $h_t^{l,f}$ and $h_t^{l,b}$, for the last layer $L$ to form the final
        output for that time step.
    Backpropagation:
22.     Compute gradients of the loss function with respect to all parameters.
23.     Update model parameters using Stochastic Gradient Descent optimizer.
    Iteration/Epoch Control:
24.     Repeat steps 5–23 for each batch of data, and for each epoch, until convergence or the
        maximum number of epochs is reached.
25. Output = $h_t^l\left[h_t^{l,f}, h_t^{l,b}\right]$.
```

The process begins with the initialization of model parameters, including the weights and biases for both the forward and backward LSTM layers. Each LSTM layer in the model consists of multiple gates (input, forget, output, and cell candidate) that control the flow of information. These gates manage how information is retained or forgotten over time, making LSTMs particularly effective for tasks where long-range dependencies are crucial.

During the forward pass of a Bidirectional LSTM, the input sequence is fed through the network in two directions: forward and backward. For each time step, the forward LSTM processes information as it appears in the sequence, while the backward LSTM processes information in reverse order. This setup results in each time step producing two sets of hidden states from both the forward and backward passes, which capture different aspects of the sequence's context. These hidden states are then typically concatenated to form a comprehensive representation of the data at each point in the sequence. This concatenated output can then be used to further process layers or directly influence the final output.

For our ATP task, where the evolution of a path is intrinsically linked to both past and future UAS-S4 trajectories, BiLSTMs present a compelling modeling choice, offering a richer, more integrated perspective on sequence data

4. Results and Discussion

The LSTM, as a deep neural network algorithm, can enhance its performance when provided with a large and diverse dataset. To create such a dataset, the UAS-S4 was utilized to generate a significant amount of aircraft trajectory data. Figure 4 illustrates the UAS-S4 Ehécatl, developed by Hydra Technologies. Table 1 represents an in-depth overview of its geometrical dimensions and flight data characteristics [53].

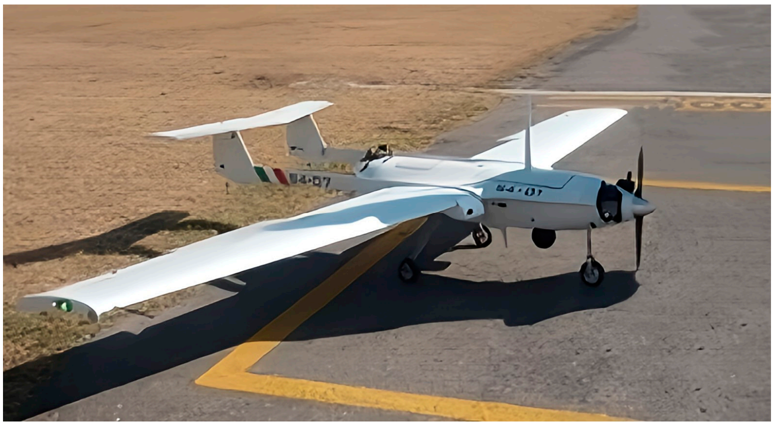

Figure 4. Hydra Technologies UAS-S4 Ehécatl.

Table 1. The UAS-S4 geometrical and flight data specification.

Specification	Value
Wingspan	4.2 m
Wing area	2.3 m^2
Total length	2.5 m
Mean aerodynamic chord	0.57 m
Empty weight	50 kg
Maximum take-off weight	80 kg
Loitering airspeed	35 knots
Maximum speed	135 knots
Service ceiling	15,000 ft
Operational range	120 km

The aircraft trajectory database was developed using a simulator that incorporated our UAS-S4 flight dynamics model, developed at Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity (LARCASE), Montreal, Canada [54–57]. This model integrates a Support Vector Regression algorithm and a resilient adaptive fuzzy controller [58]. The database contained 1820 individual trajectories, totaling 218,400 samples.

Each sample in the database was a vector comprising elements (i.e., [latitude, longitude, altitude, heading, speed, time]^T derived from GPS data.

The UAS-S4 trajectory dataset underwent standardization to be prepared for SLSTM and to enhance the model's sensitivity to input scale variance. These processed data were then divided, allocating 70% for training and 30% for testing. For compatibility with the SLSTM models, the data were reshaped into sequences defined by the 'time_steps' T parameter. The SLSTM layers were configured with 40 units each, and dropout layers with a rate of 0.2 were integrated to mitigate overfitting. The 'tanh' activation function was chosen for its superior accuracy over 'ReLU'.

Additionally, L2 regularization was employed in the SLSTM layers as a further measure against overfitting. The learning rate was determined to be 0.004, which was optimized through the Stochastic Gradient Descent (SGD) optimizer, known for its effectiveness in fitting regressors with convex loss functions. The model's training involved 40 epochs, and hyperparameters were optimized using 'KerasTuner'.

