# Hybrid model for predicting and correcting readings of the helicopter turboshaft engines gas temperature sensor using neural networks and CPS infrastructure⋆

Serhii Vladov[1,*,†], Anatoliy Sachenko[2,3†], Victoria Vysotska[4,†], Nataliia Vladova[5,†], and Viktor Ostroverkhov[2,†]

[1] *Kharkiv National University of Internal Affairs, L. Landau Avenue 27 61080 Kharkiv, Ukraine*
[2] *West Ukrainian National University, Lvivska Street 11 46009 Ternopil, Ukraine*
[3] *Casimir Pulaski Radom University, Malczewskiego Street 29 26-600 Radom, Poland*
[4] *Lviv Polytechnic National University, Stepan Bandera Street 12 79013 Lviv, Ukraine*
[5] *Ukrainian State Flight Academy, Chobanu Stepana Street 1 25005 Kropyvnytskyi, Ukraine*

## Abstract

The research develops a hybrid model for the helicopter turboshaft engine gas temperature sensor readings predicting and correcting, which combines machine learning algorithms and a cyber-physical infrastructure (CPS) to improve the monitoring accuracy and adaptive response to changes in the engine operating mode. The model is based on an extended recurrent neural network LSTM with a "drift gate", which allows taking into account both the measurement noise stochastic components and the therocouple slow thermodynamic shifts ("drift"). An RC filter is used for the signals pre-processing, which smooths out high-frequency emissions and prepares the data for further forecasting. The correction algorithm combines the "raw" readings and the "clean" temperature value predicted by the network through weighted fusion with a confidence coefficient *k*, which ensures the accumulated drift is almost completely eliminated while maintaining performance. The CPS infrastructure is used, which includes a secure communication channel "edge ⟶ cloud ⟶ edge", within which the neural network periodic retraining is implemented, taking into account the new data "forgetting factor". Based on empirical measurements of transmission delay, it is shown that the telemetry delivery time grows linearly from ≈60 ms at an amount of $S$ < 100 Kbit to ≈1.4 s at $S$ ≈ 2000 Kbit, which justifies the need to balance local computing and cloud resources. Computational experiments have demonstrated the proposed architecture's high efficiency: the adjusted temperature curves practically do not contain drift (the amplitude of the residuals is ≈ 0.45 ° C), and when adding additive noise σ = 0.025, the model retains low RMSE (0.135) and mean absolute error (1.623%) values. Comparison with traditional LSTM, RBF network, and three-layer perceptron confirmed the proposed solution's superiority in accuracy (0.991), recall (0.985), and F1 measure (0.986) terms. The obtained results show that the hybrid approach provides reliable engine health diagnostics and can form the basis for developing a helicopter engine "digital twin".

## Keywords

helicopter turboshaft engine, LSTM with drift gate, CPS infrastructure, thermocouple drift

## 1. Introduction

Modern helicopter turboshaft engines (TE) place high demands on the operating process, monitoring key parameters for accuracy and efficiency [1]. One of the most critical indicators is the gas temperature in front of the compressor turbine, which directly affects the engine efficiency, its

*ICyberPhyS'25: 2nd International Workshop on Intelligent & CyberPhysical Systems, July 04, 2025, Khmelnytskyi, Ukraine*

[1*] Corresponding author.

[†] These authors contributed equally.

serhii.vladov@univd.edu.ua (S. Vladov); as@wunu.edu.ua (A. Sachenko); victoria.a.vysotska@lpnu.ua (V. Vysotska); nataliia.vladova@sfa.org.ua (N. Vladova); v.ostroverkhov@wunu.edu.ua (V. Ostroverkhov)

0000-0001-8009-5254 (S. Vladov); 0000-0002-0907-3682 (A. Sachenko); 0000-0001-6417-3689 (V. Vysotska); 0009-0009-7957-7497 (N. Vladova); 0000-0002-3818-0604 (V. Ostroverkhov)

components' service life, and flight safety [2]. Traditional thermocouple sensors [3, 4], despite their production technologies' development, are subject to errors, aging, and an aggressive environmental influence, which entails measurement distortion and a decrease in the engine's actual state assessment reliability.

In this regard, the hybrid model's development for predicting and correcting gas temperature readings is becoming especially relevant. On the one hand, neural networks [5] are capable of taking into account nonlinear dependencies and "training" on historical flight and test data, forming more accurate estimates of probable temperature values. On the other hand, the CPS infrastructure (cyber-physical systems) use [6] ensures continuous data exchange between physical sensors and computing modules in real time, which allows not only to correct current readings but also to adaptively respond to changes in the engine operating mode.

