Assessing the drone threat to aviation security

Florin Daniel SIMION*[,a]

*Corresponding author
Faculty of Mechanical Engineering and Mechatronic,
National University of Science and Technology POLITEHNICA Bucharest,
Splaiul Independentei 313, 060042, Bucharest, Romania,
simion.dan@hotmail.com

DOI: 10.13111/2066-8201.2025.17.3.8

Received: 11 August 2025/ Accepted: 25 August 2025/ Published: September 2025

Abstract: *The safety of airport operations is increasingly threatened by the proliferation of civilian drones that can enter unauthorised airport airspace. This article analyses the risks associated with drones in civil aviation security from both a theoretical and a practical perspective, drawing on recent literature and risk prediction models. In the first part, we review the rapid growth in the number of recreational drones being sold and their implications, including the possibility of terrorist groups easily accessing this technology. In the methodology part, the autoregressive (AR), moving average (MA), autoregressive moving average (ARMA), and long-short-term memory (LSTM) neural networks used for risk modeling and forecasting are presented, along with the key mathematical formulas and advantages of each model. Next, a simulation on synthetic data is performed, demonstrating the practical application of these models in drone incident prediction, how to select the optimal parameters, and how to evaluate the performance using multiple error metrics - Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Normalized RMSE (NRMSE). The latter is used to provide a scale-independent assessment of model accuracy, enabling fairer and more interpretable comparisons. The results obtained highlight the potential of the LSTM model to deliver the most accurate and stable predictions, particularly in scenarios involving nonlinear temporal patterns. The conclusions underscore the practical relevance of this approach and recommend its application in real airport security scenarios, where proactive risk forecasting can support more efficient resource allocation and mitigation planning.*

Key Words: *UAV, risk forecasting, LSTM, ARMA, airport security*

1. INTRODUCTION

Recent technological developments have led to increased accessibility and popularity of recreational drones worldwide [14]. Due to their ease of operation and relatively low cost, the barrier to entry for users is very low, increasing the number of drones in circulation [1].

This upward trend, while beneficial for innovation and various commercial and recreational applications, poses significant risks to civil aviation safety. Incidents involving drones entering the vicinity of airports are increasingly reported around the world, causing disruptions to air traffic. In many cases, the unauthorized presence of a drone in the controlled airspace of an airport has led to massive flight delays and even temporary runway closures.

[a] Doctoral student

A notorious incident occurred at Heathrow Airport in April 2016, when a drone was suspected of colliding with an Airbus A320 aircraft during landing; although the incident was later clarified, it highlighted the potentially disastrous consequences of a drone-aircraft collision. According to BBC reports [2], the 2018 incident at Gatwick Airport highlighted the extent of disruption that drones can cause to air traffic.

In addition to careless users, there is also the threat of deliberate action: easy access to advanced drones allows terrorist or criminal groups to use these devices for hostile purposes. Studies show a substantial increase in the sophistication and availability of drone technologies in the civilian market, making them attractive to malicious actors [2].

Drones thus offer new aerial capabilities to terrorists without requiring traditional aviation infrastructure, making them ideal for asymmetric warfare. For example, terrorist organizations have adapted consumer drones for surveillance or even to carry improvised explosive devices, extending their range and danger to civilian targets. This democratization of aviation technology allows drones to be increasingly acquired and integrated by violent groups into a wide range of nonlethal and lethal operations [3].

As a result, future technological developments could significantly improve terrorist aerial capabilities, increasing the level of threat. In response to these emerging risks, the relevant authorities have taken steps to assess and counter the threat posed by drones. At the European level, the European Union Aviation Safety Agency (EASA) recognized the danger and established a dedicated task force (Drone Collision Task Force) in 2016 to investigate the risks of drone-aircraft collisions and identify research needs [4].

The report of this working group formed the basis for subsequent research projects funded by the European Commission, aimed at quantifying the potential severity of collisions between different types of drones and aircrafts. At the same time, EASA issued guidelines for the management of drone incidents in aerodromes, promoting a coordinated ”defense-in-depth” approach to prevent, detect and neutralize unauthorized drones within airport perimeters [5].

Internationally, aviation authorities (FAA in the US, etc.) have tightened regulations on drone operations in the vicinity of airports and invested in anti-drone systems. However, incidents continue to multiply. A recent study by Pyrgies (2019) inventoried 139 serious drone incidents in the vicinity of airports around the world, concluding that they are more numerous than anticipated and occur at higher altitudes and distances from airports than expected, including in terminal control areas (TMAs) tens of kilometers from the airport [6].

These findings suggest that the risk posed by drones goes beyond classic threat scenarios, requiring new tools for risk analysis and prediction. To date, the literature has mainly addressed the threat posed by drones in terms of accident/incident analysis and qualitative risk assessment. Wild et al. (2016) analyzed 152 RPAS (Remotely Piloted Aircraft Systems) events between 2006 and 2015 to identify incident types and contributing factors, highlighting major differences from traditional commercial aviation [7].

Pyrgies [6] applied the FAA’s formal Safety Risk Management process to determine the severity and probability of hazards associated with drones at airports, while also proposing risk mitigation measures. Similarly, Zhang et al. (2020) proposed a stochastic model for assessing the probability of collision between an intruding drone and a commercial aircraft, demonstrating through simulations that risks can be quantified even with limited information about the drone’s trajectory [1]. The conclusions of these studies highlight the need for more advanced methods of anticipating dangerous drone events. Proactive risk prediction - for example, estimating the probability or frequency of drone incursions over a time horizon - would allow airport authorities to take preventive measures (such as raising the alert level, activating countermeasure systems, or informing pilots) before an incident escalates.

However, there is a gap in the current literature regarding quantitative predictive modeling of drone-related risks, particularly in the context of airport operational forecasting. While most studies adopt qualitative assessments or focus on stochastic collision simulations, few - if any - attempt to employ sequential models such as ARMA or LSTM for proactive risk prediction.