Similar to the SLSTM model, UAS-S4 trajectory data were standardized for the BiLSTM model to address its sensitivity to variations in the input scale. The standardized data were then split, with 70% allocated for training and 30% for testing. To set up the architecture, each BiLSTM layer was composed of 40 units, and dropout layers with a rate of 0.2 were employed to reduce overfitting. The 'tanh' activation function was selected due to its higher accuracy compared to 'ReLU', and L2 regularization was applied within the LSTM layers as an additional measure against overfitting. The learning rate, established at 0.004, was determined using the Stochastic Gradient Descent (SGD) optimizer, chosen for its effectiveness in handling regressors with convex loss functions. The training of the model was conducted over 40 epochs, with hyperparameters finely tuned via KerasTuner. KerasTuner involves defining the model architecture, specifying the hyperparameter search space, selecting a tuner (RandomSearch, Hyperband, and BayesianOptimization), and running the search process to find the optimal hyperparameters. This process helps in improving model performance by finding the most suitable hyperparameters for the UAS-S4 dataset and ATP problem.

In the context of BiLSTM, the chosen "merge mode" was "sum", where outputs from both the forward and backward passes of the LSTM were combined by addition. This merge mode was selected to maintain a more manageable model size while still leveraging the advantages of bidirectional processing. It proved to be especially beneficial when the forward and backward states were anticipated to be similar or to have overlapping information.

Evaluating and comparing error loss figures during LSTM-based model training is fundamental for effective model development. This process assists in performance assessment, helps to prevent overfitting, guides hyperparameter tuning, ensures model convergence, enables comparative analysis, facilitates early stopping, assists in algorithmic diagnostics, and builds confidence for model deployment. Hence, Figures 5 and 6 are represented for loss analysis when utilizing SLSTM and BiLSTM, respectively.

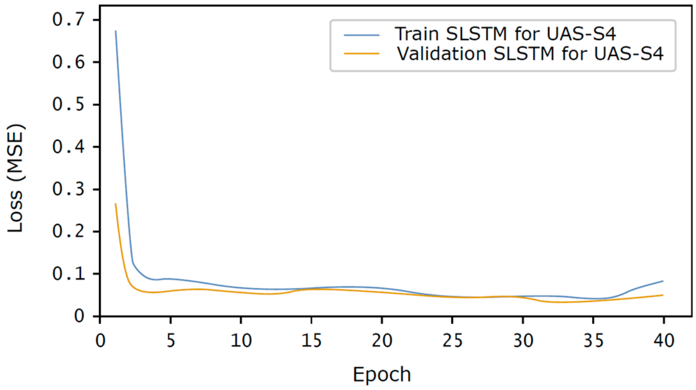

Figure 5. Mean square error loss during SLSTM training.

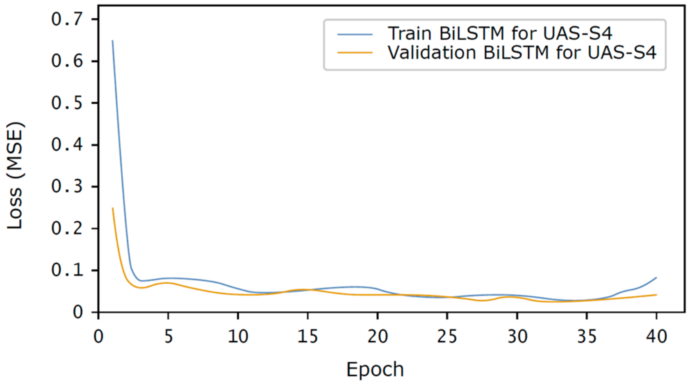

Figure 6. Mean square error loss during BiLSTM training.

In the SLSTM training graph, to solve our UAS-S4 trajectory prediction problem (Figure 5), we observed initial high loss values (0.67 for the training phase and 0.26 for the validation phase) due to the random starting weights. However, the SLSTM quickly learned, as shown by the sharp decrease in loss during the first three epochs. As training progressed beyond epoch 25, the loss reduction slowed down considerably and began to plateau, indicating that the model largely adapted to the patterns in the training data. The small fluctuations in losses are a normal part of the optimization process. There is a sign of overfitting, particularly when the validation loss starts to diverge from the training loss at epoch 38. Hence, we assumed that 27 epochs were enough for training.

In our exploration of the BiLSTM training to solve the UAS-S4 trajectory prediction problem, the initial stages were characterized by a significantly high loss (0.65 for the training phase and 0.25 for the validation phase), which is a direct consequence of the model's initial untrained state, as shown in Figure 6. However, the BiLSTM rapidly evolved, mastering the complexity of the dataset, as reflected in the steep loss reduction at the third epoch. Moving forward, as we crossed the 32nd epoch, the loss began to level off, suggesting that the model was near its learning saturation point. The graph's small fluctuations indicate the iterative nature of the BiLSTM learning process. To avoid overfitting, obtained at epoch 39, we stopped at epoch 32.