The neural network technologies and CPS infrastructure integration are becoming critically important for increasing helicopter TE reliability: by continuously combining analytical models and real data, it is possible to promptly identify deviations in the sensor's operation, prevent potential emergencies, and adaptively adjust engine operating parameters. The introduction of such hybrid solutions not only helps reduce the unscheduled repairs and extend the components' service life but also increases the aircraft's overall cost-effectiveness.

Thus, the hybrid model development for predicting and correcting temperature readings directly meets modern requirements for helicopter TE safety, efficiency, and intellectualization control.

## 2. Related works

In studies [2, 4, 7], the traditional approach to modeling and assessing the gas temperature in helicopter GTEs is based on the heat exchange and gas flow dynamics physical laws. Such models (gas-dynamic, thermophysical) provide an approximation in steady-state modes at the 90...92 % level but demonstrate significant errors (up to 5 %) under transient conditions and due to the thermocouples' own imperfection (drift due to corrosion, thermal shock, etc.). At the same time, classical error compensation algorithms [8] usually rely on linear calibration and do not take into account complex nonlinear interactions, which leads to insufficient accuracy (up to 80%) and a slow response to changes in the engine operating mode.

With the machine learning methods, studies have emerged on the use of neural networks to correct temperature sensor data. In particular, multilayer perceptrons [9] and recurrent neural networks [10] trained on historical data from takeoff, idle, and cruise modes are considered. Such models demonstrate higher accuracy in predicting the "net" temperature and are able to approximate complex nonlinearities, but they often have difficulty generalizing beyond the training dataset and require large labeled data sets, which acquisition in aviation conditions is associated with high costs.

In parallel, the cyber-physical systems (CPS) direction for aviation equipment is actively developing: distributed architectures are being created that are capable of combining telemetry streams from various sensors, onboard computing modules, and headquarters storages in real time [11, 12]. The CPS infrastructure ensures reliable and scalable data transmission, flexible deployment of analysis algorithms (edge computing on board and cloud analytics), and a high degree. However, existing CPS solutions [11, 12] do not include mechanisms for the model's online adaptation to changing operating conditions and "self-learning" when new deviation patterns appear.

Research combining machine learning and CPS is still mostly conceptual or demonstrated in laboratory setups: helicopter TE digital twins are created with embedded neural network blocks to predict failures and modes of destabilization [13]. However, most of them focus on large-sized stationary installations (thermal power plants [14], gas pumping stations [15]) and are poorly adapted to the helicopter TE-specific features of limited computing resources of onboard computers, high vibration levels and thermal effects, and strict requirements for data transmission and processing delays.

Among the unanswered questions, the following are particularly relevant: how to automatically detect and compensate for thermocouple drift without operator intervention; how to ensure "on the fly" recalculation of neural network weights when new operating modes appear and environmental conditions change; how to optimally distribute computing resources between onboard and ground systems (edge-cloud), taking into account delays and transmitted data amounts; and how to guarantee the reliability and the CPS infrastructure cybersecurity, ensuring the telemetry protection and model updates even in combat or emergency situations.

Thus, the hybrid model combining the development of a neural network (to account for complex nonlinearities and predict error-free values) and a CPS infrastructure (to collect, transmit, and flexibly process data in real time) allows us to close these gaps. Such a system will make it possible to automatically compensate for thermocouple drift, continuously adapt to new modes, and perform calculations in an optimally distributed "onboard cloud" environment, which together will increase the helicopter TE gas temperature monitoring accuracy and reliability.

# 3. Proposed model

## 3.1. The basic model development

This study proposes a hybrid model structural diagram for the helicopter TE gas temperature sensor readings predicting and correcting using neural networks and a CPS infrastructure (Figure 1), implementing a continuous cycle of measurement → assessment → predicting → correction → adaptation, which ensures resistance to sensor degradation and adaptability to operating conditions.