Original contribution. This paper proposes a novel modeling framework for forecasting the monthly evolution of drone-related risks near airports, using both classical autoregressive methods and LSTM neural networks. To our knowledge, this is the first structured application of LSTM in the context of UAV threats to aviation security, benchmarked against AR, MA, and ARMA models over synthetic time series data. The use of Normalized RMSE (NRMSE) further ensures a robust, scale-independent performance evaluation. The approach contributes both methodological innovation and empirical validation for integrating predictive tools into airport security decision-making processes.

2. METODOLOGY

To quantify and forecast the risk of drone incursions in airport space, we utilized time series mathematical models and machine learning algorithms specialized in modeling sequential data. This section describes in turn the classical autoregressive models (AR, MA, ARMA) and the Long Short-Term Memory (LSTM) neural network model, highlighting the basic mathematical formulations, implicit assumptions, advantages of each, and how they can be used in the context of drone risk analysis.

The choice of these models is motivated by their proven ability to capture temporal dependencies in sequential data. Classical statistical models like AR, MA, and ARMA have been widely used in operational forecasting across security, traffic, and infrastructure domains, where risk evolves over time with short-term memory effects [14, 15]. Their transparency and ease of implementation make them attractive for real-time monitoring systems, especially in regulated environments like aviation. In contrast, neural networks such as LSTM allow for the modeling of nonlinear, long-range dependencies and adapt well to irregular patterns—features often observed in incident data involving UAVs [13, 16].

Recent studies demonstrate that LSTM-based approaches can outperform traditional models in security forecasting applications, including anomaly detection in restricted airspace, early warning systems, and adaptive control [12, 13].

By combining traditional and modern approaches, we aim to explore whether simple models suffice in practice, or whether the added complexity of deep learning architectures brings measurable improvements. The comparison is performed under controlled conditions, using synthetic time series reflecting realistic operational scenarios, thus enabling a fair performance assessment across models.

This methodological framework allows us to simulate, evaluate, and compare predictive capabilities relevant to drone risk forecasting, while preserving both interpretability (via AR/MA/ARMA) and adaptability to complex patterns (via LSTM).

2.1 The Autoregressive (AR) model

An autoregressive model of order p, denoted AR(p), expresses the current value of a time series (X_t) as a function of its immediate past values. In its general form, an AR(p) model can be written as:

$$X_t = c + \varphi_1 X(t-1) + \varphi_2 X(t-2) + \ldots + \varphi_p X(t-p) + \varepsilon_t \tag{1}$$

where (c) is a constant (the process mean), the coefficients ($\varphi_1, \varphi_2, ..., \varphi_p$) indicate the influence of each past value, and (ε_t) is the error term at time (t), usually assumed to be "white noise" with zero mean and constant variance. Intuitively, an AR model estimates that the series has finite memory: the last (p) previous values contain all the necessary information to predict the near future [8].

The stationarity condition of the process requires that the roots of the characteristic polynomial

$$1 - \varphi_1 z - ... - \varphi_p z^p = 0 \tag{2}$$

must satisfy $|\varphi_i| < 1$ in simple cases (for example, for AR(1), it is necessary that $|\varphi_1| < 1$) [8].

Advantages of AR: The model is relatively simple and interpretable the coefficients indicate the degree of dependence on past values, and parameter estimation can be efficiently performed using the least squares method. AR models effectively capture short-term trends and system inertia; for example, if many drone incursions are recorded in one month, an AR model can incorporate this information and increase the estimated probability for the following month. In the context of airport risk assessment, an AR model may be suitable if the frequency of drone incidents follows an autoregressive behavior (either due to short-term seasonality or because of authorities' reactions - e.g., a recent incursion may temporarily increase vigilance and reduce the immediate likelihood of another, or conversely, it may indicate a pattern that tends to repeat).

However, the AR model assumes linearity and stationarity. If the event series exhibits a trend (e.g., an increasing number of incidents annually as more drones are used) or pronounced seasonality (e.g., peaks in summer), the simple AR model needs to be extended—usually by applying data differencing (resulting in ARIMA models) or by adding seasonal terms. In this work, we will limit ourselves to the hypothesis of local stationarity over the analyzed interval, with trends addressed separately if necessary. Additionally, selecting the order p is critical: a too small p may omit important dependencies, while a too large p risks overfitting. In practice, the partial autocorrelation function (PACF) and the Akaike Information Criterion (AIC) are used to determine an optimal p [8].

$$\text{AIC} = -2 \times \ln(L) + 2k \tag{3}$$

where L is the likelihood function of the model, which measures how well the model fits the observed data, and k is the number of parameters in the model [6].

$$L(\theta|y_i) = \prod_{i=1}^{n} f(y_i\,|\theta) \tag{4}$$

where $L(\theta|y_i)$ depends on the parameters θ and the observed data (y_i), and the product $\prod_{i=1}^{n} f(y_i\,|\theta)$ signifies the product of all values $f(y_i|\theta)$ for each i. This is the probability density (or probability mass) function of the data y_i, given the parameters θ [8].

2.2 The Moving Average (MA) model

The moving average model of order q, denoted $MA(q)$, represents the series as a weighted sum of recent shocks (errors). In mathematical form:

$$X_t = \mu + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + ... + \theta_q\varepsilon_{t-q} \tag{5}$$

where (μ) is the mean (constant level) of the process, and ($\varepsilon_{t-1},\varepsilon_{t-2},...,\varepsilon_{t-q}$) are independent, identically distributed error terms affecting the system at those moments [8]. The coefficients ($\theta_1,...,\theta_q$) indicate the influence of past shocks on the current value.