In addition to loss figures, the mean absolute percentage error (MAPE) is a robust, interpretable, and efficient metric that provides consistent and comparable evaluations of LSTM-based models for different prediction horizons. Its insensitivity to outliers and direct interpretability in data units make it a practical choice for our trajectory prediction problem. Figure 7 illustrates measured MAPEs for LSTM, SLTSM, and Bi LSTM within an 8 min prediction horizon.

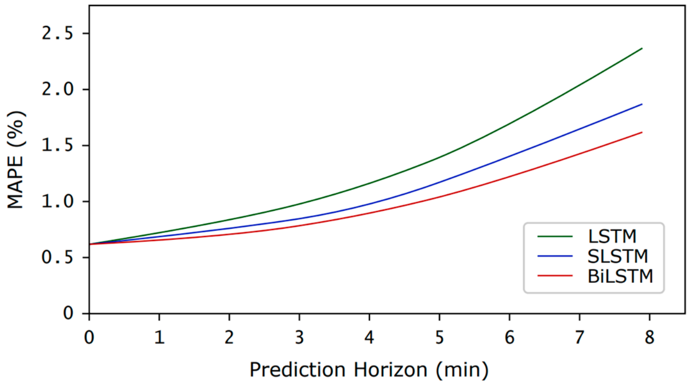

Figure 7. Trajectory prediction performance in terms of MAPE for different horizons.

As shown in Figure 7, MAPE is plotted against the prediction horizon, revealing a trend in predictive modeling. The initial low MAPE values (0.62%) indicate good accuracy for near-term forecasts. As the horizon extends, the MAPE gradually increases for all LSTM-based models, suggesting a decrease in predictive accuracy for longer-term forecasts. This increase aligns with the inherent uncertainty and complexity involved in making long-range predictions. The graph also allows us to compare different LSTM models, where BiLSTM with lower MAPE values indicates a more accurate model than the LSTM and SLSTM. Towards the longer horizons, the MAPE values start to increase exponentially, suggesting a limit to the forecasting ability of the models. For our UAS-S4s that were arranged in a range of 16 km^2, a 4 min prediction horizon was considered for trajectory prediction.

The learning rate is a pivotal hyperparameter in the LSTM models' training. It plays a significant role in the way in which a model effectively learns, as shown by its impact on the MAPE. When the learning rate is set at 0.04 (high), models can learn rapidly, which can be advantageous in the initial stages of training as it helps the model quickly approach a lower error rate. As shown in Table 2, by considering 40 layers in LSTM-based models, a 10-times increase in the learning rate caused 37, 34, and 32 min training time reductions for the LSTM, SLSTM, and BiLSTM, respectively. However, this high learning rate comes with risks, including the risk of model overshooting, leading to unstable updates and potentially higher MAPE values. As indicated in Table 2, a 10-times increase in the learning rate caused 0.14, 0.12, and 0.11% growth for MAPE for LSTM, SLSTM, and BiLSTM, respectively.

Table 2. LSTM-based prediction models' performance in terms of MAPE and training time.

	Hidden Layers	Learning Rate	MAPE %	Training Time (min)
LSTM	20	0.004	1.96	461
		0.04	2.11	425
	40	0.004	1.81	474
		0.04	1.95	437
SLSTM	20	0.004	1.55	511
		0.04	1.67	475
	40	0.004	1.41	523
		0.04	1.53	489
BiLSTM	20	0.004	1.34	567
		0.04	1.45	535
	40	0.004	1.21	581
		0.04	1.32	549

Conversely, a low learning rate ensures more stable and smaller updates to the LSTM-based model's parameters. This stability often leads to a more reliable convergence towards a lower MAPE. However, the trade-off here is the pace of learning as the model adjusts its weights only marginally in each iteration. As shown in Table 2, by considering 20 layers in LSTM-based models, a 10-times lower learning rate causes 0.15, 0.12, and 0.11 smaller MAPEs in the LSTM, SLSTM, and BiLSTM, respectively. We considered 40 layers for the LSTM-based model and set the learning rate to 0.004, which allowed the BiLSTM to learn efficiently and converge to a low MAPE (1.21%) without any issues of overshooting or stagnation. These settings have led to a better and faster convergence to a lower MAPE, blending the benefits of both high and low learning rates.

With the aim of a trade-off between training stability and learning time, we initially adopted a high learning rate at the beginning, and this decreased progressively with the epoch to enhance stability. To dynamically adjust the learning rate during training epochs within a single trial, we used '`LearningRateScheduler`' as a callback function provided by Keras. This callback was used alongside KerasTuner to fine-tune the learning rate policy as part of the model's architecture and training process. The learning rate was initially considered 0.004 and dynamically adjusted during training epochs. For the learning rate schedule function, we considered `decay_factor = 0.5`, `step_size = 10`. Table 3 shows the performance of LSTM-based prediction models while learning rates were being fine-tuned.