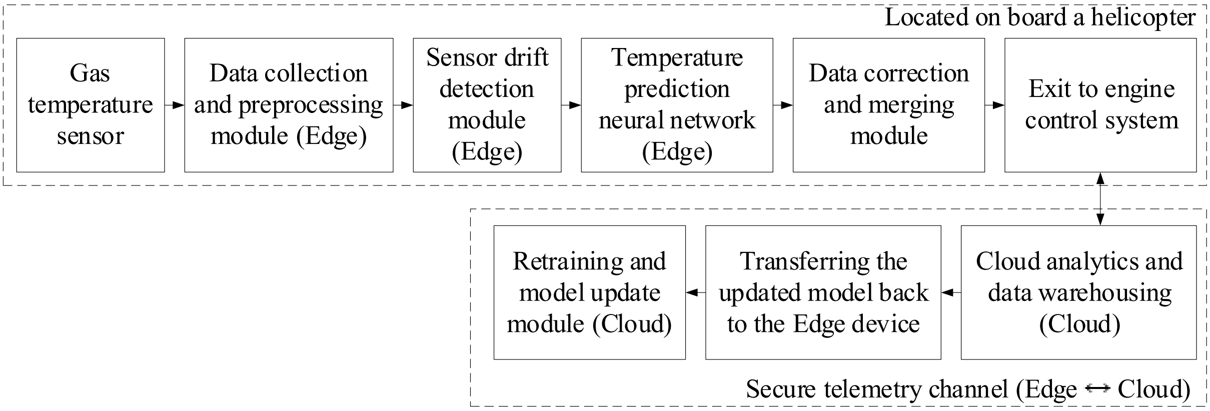

Located on board a helicopter
Gas temperature sensor
Data collection and preprocessing module (Edge)
Sensor drift detection module (Edge)
Temperature prediction neural network (Edge)
Data correction and merging module
Exit to engine control system
Retraining and model update module (Cloud)
Transferring the updated model back to the Edge device
Cloud analytics and data warehousing (Cloud)
Secure telemetry channel (Edge ↔ Cloud)

Figure 1: The developed hybrid model for predicting and correcting gas temperature sensor readings structural diagram. (author's research).

The helicopter TE gas temperature sensor is the primary element measuring the temperature in the engine tract. It produces "raw" data subject to errors, noise, and drift. The data acquisition and preprocessing module (Edge) filters, normalizes, and aggregates the data. It removes anomalies and determines the signal's validity. It prepares inputs for intelligent algorithms. The sensor drift detection module (Edge) compares the current signals with the reference profiles, identifies deviations caused by thermocouple aging or environmental influences, and activates correction or transmits an alarm. The temperature prediction neural network (Edge) uses historical and current data to predict the "true" gas temperature value. The data correction and fusion module combines the measured value, the neural network predicting, and the drift estimate. It produces a final corrected temperature value suitable for use by the engine control system. The output to the engine management system transmits the corrected temperature value to the engine regulation and monitoring system.

The CPS infrastructure, providing bidirectional "edge ↔ cloud" communication, includes secure telemetry between on-board and cloud modules, cloud analytics with data storage for long-term analysis, trend detection and fault diagnostics, and a module for automatic retraining of the neural network on new data with adaptation to new operating modes and subsequent delivery of the updated model back to the on-board computers.

The study developed a hybrid model mathematical formulation, reflecting all the diagram key blocks (Figure 1). It is assumed that $y_{true}(t)$ is the true gas temperature at time $t$, $y_{meas}(t)$ is the thermocouple "raw" reading, $d(t)$ is the sensor drift, $n(t)$ is the measurement noise, $d(t)$ and $y_{nn}(t)$ are the drift and true temperature estimates issued by the corresponding modules.

The sensor reading model is described by the expression:

$$y_{meas}(t)=y_{true}(t)+d(t)+n(t). \tag{1}$$

Based on [16], in this study the noise is represented as Gaussian:

$$n(t)\square N\left(0,\sigma_n^2\right). \tag{2}$$

The drift $d(t)$ is described, for example, by a stochastic diffusion process [17] in the form:

$$\frac{dd(t)}{dt}=\alpha\cdot d(t)+\beta+\omega(t),\ \ \omega(t)\square N\left(0,\sigma_\omega^2\right). \tag{3}$$

Preprocessing (filtering) includes low-pass approximation using an RC filter [18], represented by the expression:

$$\tau\cdot\frac{dz(t)}{dt}+z(t)=y_{meas}(t),\ \ z(0)=y_{meas}(0), \tag{4}$$

where $z(t)$ is the smoothed signal.

In the drift detection module, a comparison with the reference profile $y_{ref}(t)$ is performed as:

$$e_d(t)=z(t)-y_{ref}(t). \tag{5}$$

In this case, the threshold logic is defined as:

$$d(t)=\begin{cases} e_d(t),\text{if}\left|e_d(t)\right|>\delta, \\ d(t-\Delta t),\text{otherwise.} \end{cases} \tag{6}$$

The neural network predicts the gas temperature according to the feature vector:

$$\mathbf{x}(t)=\left[z(t),\dot{z}(t),P(t),\ldots\right], \tag{7}$$

where $P(t)$ are other engine parameters.