Unlike the AR model, which uses past values of the series, the MA model uses past errors: a simple example, MA(1), has the form:

$$X_t = \mu + \varepsilon_t + \theta_1\varepsilon_{t-1} \tag{6}$$

this means that an unexpected disturbance at time (t - 1) (e.g., a rare event - in our context, an unexpected security incident) may influence the value at time t (for example, the number of incidents in the next month could be affected by the previous month's incident through a residual effect) [8].

Advantages of MA: It can efficiently model processes in which the effect of shocks diminishes relatively quickly. In the context of drone risks, if response measures to an incident (a shock) are intense immediately afterward - for example, after a major incident, security measures are temporarily increased, reducing the likelihood of another incident in a short period - then an MA model could capture this sudden decrease in risk later on. Additionally, MA models are finite (of order (q)) and always stationary (they do not require special stationarity conditions on the coefficients, unlike AR models).

A limitation of the MA model is that long-term relationships (more than q periods) cannot be directly captured because the influence of any shock completely disappears after q lags in the model. If the series of incidents exhibits long-term dependencies, a pure MA model may not be appropriate. The choice of q is typically based on the Akaike Information Criterion (AIC) or the autocorrelation function (ACF): an ACF that shows a limited number of significant lags followed by values close to zero suggests an MA model, with the number of significant lags indicating q [8].

2.3 The Autoregressive Moving Average (ARMA) model

The ARMA (Autoregressive Moving Average) model combines the autoregressive and moving average components, providing a more flexible framework for modeling stationary series. An ARMA of order (p,q) lends the two previous relationships as follows:

$$X_t = c + \varphi_1 X_{t-1} + ... + \varphi_p X_{t-p} + \varepsilon_t + \theta_1\varepsilon_{t-1} + ... + \theta_q\varepsilon_{t-q} \tag{7}$$

thus, $ARMA(p,q)$ can represent short-term behaviors through the MA component and longer dependencies via the AR part [8]. In practice, many time series can be adequately modeled with a low-order ARMA, which is why classical time series analysis methodologies focus on correctly identifying the parameters p and q that strike the best balance between model complexity and autocorrelation decay speed.

Model selection between AR/MA vs. ARMA: The analysis of correlograms provides clues: if the autocorrelation function (ACF) gradually decreases exponentially, while the partial autocorrelation function (PACF) has a finite number of significant peaks, this suggests a predominantly AR model; conversely, if the ACF has a finite number of peaks and the PACF decays exponentially, it indicates an MA model. When both ACF and PACF display gradual declines or more complex patterns, an ARMA model (a combination of the two) is necessary to explain the correlations in the data.

In the case of non-stationary series (where the ACF does not tend toward zero even for large lags), differencing is applied and the model is extended to ARIMA. However, as mentioned, in the current analysis we will assume the input data are stationary (for example, focusing on deviations from a known mean or trend).

The main advantage of the ARMA model is its versatility: it can represent a wide range of behaviors by simply adjusting the two parameters p and q. Moreover, if the true underlying structure of the series is AR or MA, a wellcalibrated ARMA model can approximate it very accurately (for example, a purely AR series can be captured by an ARMA where $q = 0$, and similarly, a purely MA series with $p = 0$). In the context of drone risks, if the frequency of incidents or a composite risk index exhibits both persistence (e.g., a high risk level in one month increases the chances that the following month will also be risky) and transient effects (e.g., a sudden implementation of countermeasures reduces risk only in the short term), then an ARMA model can incorporate both aspects simultaneously.

2.4 The Long Short-Term Memory (LSTM) neural network model

The previously discussed autoregressive models are all fixed linear parametric models, suitable for stationary series and direct relationships between variables. However, in practice, the evolution of risks may be influenced by complex and nonlinear factors - such as the interaction between drone traffic density in the airspace, weather conditions affecting drone detection, the level of active anti-drone measures, and so forth. To capture such relationships and leverage larger datasets, models based on recurrent neural networks are used, particularly the Long Short-Term Memory (LSTM) architecture [9].

LSTM (Long Short-Term Memory) is a special type of recurrent neural network (RNN) introduced by Hochreiter and Schmidhuber (1997) [10], designed to address the problem of long-term memory loss in traditional RNNs. Through an internal input-output mechanism, LSTM can learn dependencies over long periods, managing to retain important past information while ignoring noise or irrelevant data.

At each time step t, an LSTM cell receives an input vector x_t (which may include the time series values at t and possibly other exogenous factors) and produces a hidden state h_t (serving as the network's output, such as the predicted value), while updating its internal memory state C_t. The update process involves three sigmoidal gates and an intermediate state, as follows:

Forget Gate

$$f_t = \sigma\,(W_f\,[h_{t-1},x_t] + b_f) \tag{8}$$

this gate determines which parts of the previous cell state C_{t-1} are retained (value close to 1) or forgotten (value close to 0) [9]. The sigmoid function, as used in neural networks [10], is a non-linear "S"- shaped function that transforms any real value into a range between 0 and 1. Its formula is:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{9}$$

where x is the input to the sigmoid function, and e is Euler's number (approximately 2.71828):

Input Gate

$$i_t = \sigma\,(W_i[h_{t-1},x_t] + b_i) \tag{10}$$

controls how much of the candidate new information $\tilde{C}_t$ will be added to the cell state, where:

$$\{\tilde{C}\}_t = \tanh(W_C[h_{t-1}, x_t] + b_c) \tag{11}$$

$\tilde{C}_t$ is a candidate state computed similarly to a standard RNN [9]. The hyperbolic tangent function (tanh) is another non-linear function with a similar shape to sigmoid but outputs values between -1 and 1. Its formula is:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{12}$$

where *x* is the input to tanh, and *e* is Euler's number.

Memory Cell State

The cell state is updated by combining the old state and the candidate, via element-wise multiplication:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{13}$$

where ⊙ denotes element-wise product [8].