Table 3. LSTM-based prediction models' performance while learning rates are fine-tuned.

	Hidden Layers	MAPE %	Training Time (min)
LSTM	20	2.03	445
	40	1.88	457
SLSTM	20	1.61	494
	40	1.47	509
BiLSTM	20	1.39	554
	40	1.26	569

Following Table 3, the results confirm the superiority of BiLSTM over SLSTM and LSTM in terms of MAPE and training time, while '`LearningRateScheduler`' was responsible for fine-tuning learning rates. The callback function could provide a trade-off between MAPE and training time and achieve the desired accuracy in a reasonable training time.

Following the investigated LSTM-based models, the LSTM (as a type of recurrent neural network) is designed to overcome the vanishing gradient problem common in traditional RNNs, making it suitable for modeling trajectory sequence data. It includes memory cells with input, forget, and output gates that regulate trajectory information flow, allowing them to capture long-term dependencies more effectively.

Stacked LSTM builds on the basic LSTM architecture by layering multiple LSTM layers, each passing outputs to the layer above, allowing the network to learn at multiple levels of abstraction. This structure enables the model to handle more complex trajectory patterns and learn nuanced features from the data. However, the increased complexity raises the risk of overfitting and requires significantly more computational resources.

Bidirectional LSTM (BiLSTM) enhances traditional LSTMs by processing data in both forward and backward directions, thus gaining information from both past and future contexts. This dual-direction processing makes BiLSTM effective for trajectory prediction tasks. Despite its superior trajectory context awareness and performance on sequence data, BiLSTM also introduces more complexity and requires the full sequence before processing.

With respect to the superiority of the BiLSTM among LSTM-based architectures in terms of MAPE, this model can be compared with other deep neural networks aimed at trajectory prediction. Deep architectures such as Generative Adversarial Net [29], the autoencoder [30], and Random Forest [31] have been investigated and developed for ATP at the Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity (LARCASE).

Table 4 presents a comparative analysis of the mean absolute percentage error (MAPE%) for four different deep learning and machine learning methodologies across three prediction horizons at 1 min, 5 min, and 10 min. The purpose of this comparison is to evaluate how well each method can predict future values in a time-series, with an increasing horizon to assess their reliability over time.

Table 4. Comparison of deep neural networks performance for increasing prediction horizon.

Methodology	MAPE% along with Prediction Horizons		
	1 min	5 min	10 min
Generative Adversarial Net [29]	0.76	1.24	2.92
Autoencoder [30]	0.71	1.09	**2.46**
Random Forest [31]	**0.61**	1.14	2.75
Bidirectional LSTM	0.65	**1.02**	2.53

In accordance with Table 4, Generative Adversarial Net (GAN) shows increasing error rates as the prediction horizon extends. This trend highlights the model's challenges with maintaining accuracy over longer periods due to its inherent design, which focuses more on generating realistic data rather than forecasting. The autoencoder demonstrates the least MAPE (2.46%) among all the models over a longer-term (10 min) horizon. This indicates its potential suitability for tasks requiring the preservation of complex time-series patterns across extended durations.

Random Forest, a non-sequential model, presents the lowest MAPE (0.61%) for the short-term (1 min) horizon, illustrating its strength in capturing essential features over short periods. However, its performance degrades more noticeably than that of the BiLSTM as the prediction horizon increases, reflecting its limitations in handling temporal dependencies without specific feature engineering. Bidirectional LSTM (BiLSTM), known for its ability to capture both past and future contexts effectively, exhibits the most stable MAPE across all prediction horizons. This stability underscores its applicability to various time-series forecasting tasks, particularly where long-term dependencies are critical.

Given the superiority of the BiLSTM over the LSTM and the SLSTM in terms of prediction accuracy, it was utilized for UAS-S4 trajectory prediction. A UAS-S4 was considered in a customized 16 km^2 flying area, and an air corridor was allocated accordingly. Based on 10 actual UAS-S4 trajectories (extracted from the validation set) recorded at a 1500 m altitude, the trajectory prediction performance of the BiLSTM in terms of longitude and latitude is visualized in Figure 8.