Output Gate

$$C_t = f_t \odot C_{\{t-1\}} + i_t \odot \{\widetilde{C}\}_t \tag{14}$$

controls how much of the internal memory C_t is reflected in the hidden state (output). The final hidden state becomes

$$h_t = o_t \odot \tanh(C_t) \tag{15}$$

this hidden state h_t serves as the output for the current time step, as well as the input to the next step [9].

Advantages of LSTM: Thanks to its memory mechanism, LSTM can better capture long-term dependencies compared to AR/MA/ARMA models or simple RNNs. In time series prediction tasks, LSTM has become a benchmark for performance across various applications, from speech recognition to traffic modeling, precisely because it can model complex and nonlinear relationships in data. [9], [11]

In the case of drone-related risks, an LSTM could, for example, learn complex patterns: it can identify annual seasonality (e.g., incident peaks in summer), correlate fluctuations in incidents with other contextual variables (such as monthly drone sales, legislative conditions if provided), and respond to subtle signals in past dynamics that linear models might overlook. Another advantage is that LSTM does not require the series to be stationary - the network can learn and capture overall trends or seasonality directly from raw data without explicit transformations like differencing.

Challenges in using LSTM: On the other hand, LSTM networks have many parameters, which require a larger volume of data for training and pose a risk of overfitting if the architecture is not properly tuned. Training typically involves algorithms like backpropagation through time (BPTT) and can be computationally intensive due to the sequential nature of the data and the need to compute gradients at each time step for long sequences. The choice of parameters (such as the number of LSTM cells, number of layers, learning rate, input window size, etc.) is often empirical and requires experimentation.

Additionally, the results of a neural network are less interpretable compared to statistical models - for example, they do not directly provide coefficients indicating simple causal relationships. Nonetheless, when prediction accuracy is critical, LSTMs tend to deliver superior performance, which is why we will explore their applicability in our simulation.

To apply the above models to the drone risk problem, we will generate a time series quantifying the risk level or incident frequency within a regular interval (for example, the number of drone-related incidents reported monthly at a certain airport or a risk score computed quarterly based on various factors).

3. TRAINING THE MODELS

To operationalize the described forecasting models, a simulation environment was implemented in Python. The logical flow of this implementation is illustrated in Figure 1, which summarizes the key computational steps: synthetic data generation, model training, prediction, and performance evaluation.

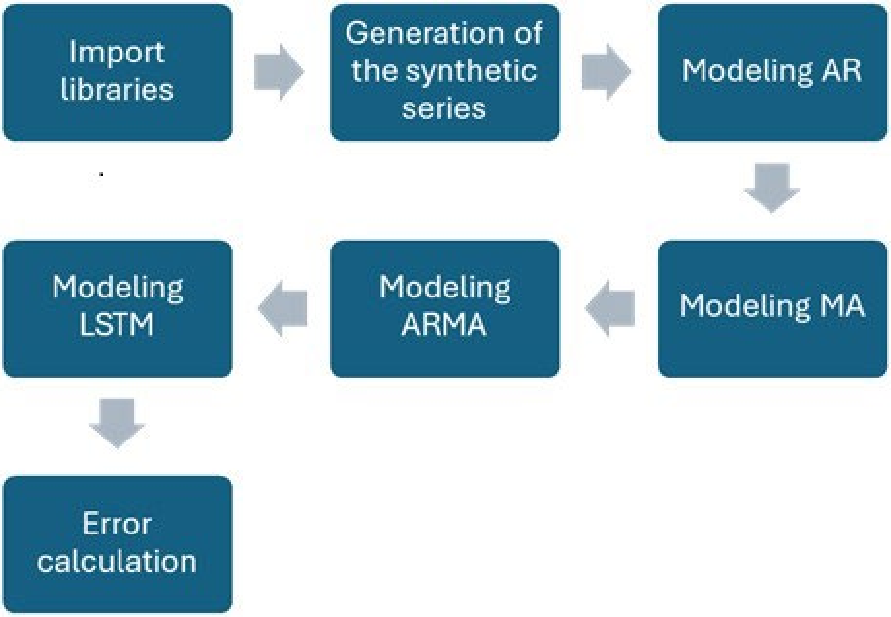

Figure 1: Logical diagram of the Python code

As illustrated, each model follows a consistent structure in its application—from input preparation to output evaluation—allowing for a controlled comparison of forecasting accuracy across different methodologies.

The following sections present the steps for running the code and performing the corresponding calculations, noting that each execution of the code generates a new dataset due to implicit randomness. Although the numeric values may vary slightly between runs, the overall trend of the results remains consistent. For clarity and practical applicability, some calculations have been deliberately simplified, as certain operations - especially those related to the LSTM model - cannot be fully reproduced manually as they are processed by Python libraries.

3.1 Generation of the synthetic series

To illustrate the practical application of the discussed models, a synthetic time series was generated representing a monthly risk index (or number of drone incidents) over several years.

The series was constructed to exhibit stationary behavior but with short-term dependencies, making it suitable for approximation with an ARMA(1,1) model (an autoregressive model of order 1 combined with a moving average of order 1).

Although, in reality, the number of incidents is a count variable (an integer and positive), the series generated via the ARMA model is continuous and contains real-valued data. This intentionally reflects an operational risk index rather than an actual incident counter. Using continuous values allows for more refined trend analysis and facilitates the application of numerical models such as LSTM, which require normalized and dense data. In an operational context, these values can later be discretized or calibrated to correspond to alert thresholds or pragmatic risk interpretations.

Such characteristics could correspond to a scenario where the risk in a given month is moderately dependent on the level in the previous month (autoregressive component) and influenced by a transient shock (for example, a recent alert or incident that has immediate effects but does not last long, captured by the MA component).

In this stage, an artificial dataset is constructed to simulate the monthly evolution of a risk index associated with drone operations near airports. To reflect both the characteristics of temporal dependence and the effects of unexpected shocks, the series is generated using a stochastic ARMA(1,1) model. This model relies on two key parameters: the autoregressive coefficient $\varphi = 0.6$, which measures the influence of the previous value on the current one, and the moving average coefficient $\theta = 0.5$, which quantifies the impact of past shocks.

The value $\varphi = 0.6$ indicates a moderate dependence on previous values, suggesting that risk levels in one month significantly, but not definitively, influence the following month—an often observed feature in security processes exhibiting operational inertia. The value $\theta = 0.5$ captures the effects of temporary shocks, such as an isolated event (e.g., a serious drone intrusion) that immediately impacts the risk level, but whose effect dissipates quickly.

Starting from formula (7), which defines an ARMA(1,1) model with initial values $X^0 = 0.4967$, $\varepsilon^0 = 0.6477$, and $\varepsilon^1 = -0.2342$ generated randomly, we obtain:

$$X^1 = 0.6 \times 0.4967 + (-0.2342) + 0.5 \times 0.6477 \tag{16}$$

Calculating step by step, we have $0.6 \times 0.4967 = 0.2980$ and $0.5 \times 0.6477 = 0.3239$, which leads to:

$$X^1 = 0.2980 - 0.2342 + 0.3239 = 0.3877 \tag{17}$$

In this modeling, the constant term c in the general ARMA form was assumed to be zero, considering that the simulated series is centered around zero. The focus of the analysis is solely on the dynamics of fluctuations and the impact of stochastic shocks.

By consecutively applying this relation at each time step, the entire synthetic series is constructed, reflecting both the persistence of the phenomenon and the stochastic influence of unpredictable events. The resulting dataset serves as a standardized testing basis, allowing evaluation and comparison of forecasting models within a controlled environment that sufficiently approximates the real behavior of drone-induced risks at airports.

The resulting series was divided into two subsets: a training set of 200 values and a test set of 50 values, in order to evaluate the forecasting performance on data outside the training sample.

3.2 Modeling AR

This model assumes that the value of a variable at time t is linearly dependent on its previous value at time $t-1$, with the relationship expressed by formula (1).

In this initial modeling step, it was considered that the constant c equals zero. This choice is based on the observation that the data series does not display a systematic upward or downward trend, justifying the omission of a constant offset. Additionally, to more clearly evaluate the relationship between successive values, it was initially assumed that the stochastic error is zero ($\epsilon_t = 0$).

The estimation of the coefficient φ was performed using the 200 values of the series generated in the previous step, employing the formula:

$$\varphi = \frac{\sum(X_{t-1} - \bar{X})(X_t - \bar{X})}{\sum(X_{t-1} - \bar{X})^2} \tag{18}$$

where X_{t-1} represents the lagged values (shifted by one time step), X_t is the current value, and $\bar{X}$ is the arithmetic mean of the series. The numerator in this expression measures the covariance between successive values, while the denominator indicates the variance of the lagged values relative to the mean. The ratio of these two components quantifies the strength and direction of the linear relationship between consecutive values in the series.

After applying the formula, the arithmetic mean of the series was calculated as $\bar{X} \approx -0.1365$, and the estimated regression coefficient was $\varphi \approx 0.7375$. Using this coefficient, the AR(1) model was constructed. For each forecast, the following formula is applied:

$$X_t = \varphi_1 X_{t-1} \tag{19}$$

thus, to estimate the value X_{201}, the observed value X_{200} was used. Given that $X_{200} = -1.3094$ the prediction for X_{201} was calculated as:

$$X_{201} = 0.7375 \times (-1.3094) \approx -0.9667 \tag{20}$$

The AR(1) model will subsequently be used to generate a sequence of 50 future predictions, each based on the previously estimated value. These predictions will be compared with the actual series values to evaluate the model's performance.

The negative values in the predictive model output reflect fluctuations around a theoretical mean and do not indicate negative incidents in practice. Real-world applications would require post-processing adjustments to ensure physically meaningful, non-negative predictions.

3.3 Modeling MA

This model assumes that the value of a variable at time t is influenced by a linear combination of the current error and the error recorded at the previous step. Using formula (7), in this modeling stage, it was considered that the constant c equals zero, as no systematic trend of shift was observed in the series. Additionally, to initialize the prediction process, it was assumed that the current stochastic error ε_t is zero.

The estimation of the coefficient φ was performed based on the approximate relation:

$$\theta = \frac{\sum(X_{t-1} - \bar{X})(X_t - \bar{X})}{\sum(X_{t-1} - \bar{X})^2} \tag{21}$$

where X_{t-1} represents the lagged values (shifted by one time step), X_t the current values, and $\bar{X}$ the arithmetic mean of the series.

It should be noted that this estimation method constitutes an approximation, since formally, the coefficient θ_1 in an MA(1) model should be estimated based on the autocorrelation of the errors, not directly from the series values. This choice was made to maintain methodological consistency and simplicity in practical applications.

By calculating the mean of the series of 200 values, we obtained $\bar{X} \approx -0.1365$, and the moving average coefficient is $\theta \approx 0.7375$.

Based on these parameters, future predictions of the series can be made. To estimate the value X_{201}, the last known value $X_{200} = -1.3094$ was used, along with the MA(1) model formula, assuming the current error is zero, as follows:

$$X_{201} = \theta_1 * (X_{200} - (-0{,}1365)) = 0{,}7375 * (-1{,}1729) \approx -0{,}86450000 \tag{22}$$

Next, the MA(1) model will be used to generate a sequence of 50 future predictions, each based on the previously estimated error. The accuracy of the model will be evaluated by comparing the predicted values with the actual series values, similar to the AR model approach.

The negative values in the predictive model output reflect fluctuations around a theoretical mean and do not indicate negative incidents in practice. Real-world applications would require post-processing adjustments to ensure physically meaningful, non-negative predictions.