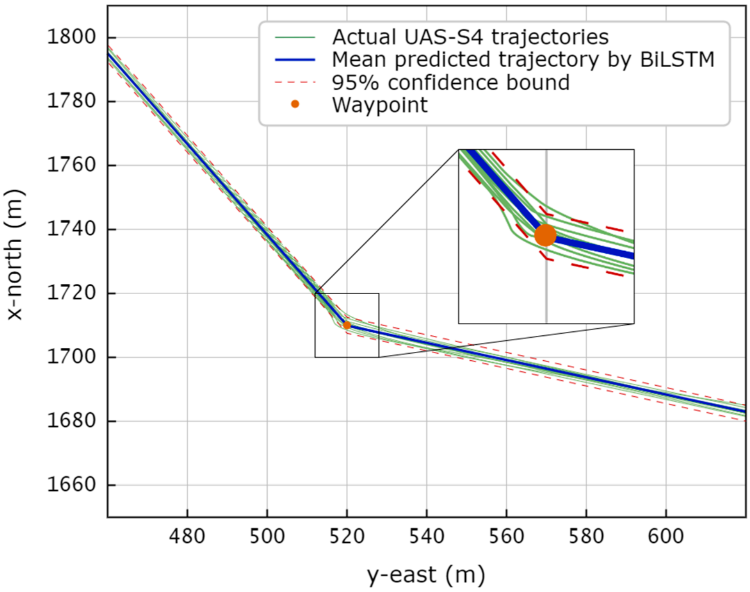

Figure 8. The BiLSTM performance for UAS-S4 trajectory prediction.

The information displayed in Figure 8 was extracted from a 16 km^2 flying area; it depicts 10 actual trajectories (green lines) of a UAS-S4 flying through an air corridor within a 95% confidence bound (red dashed lines). A waypoint was given in both coordinates: longitude = 1710 m and latitude = 520 m (the orange point). The mean predicted trajectory of the UAS-S4 given by the BiLSTM (blue lines) confirms its excellent prediction accuracy.

The BiLSTM model significantly enhances prediction accuracy in aircraft trajectory prediction, which is critical for air traffic control systems. However, the increased training time associated with BiLSTM models is a notable trade-off. In real-world scenarios, this requires a balance between computational resources and operational efficiency.

For practical implementation, especially in air traffic control systems where decision-making speed is crucial, it might be beneficial to use BiLSTM models in a hybrid approach. For instance, rapid trajectory predictions could initially be made using fewer complex models during real-time operations, with BiLSTM models being employed offline to continuously refine and update the system's understanding of complex trajectory patterns. These updates can then be cyclically integrated into the real-time system, ensuring that the model remains both accurate and efficient.

5. Conclusions

An innovative data-driven approach was designed to solve UAS-S4 Ehécatl trajectory prediction, utilizing the Long Short-Term Memory (LSTM) algorithm. Initially, a basic LSTM model was created and then evolved into a Stacked LSTM to better capture complex temporal patterns in UAS-S4′ trajectories. Subsequently, a Bidirectional LSTM was developed to achieve a dual-directional understanding of contextual trajectories, considering both past and future data for a better temporal analysis, thereby aiming to enhance prediction accuracy. The performance of these LSTM-based models was assessed using the mean absolute percentage error (MAPE) metric.

The UAS-S4 trajectories were predicted for various prediction horizons. With the extension of the prediction horizon, there was a consistent increase in the mean absolute percentage error (MAPE) across all LSTM-based models, indicating a reduced accuracy in forecasts. For the LSTM-based models, a configuration of 40 layers was chosen, along with a learning rate initially set to 0.004 and progressively fine-tuned. This setup enabled the BiLSTM model to learn effectively and ensured convergence to a low MAPE of 1.26%, thus avoiding the adverse effects of overshooting or stagnation. The results demonstrate that the BiLSTM outperformed the Stacked LSTM in accurately predicting UAS-S4 trajectories.

The primary limitation of the current study is the extended training time required for BiLSTM models, which may not be viable for all operational environments, particularly where on-the-fly retraining is necessary. Additionally, while BiLSTMs handle the dual-directional trajectory context effectively, they rely on complete sequence availability, which might not always be possible in scenarios where real-time data streaming is incomplete or delayed. Future research could focus on the exploration of real-time adaptive learning frameworks that can integrate new data into the trained model without the need for complete retraining.

Author Contributions: Conceptualization, S.M.H.; Methodology, S.M.H.; Software, S.M.H.; Validation, S.M.H., R.M.B. and G.G.; Formal analysis, S.M.H. and G.G.; Investigation, S.M.H.; Resources, R.M.B. and G.G.; Data curation, S.M.H. and G.G.; Writing—original draft, S.M.H.; Writing—review and editing, R.M.B. and G.G.; Supervision, R.M.B. and G.G.; Project administration, R.M.B.; Funding acquisition, R.M.B. and G.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSERC within the Canada Research Chairs program under the contract number: 231679, which made the realization of this research and the publication of this paper possible. Ruxandra Botez is the Canada Research Chair Tier 1 Holder in Aircraft Modeling and Simulation of New Technologies.

Data Availability Statement: Data is confidential regarding the confidential contract between LARCASE and Hydra Technologies company.