3.4 Modeling ARMA

The data series was modeled using a first-order autoregressive moving average model, referred to as ARMA(1,1). This model combines autoregressive (AR) and moving average (MA) components, aiming to capture both the dependence on past values and the influence of previous errors on the series evolution, as expressed by formula (7).

In this modeling stage, similar to previous models, it was assumed that the constant *c* equals zero to reflect the absence of a systematic trend in the data series. Additionally, to initiate the prediction process, the current error ϵ_t was presumed to be zero. The model coefficients were previously estimated as follows: the autoregressive coefficient $\varphi \approx 0.7375$, the moving average coefficient $\theta \approx 0.7375$, and the arithmetic mean of the series $\bar{X} \approx -0.1365$.

To estimate the value X_{201}, the known values X_{199} and X_{200} were used. First, the previous error ε_{200} was estimated using the relation:

$$\in 200 = X200 - \varphi X199 \tag{23}$$

Substituting the values:

$$\in_{200} = -1{,}3094 - (0{,}7375 \times -0{,}7683) \approx -0{,}7427 \tag{24}$$

Subsequently, the prediction for X_{201} was made by applying the full ARMA(1,1) model formula:

$$X_{201} = \varphi X_{200} + \theta_1 \in_{200} = (0{,}7375 \times -1{,}3094) + (0{,}7375 \times -0{,}7427) \approx -1{,}51440000 \tag{25}$$

The ARMA(1,1) model will be used to generate a sequence of 50 future predictions, each based on the previously estimated values and associated errors. The accuracy of the model will be evaluated similarly to the previous models.

3.5 Modeling LSTM

Building on the previously used classical methods, the data series was modeled using an LSTM (Long Short-Term Memory) neural network. This type of network is renowned for its ability to learn long-term dependencies in time series, surpassing the limitations of traditional statistical models.

An LSTM cell involves four main components: the forget gate (expressed by formula (8)), the input gate (10), the candidate state (11), and the output gate (14). To facilitate an intuitive understanding of the internal mechanism of the LSTM model at this stage, a simplification was applied: all weights (W_f, W_i, W_C) were set equal to 1, and all biases (b_f, b_i, b_C) were set equal to 0.

In general, weights represent adjustable coefficients that control the influence of each input element (either the previous value or the hidden state) on the activation of the gates within the LSTM cell. These weights are typically optimized automatically during the network's training process to minimize the prediction error.

Biases are additional terms introduced into each gate's calculations, serving to shift the activation function and provide the network with extra flexibility in adjusting its response to the input data. By setting all weights to 1 and biases to 0, the aim was to create a transparent and reproducible calculation example, where results can be explicitly derived without the need for a complex training or automatic optimization process.

For simplification of the manual calculation process, it was assumed that the previous hidden state h_{t-1} is equal to the last observed value, thus:

$$ht-1 = X200 \tag{26}$$

Typically, in trained LSTM networks, h_{t-1} represents a function of the internal memory state C_{t-1}, activated via the tanh function (12).

However, in the absence of a predefined hidden state and to maintain consistency with the available data, the assumption $h_{t-1} = X_{200}$ was adopted to enable transparent calculation procedures.

Starting from the last available value $X_{200} = -1.3094$, the following can be calculated:

Forget Gate

$$f_t = \sigma(h_t - 1 + xt) = \sigma(-2{,}6188) \approx 0{,}0679 \tag{27}$$

Input Gate

$$i_t = \sigma(h_t - 1 + xt) \approx 0{,}0679 \tag{28}$$

Candidate State

$$\{\tilde{C}\}_t = \tanh(-2{,}6188) \approx 0{,}679 \tag{29}$$

Memory state

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t = -0.1561 \tag{30}$$

Output Gate

$$o_t = \sigma(ht - 1 + xt) \approx 0{,}0679 \tag{31}$$

Table 1: Results for AR, MA, ARMA, and LSTM

Model	NRMSE	RMSE	MAE
AR(1)	0,0638	1.8360	1.3189
MA(1)	0,0712	1.7759	1.3686
ARMA(1,1)	0,0641	1.8626	1.3506
LSTM	0,0639	1.8063	1.2994

Final hidden state (prediction):

$$h_t = o_t \odot tanh\ (C_t) \approx -0{,}0105 \tag{32}$$

Therefore, the predicted value of X_{201} using the LSTM model was obtained as:

$$X_{201} \approx -0{,}01050000 \tag{33}$$

The LSTM model will be used to generate a sequence of 50 future predictions, each based on previously estimated values in an autoregressive regime.

The model’s accuracy will be assessed by comparing predicted values with the actual series values.

The constructed models (AR(1), MA(1), ARMA(1,1), and LSTM) were employed to produce forecasts over a horizon of 50 steps ahead, starting from the last available data point. Each model was applied according to its specific methodology, using previously estimated parameters and assuming controlled initial conditions (constant $c = 0$, zero current error for statistical models, and approximate initial states for the LSTM).

3.6 Performance evaluation

The predictions generated by each method were compared with the actual series values extracted from the original dataset to assess the accuracy of each model. The evaluation was performed using three well-established metrics in the literature: the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), and the Normalized Root Mean Square Error (NRMSE).

The Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\hat{y}_t - y_t)^2} \tag{34}$$

This metric penalizes larger errors more strongly and is sensitive to significant deviations from the actual values. The Mean Absolute Error (MAE):

$$MAE = \frac{1}{N}\sqrt{\sum_{t=1}^{N}|\hat{y}_t - y_t|} \tag{35}$$

where y_t is the actual value and ybt is the predicted value.

RMSE penalizes large errors more strongly due to the squaring, while MAE provides a more direct interpretation. This metric offers an intuitive measure of the average deviation between predictions and actual values and is more robust to extreme values compared to RMSE.

In addition to RMSE and MAE, the Normalized RMSE (NRMSE) was also calculated to provide a scale-independent assessment of predictive accuracy. This metric enables a fairer comparison of model performance across different data ranges and contexts. NRMSE is computed as:

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \tag{36}$$

where $y_{\max}$ and $y_{\min}$ are the maximum and minimum observed values in the dataset. By normalizing RMSE, this metric contextualizes the prediction error relative to the range of variation in the actual data.

By using all three measures simultaneously, a balanced evaluation of each model’s performance is achieved—considering average errors, the impact of large deviations, and scale-independence.

4. RESULTS AND DISCUSSION

The performance evaluation of the models using the three previously mentioned metrics (RMSE, MAE, and NRMSE) was carried out after computing the 50 predictions. Table 1 presents the computed error metrics—RMSE, MAE, and NRMSE—for each forecasting model, evaluated over a 50-step prediction horizon. These values provide a synthetic comparison of model performance, highlighting both absolute and normalized prediction errors.

The presented RMSE, MAE, and NRMSE values are calculated as overall averages over the test set containing 50 observations. RMSE reflects the sensitivity to large errors, MAE quantifies the average deviation from actual values, while NRMSE contextualizes these errors relative to the variability of the observed data, offering a normalized and thus comparable performance metric.

A comparative analysis of the errors reveals several relevant observations. The MA(1) model achieved the lowest RMSE value, suggesting that, on average, the squared deviations between predicted and actual values are the smallest in its case. This performance indicates that the MA(1) better handles large deviations, being less sensitive to extreme fluctuations in the series.

On the other hand, the LSTM model stood out with the lowest MAE value. This shows that, on average, the absolute deviations from the real values are the smallest, suggesting more stable and uniformly distributed predictions. Given that MAE is less sensitive to isolated large errors, it can be inferred that the LSTM tends to produce more controlled and predictable errors throughout the entire prediction sequence.

In terms of NRMSE, which accounts for the scale of the data, the AR(1) model achieved the lowest score (0.0638), closely followed by LSTM (0.0639) and ARMA(1,1) (0.0641). These values indicate that, relatively to the data's range, all three models maintain similarly low normalized errors. Interestingly, although MA(1) had the lowest RMSE in absolute terms, it performed the worst in normalized terms (NRMSE = 0.0712), reinforcing that absolute performance can be misleading when comparing across datasets or metrics.

These findings confirm the added value of NRMSE as a complementary indicator: it highlights that AR(1), despite its simplicity, performs competitively in relative terms, while LSTM maintains a strong balance across all three metrics.

It is also noteworthy that the ARMA(1,1) model, although combining two theoretically complementary components (autoregression and moving average), did not outperform the simpler models within the given data context. This outcome suggests that, under certain conditions, added complexity does not automatically guarantee higher predictive accuracy.

Based on these results, it can be stated that, in the present analysis, the LSTM model offers the best overall performance, thanks to its minimal MAE value and near-optimal NRMSE. The differences between the MA(1) and LSTM models are relatively small, so the final choice between them may depend on the specific application and on the importance given to stability versus sensitivity to extreme deviations.

5. PRACTICAL APPLICABILITY IN REAL-WORLD CONTEXT

The prediction models analyzed, especially the LSTM model, show significant practical potential for integration into systems dedicated to monitoring and forecasting drone-related risks at airports. In a dynamic and complex operational environment, the capability to anticipate potential threats is essential for maintaining high security levels.

LSTM, by its nature, is well-suited for processing and learning from complex time series data, where relationships between successive observations can be nonlinear and span long periods. This characteristic makes it particularly recommended for applications like predicting drone incidents, where traditional statistical models may struggle to capture all nuances of the phenomenon. Moreover, the inclusion of NRMSE allowed a scale-independent evaluation of performance, confirming that models like AR(1) may still offer competitive relative accuracy in simpler operational settings.

However, practical deployment of an LSTM model in a real-world operational context requires meeting several essential conditions:

1. Data availability and quality: LSTM models need large volumes of well-labeled, relevant data for training.Lack of such data can considerably reduce accuracy;
2. Computational power and infrastructure: Implementing an LSTM in real-time systems demands infrastructurecapable of continuous data processing and delivering predictions with minimal delay;
3. Robustness to unexpected variations: Given the unpredictable factors in airport environments, models must beresilient enough to handle such variations without significant performance loss.

While in this study the LSTM didn't demonstrate full superiority - mainly because synthetic data didn't include external factors or complex patterns - it could, in a more complex real scenario, capture nonlinear relationships - such as risk increasing more than linearly with the number of drones or seasonal dependencies (more incidents in summer).

If trained on sufficiently rich data - including incident history and explanatory variables such as weather, drone sales, air traffic levels - it could offer multidimensional risk forecasts, not merely trend extrapolation. Practically, robustness and interpretability are crucial: aviation decision-makers prefer transparent tools. ARMA models are attractive because they are understandable and explainable (e.g., "if an incident happened last month, the risk increases by X this month"). Conversely, LSTM acts as a "black box," which raises trust issues—if it indicates a sudden risk rise, explaining what pattern it identified becomes difficult, making it harder for authorities to act confidently.

A potential approach is combining both: using LSTM for accuracy and traditional statistical models or sensitivity analysis for interpretability. Moreover, risk-based methodologies tailored to infrastructure resilience have been proposed as essential in proactive threat detection [13]. In the literature, LSTM has been explored for risk monitoring in aviation and surveillance, focusing on anomaly detection or trajectory prediction in restricted airspace [12][13]. However, I have not found studies specifically addressing integrating LSTM into airport security strategies aimed at anticipating drone threats.

6. CONCLUSIONS

The increasing prevalence of drone-related incidents in airport environments poses a multidimensional threat to civil aviation, requiring proactive, data-driven risk forecasting strategies. The current study addressed this need by exploring whether time series prediction models can effectively estimate short-term UAV-related risks and support airport decision-making processes.