Acknowledgments: Special thanks are due to the Natural Sciences and Engineering Research Council of Canada (NSERC) for the Canada Research Chair Tier 1 in Aircraft Modeling and Simulation Technologies funds. We would also like to thank Odette Lacasse and Oscar Carranza for their support at ETS, as well as Hydra Technologies' team members Carlos Ruiz, Eduardo Yakin, and Alvaro Gutierrez Prado in Mexico. Finally, we wish to express our appreciation to the Canada Foundation for Innovation CFI, the Ministère de l'Économie et de l'Innovation, and Hydra Technologies for their support in the acquisition of the UAS-S4 at the LARCASE.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results. All authors have read and agreed to the published version of the manuscript.

References

1. Bijjahalli, S.; Sabatini, R.; Gardi, A. Advances in intelligent and autonomous navigation systems for small UAS. *Prog. Aerosp. Sci.* **2020**, *115*, 100617. [CrossRef]
2. Wargo, C.A.; Church, G.C.; Glaneueski, J.; Strout, M. Unmanned Aircraft Systems (UAS) research and future analysis. In Proceedings of the 2014 IEEE Aerospace Conference, Big Sky, MT, USA, 1–8 March 2014; pp. 1–16.
3. Madden, M.; Jordan, T.; Cotten, D.; O'Hare, N.; Pasqua, A.; Bernardes, S. *The Future of Unmanned Aerial Systems (UAS) for Monitoring Natural and Cultural Resources*; Photogrammetrie Wichmann/VDE Verlag: Belin/Offenbach, Germany, 2015; pp. 369–385.
4. Vincenzi, D.; Ison, D.C.; Terwilliger, B.A. The Role of Unmanned Aircraft Systems (UAS) in Disaster Response and Recovery Efforts: Historical, Current and Future. In Proceedings of the Association for Unmanned Vehicle Systems International, Orlando, FL, USA, 12–15 May 2014; pp. 763–783.
5. Wang, J.; Liu, Y.; Song, H. Counter-unmanned aircraft system (s)(C-UAS): State of the art, challenges, and future trends. *IEEE Aerosp. Electron. Syst. Mag.* **2021**, *36*, 4–29. [CrossRef]
6. Ghommam, J.; Saad, M.; Wright, S.; Zhu, Q.M. Relay manoeuvre based fixed-time synchronized tracking control for UAV transport system. *Aerosp. Sci. Technol.* **2020**, *103*, 105887. [CrossRef]
7. Ghommam, J.; Rahman, M.H.; Saad, M. Design of distributed event-triggered circumnavigation control of a moving target by a group of underactuated surface vessels. *Eur. J. Control* **2022**, *67*, 100702. [CrossRef]
8. Tuzcu, I.; Marzocca, P.; Cestino, E.; Romeo, G.; Frulla, G. Stability and control of a high-altitude, long-endurance UAV. *J. Guid. Control Dyn.* **2007**, *30*, 713–721. [CrossRef]
9. Ghommam, J.; Saad, M.; Mnif, F.; Zhu, Q.M. Guaranteed performance design for formation tracking and collision avoidance of multiple USVs with disturbances and unmodeled dynamics. *IEEE Syst. J.* **2020**, *15*, 4346–4357. [CrossRef]
10. Romeo, G.; Borello, F.; Cestino, E.; Moraglio, I.; Novarese, C. ENFICA-FC: Environmental Friendly Inter-City Aircraft and 2-seat aircraft powered by Fuel Cells electric propulsion. In Proceedings of the Airtec 2nd International Conference "Supply on the Wings", Frankfurt, Germany, 15–17 April 2007; pp. 24–25.
11. Hashemi, S.M.; Hashemi, S.A.; Botez, R.M. Reliable Aircraft Trajectory Prediction Using Autoencoder Secured with P2P Blockchain. In Proceedings of the International Symposium on Unmanned Systems and the Defense Industry, Madrid, Spain, 30 May–1 June 2022; pp. 271–275.
12. Hashemi, S.M. *Novel Trajectory Prediction and Flight Dynamics Modelling and Control Based on Robust Artificial Intelligence Algorithms for the UAS-S4*; École de Technologie Supérieure: Montreal, QC, Canada, 2022.
13. Izadi, H.; Gordon, B.; Zhang, Y. Safe path planning in the presence of large communication delays using tube model predictive control. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, ON, Canada, 2–5 August 2010; p. 8425.