The main objective was to compare the forecasting performance of classical autoregressive models - AR(1), MA(1), and ARMA(1,1) - with that of a recurrent neural network model (LSTM), using synthetic time series data designed to simulate risk fluctuations over time. The modeling process included standardized training-test splits and controlled assumptions to ensure a fair comparative framework.

From a methodological standpoint, the study introduced a triad of performance metrics—RMSE, MAE, and the less frequently used Normalized RMSE (NRMSE)—which collectively provide a robust evaluation of model performance. The inclusion of NRMSE was particularly relevant given the continuous and normalized nature of the synthetic data, allowing for scale-independent comparison. This contributed to a more nuanced understanding of forecasting accuracy, especially when differences in absolute error magnitudes were marginal.

The results demonstrate that LSTM delivered the lowest average absolute error (MAE), indicating more stable and evenly distributed predictions across the forecast horizon. Surprisingly, the AR(1) model, despite its simplicity, achieved the lowest normalized error (NRMSE), reinforcing the value of basic autoregressive structures in data-limited or low-complexity contexts. The ARMA(1,1) model, although theoretically more expressive, did not significantly outperform the simpler alternatives, suggesting that increased model complexity does not automatically translate into superior predictive accuracy when applied to stationary or low-noise series.

From an applied perspective, these findings validate the potential for integrating such models into operational airport security systems. LSTM, with its ability to capture non-linear temporal dependencies, is especially suited for complex real-world scenarios where risk dynamics are influenced by multiple, interacting factors. Conversely, simpler models like AR(1) or ARMA(1,1) may offer quick, transparent, and interpretable risk forecasts—critical attributes for security personnel requiring explainable decision support. A hybrid framework that leverages both model categories could provide the optimal trade-off between accuracy and interpretability.

Nevertheless, the scope of the present work is constrained by the use of synthetic data and the exclusion of exogenous explanatory variables such as meteorological conditions, drone sales, and air traffic volume. Future research should focus on validating these findings using real-world UAV incident datasets collected from multiple airport environments and incorporating external drivers of risk. Additionally, it would be valuable to assess model performance under scenarios of sudden regime shifts - such as new anti-drone technologies or policy changes - which may affect the underlying risk dynamics.

Given the encouraging performance of the LSTM model and the operational simplicity of AR(1), the study recommends a pilot implementation wherein airport authorities periodically run the models on up-to-date data, evaluate their predictive consistency, and use the outputs to inform preventive resource allocation. Such a system, if continuously refined, could offer an anticipatory advantage in managing drone incursions - minimizing reactive responses and enhancing the resilience of airport operations.

In conclusion, this study contributes both a methodological framework and an applied insight into the role of predictive modeling in UAV threat management. By combining statistical and neural approaches, and by introducing scale-independent performance metrics, the research highlights a feasible path forward for integrating predictive analytics into airport risk assessment ecosystems, while emphasizing the need for further empirical validation in operational contexts.

REFERENCES

[1] N. Zhang, H. Liu, B. F. Ng, K. H. Low, Collision probability between intruding drone and commercial aircraft in airport restricted area based on collision-course trajectory planning, *Transportation Research Part C: Emerging Technologies*, vol. **120**, 2020.

[2] * * * BBC, Gatwick Airport drone attack: Police have 'no lines of inquiry', https://www.bbc.com/news/uk-englandsussex-49846450, 2019.

[3] R. J. Ball, *The Proliferation of Unmanned Aerial Vehicles: Terrorist Use, Capability, and Strategic Implications*, Lawrence Livermore National Laboratory, USA, 2017.

[4] W. J. Austin, S. J. Lord, S. A. Bridges, Vulnerability of manned aircraft to drone strikes*, European Union Aviation Safety Agency (EASA)*, 2020.

[5] * * * European Union Aviation Safety Agency (EASA), Drone Incident Management at Aerodromes (Part 1), 2020.

[6] J. Pyrgies, The UAVs Threat to Airport Security: Risk Analysis and Mitigation, *Journal of Airline and Airport Management*, vol. **9**, pp. 63–96, 2020.

[7] G. Wild, J. Murray, G. Baxter, Exploring Civil Drone Accidents and Incidents to Help Prevent Potential Air Disasters, *Aerospace*, 2016.

[8] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, *Time Series Analysis: Forecasting and Control*, Wiley, 2016.

[9] K. Greff, R. K. Srivastava, J. Koutn´ık, B. Steunebrink, J. Schmidhuber, LSTM: A Search Space Odyssey, *IEEE Trans. on Neural Networks and Learning Systems*, vol. **28**, no. 10, 2017.

[10] J. Schmidhuber, *Long Short-Term Memory*, https://deeplearning.cs.cmu.edu/F23/document/readings/LSTM.pdf, 2017.

[11] R. Xiao et al., *Predict Stock Prices with ARIMA and LSTM*, arXiv:2209.02407, 2014.

[12] Z. Dang et al., CA-LSTM: An Improved LSTM Trajectory Prediction Method Based on Infrared UAV Target Detection, *Electronics*, vol. **9**, 2023.

[13] M. El-Latif, Detection and Identification of Drones Using LSTM and Bayesian Optimization, *Multimedia Tools and Applications*, 2024.

[14] E. Susto, A. Schirru, S. Pampuri, M. McLoone, A. Beghi, Machine Learning for Predictive Maintenance: A Multiple Classifier Approach, *IEEE Transactions on Industrial Informatics*, **11**(3), pp. 812–820, 2015.

[15] M. Fiore, C. Spognardi, S. Murugesan, On the use of autoregressive models for cyber threat detection, ACM *International Workshop on Cyber-Physical Systems Security and Privacy (CPS-SPC)*, 2019.

[16] T. Bontempi, J. Zecchinon, M. P. Deisenroth, *Hybrid ARIMA–LSTM Model for Anomaly Detection in Time Series*, arXiv preprint arXiv:2202.09746, 2022.