14. Zhou, X.; Yu, X.; Guo, K.; Zhou, S.; Guo, L.; Zhang, Y.; Peng, X. Safety flight control design of a quadrotor UAV with capability analysis. *IEEE Trans. Cybern.* **2021**, *53*, 1738–1751. [CrossRef] [PubMed]
15. Zhou, X.; Yu, X.; Zhang, Y.; Luo, Y.; Peng, X. Trajectory planning and tracking strategy applied to an unmanned ground vehicle in the presence of obstacles. *IEEE Trans. Autom. Sci. Eng.* **2020**, *18*, 1575–1589. [CrossRef]
16. Leon, F.; Gavrilescu, M. A review of tracking and trajectory prediction methods for autonomous driving. *Mathematics* **2021**, *9*, 660. [CrossRef]
17. Hashemi, S.M.; Botez, R.M.; Ghazi, G. Blockchain PoS and PoW consensus algorithms for airspace management application to the UAS-S4 Ehécatl. *Algorithms* **2023**, *16*, 472. [CrossRef]
18. Szymanski, M.; Ghazi, G.; Botez, R.M. Development of a Map-Matching Algorithm for the Analysis of Aircraft Ground Trajectories using ADS-B Data. In Proceedings of the AIAA AVIATION 2023 Forum, San Diego, CA, USA, 12–16 June 2023; p. 3758.
19. Ghazi, G.; Botez, R.M. Aircraft mathematical model identification for flight trajectories and performance analysis in cruise. *J. Aerosp. Inf. Syst.* **2022**, *19*, 530–549. [CrossRef]
20. Wiest, J.; Höffken, M.; Kreßel, U.; Dietmayer, K. Probabilistic trajectory prediction with Gaussian mixture models. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, 3–7 June 2012; pp. 141–146.
21. Korbmacher, R.; Tordeux, A. Review of pedestrian trajectory prediction methods: Comparing deep learning and knowledge-based approaches. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 24126–24144. [CrossRef]
22. Ghazi, G.; Botez, R.M.; Bourrely, C.; Turculet, A.-A. Method for calculating aircraft flight trajectories in presence of winds. *J. Aerosp. Inf. Syst.* **2021**, *18*, 442–463. [CrossRef]
23. Petrou, P.; Nikitopoulos, P.; Tampakis, P.; Glenis, A.; Koutroumanis, N.; Santipantakis, G.M.; Patroumpas, K.; Vlachou, A.; Georgiou, H.; Chondrodima, E. ARGO: A big data framework for online trajectory prediction. In Proceedings of the 16th International Symposium on Spatial and Temporal Databases, Vienna, Austria, 19–21 August 2019; pp. 194–197.
24. Zwick, M.; Gerdts, M.; Stütz, P. Sensor-Model-Based Trajectory Optimization for UAVs to Enhance Detection Performance: An Optimal Control Approach and Experimental Results. *Sensors* **2023**, *23*, 664. [CrossRef]
25. Machin, M.; Sanguesa, J.A.; Garrido, P.; Martinez, F.J. On the use of artificial intelligence techniques in intelligent transportation systems. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Barcelona, Spain, 15–18 April 2018; pp. 332–337.

26. Jiang, H.; Chang, L.; Li, Q.; Chen, D. Trajectory prediction of vehicles based on deep learning. In Proceedings of the 2019 4th International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 5–7 September 2019; pp. 190–195.
27. Hashemi, S.M.; Botez, R.M.; Grigorie, T.L. New reliability studies of data-driven aircraft trajectory prediction. *Aerospace* **2020**, *7*, 145. [CrossRef]
28. Nikhil, N.; Tran Morris, B. Convolutional neural network for trajectory prediction. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
29. Hashemi, S.M.; Hashemi, S.A.; Botez, R.M.; Ghazi, G. Aircraft trajectory prediction enhanced through resilient generative adversarial networks secured by blockchain: Application to UAS-S4 Ehécatl. *Appl. Sci.* **2023**, *13*, 9503. [CrossRef]
30. Hashemi, S.M.; Hashemi, S.A.; Botez, R.M.; Ghazi, G. A novel fault-tolerant air traffic management methodology using autoencoder and P2P blockchain consensus protocol. *Aerospace* **2023**, *10*, 357. [CrossRef]
31. Hashemi, S.M.; Botez, R.M.; Ghazi, G. Robust trajectory prediction using random forest methodology application to UAS-S4 ehécatl. *Aerospace* **2024**, *11*, 49. [CrossRef]
32. Suo, Y.; Chen, W.; Claramunt, C.; Yang, S. A ship trajectory prediction framework based on a recurrent neural network. *Sensors* **2020**, *20*, 5133. [CrossRef]
33. Chandra, R.; Goyal, S.; Gupta, R. Evaluation of deep learning models for multi-step ahead time series prediction. *IEEE Access* **2021**, *9*, 83105–83123. [CrossRef]
34. Shi, Z.; Xu, M.; Pan, Q.; Yan, B.; Zhang, H. LSTM-based flight trajectory prediction. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
35. Hou, L.; Xin, L.; Li, S.E.; Cheng, B.; Wang, W. Interactive trajectory prediction of surrounding road users for autonomous driving using structural-LSTM network. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 4615–4625. [CrossRef]
36. Lin, K.; Peng, J.; Gu, F.L.; Lan, Z. Simulation of open quantum dynamics with bootstrap-based long short-term memory recurrent neural network. *J. Phys. Chem. Lett.* **2021**, *12*, 10225–10234. [CrossRef] [PubMed]
37. Venskus, J.; Treigys, P.; Markevičiūtė, J. Unsupervised marine vessel trajectory prediction using LSTM network and wild bootstrapping techniques. *Nonlinear Anal. Model. Control.* **2021**, *26*, 718–737. [CrossRef]
38. Nicola, F.; Fujimoto, Y.; Oboe, R. A LSTM neural network applied to mobile robots path planning. In Proceedings of the 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), Porto, Portugal, 18–20 July 2018; pp. 349–354.
39. Fernando, T.; Denman, S.; Sridharan, S.; Fookes, C. Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *Neural Netw.* **2018**, *108*, 466–478. [CrossRef] [PubMed]
40. Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; Savarese, S. Social lstm: Human trajectory prediction in crowded spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 961–971.
41. Shi, Z.; Xu, M.; Pan, Q. 4-D flight trajectory prediction with constrained LSTM network. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 7242–7255. [CrossRef]
42. Dong, L.; Fang, D.; Wang, X.; Wei, W.; Damaševičius, R.; Scherer, R.; Woźniak, M. Prediction of streamflow based on dynamic sliding window LSTM. *Water* **2020**, *12*, 3032. [CrossRef]
43. Zhao, Z.; Zeng, W.; Quan, Z.; Chen, M.; Yang, Z. Aircraft trajectory prediction using deep long short-term memory networks. In Proceedings of the CICTP 2019, Nanjing, China, 6–8 July 2019; pp. 124–135.
44. van Iersel, Q.G.; Murrieta Mendoza, A.; Felix Patron, R.S.; Hashemi, S.M.; Botez, R.M. Attack and Defense on Aircraft Trajectory Prediction Algorithms. In Proceedings of the AIAA AVIATION 2022 Forum, Chicago, IL, USA, 27 June–1 July 2022; p. 4027.
45. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
46. Li, Y.; Bao, T.; Gong, J.; Shu, X.; Zhang, K. The prediction of dam displacement time series using STL, extra-trees, and stacked LSTM neural network. *IEEE Access* **2020**, *8*, 94440–94452. [CrossRef]
47. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
48. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
49. Wang, C.; Ma, L.; Li, R.; Durrani, T.S.; Zhang, H. Exploring trajectory prediction through machine learning methods. *IEEE Access* **2019**, *7*, 101441–101452. [CrossRef]
50. Hashemi, S.; Hashemi, S.A.; Botez, R.M.; Ghazi, G. A novel air traffic management and control methodology using fault-tolerant autoencoder and P2P blockchain application on the UAS-S4 ehécatl. In Proceedings of the AIAA SCITECH 2023 Forum, National Harbor, MD, USA, 23–27 January 2023; p. 2190.
51. Li, W.; Qi, F.; Tang, M.; Yu, Z. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. *Neurocomputing* **2020**, *387*, 63–77. [CrossRef]
52. Du, X.; Zhang, H.; Van Nguyen, H.; Han, Z. Stacked LSTM deep learning model for traffic prediction in vehicle-to-vehicle communication. In Proceedings of the 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, Canada, 24–27 September 2017; pp. 1–5.
53. Botez, R.M. Editorial for the special issue "Aircraft modeling and simulation". *Appl. Sci.* **2022**, *12*, 1234. [CrossRef]
54. Hashemi, S.M.; Botez, R.M. A Novel Flight Dynamics Modeling Using Robust Support Vector Regression against Adversarial Attacks. *SAE Int. J. Aerosp.* **2023**, *16*, 305–323. [CrossRef]

55. Kuitche, M.A.J.; Botez, R.M. Modeling novel methodologies for unmanned aerial systems–Applications to the UAS-S4 Ehecatl and the UAS-S45 Bálaam. *Chin. J. Aeronaut.* **2019**, *32*, 58–77. [CrossRef]
56. Kuitche, M.; Botez, R.M. Methodology of estimation of aerodynamic coefficients of the UAS-E4 Ehécatl using datcom and VLM procedure. In Proceedings of the AIAA Modeling and Simulation Technologies Conference, Grapevine, TX, USA, 9–13 January 2017; p. 3152.
57. Kuitche, M.A.J.; Botez, R.M.; Guillemin, A.; Communier, D. Aerodynamic modelling of unmanned aerial system through nonlinear vortex lattice method, computational fluid dynamics and experimental validation-application to the uas-s45 bàlaam: Part 1. *INCAS Bull.* **2020**, *12*, 91–103. [CrossRef]
58. Hashemi, S.; Botez, R. Lyapunov-based robust adaptive configuration of the UAS-S4 flight dynamics fuzzy controller. *Aeronaut. J.* **2022**, *126*, 1187–1209. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.