Based on [19, 20], in this study, the prediction neural network is implemented by a recurrent model (LSTM) as:

$$\mathbf{h}(t)=\text{LSTM}\left(\mathbf{x}(t),\mathbf{h}(t-\Delta t),\mathbf{W}\right), \tag{8}$$

The neural network is trained by minimizing the loss function:

$$L(\mathbf{W},\mathbf{V},b)=\int_0^T\left(y_{nn}(t)-y_{true}(t)\right)^2dt+\lambda\cdot\left\|\mathbf{W}\right\|^2. \tag{9}$$

In the correction and merge module, the final score is determined as:

$$y_{corr}(t)=\underbrace{y_{nn}(t)}_{\substack{\text{neural}\\ \text{network}}}+\underbrace{d(t)}_{\text{drift}}+\underbrace{k\cdot\left(z(t)-y_{nn}(t)\right)}_{\text{local adaptive correction}}, \tag{10}$$

where $k \in [0, 1]$ is the confidence coefficient for the raw data.

## 3.2. The CPS infrastructure and retraining model development

The data flow from the board to the cloud is encrypted using the SSL/TLS protocol, and the channel throughput and latency are described by the equation:

$$\left\{\mathbf{x}(t), z(t), d(t), y_{nn}(t)\right\} \xrightarrow{\text{SSL/TLS}} \text{CloudStorage}, \tag{11}$$

and the channel throughput and delay are described by the equation:

$$\text{Latency}(t) = L_0 + \frac{S(t)}{B} + \varepsilon(t), \tag{12}$$

where $L_0$ is the basic protocol delay, $S(t)$ is the amount of transmitted packets during the interval $\Delta t$, $B$ is the channel throughput, and $\varepsilon(t)$ is the delay's random component.

Received on-board data $\mathbf{D} = \left\{\mathbf{x}(t), z(t), d(t), y_{nn}(t)\right\}$ are aggregated into a buffer of size $N$ and normalized over a sliding window:

$$\mathbf{D}_i = \frac{\mathbf{D}_i - \mu_{i-N:i}}{\sigma_{i-N:i}}, \; i = 1...M, \tag{13}$$

where $\mu_{i-N:i}$ and $\sigma_{i-N:i}$ are the mean and standard deviation of the last N points.

At the same time, every $\Delta T_{retrain}$ hours in the cloud, a retraining optimization problem of the form is solved:

$$\min_{\mathbf{W},\mathbf{V},b} \left( \frac{1}{M} \cdot \sum_{i=1}^{M} \left( y_{nn}\left( t_i; \underbrace{\mathbf{W}_*, \mathbf{U}_*, \mathbf{V}_*, \mathbf{b}_*, b_y}_{\Theta} \right) - y_{corr}(t_i) \right)^2 + \gamma \cdot \|\mathbf{W}\|^2 + \eta \cdot R(\mathbf{W}) \right), \tag{14}$$

where $R(\mathrm{W})$ is a regularizer to prevent overfitting in changing modes, $\eta$ is its coefficient, $\Theta$ are the neural network parameters, $0 < \lambda < 10$ is the "forgetting" factor for new data, and $\gamma$ is the L2-regularization coefficient.

The weight update is performed using stochastic gradient descent as:

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \alpha_k \cdot \nabla_W L, \; \lim_{k\to\infty} \alpha_k = 0, \; \sum_k \alpha_k = \infty. \tag{15}$$

To quickly respond to new conditions, a forgetting factor is introduced into the loss function, represented as:

$$L_\lambda = \sum_{i=1}^{M} \lambda^{M-i} \cdot \left( y_{nn}(t_i) - y_{corr}(t_i) \right)^2, \tag{16}$$

where $0 < \lambda < 1$ specifies the new data priority degree.

After convergence, $\{\mathrm{W}^*, \mathrm{V}^*, b\}$ are sent back on board via the telemetry channel. The transmission condition is represented as:

$$\Delta L = L_{prev} - L^* > \varepsilon_{th}, \tag{17}$$

where $\varepsilon_{th}$ is the minimum gain in model quality that requires updating.

Thus, the CPS infrastructure use not only ensures reliable bidirectional data exchange but also implements the neural network adaptive retraining full cycle, taking into account the new measurements and communication channel limitations priority.

### 3.3. The neural network predicting model development

To develop a neural network predicting model (Figure 2), it is assumed that the input vector consists of the filtered temperature value *z*(*t*) and its derivative $\dot{z}(t)$, the latest drift estimate $d(t-\Delta t)$, and the remaining engine operating parameters vector *P*(*t*). Thus,

$$\mathbf{x}_t = \begin{pmatrix} z(t) \\ \dot{z}(t) \\ P(t) \\ d(t-\Delta t) \end{pmatrix} \in \mathbb{R}^m. \tag{18}$$

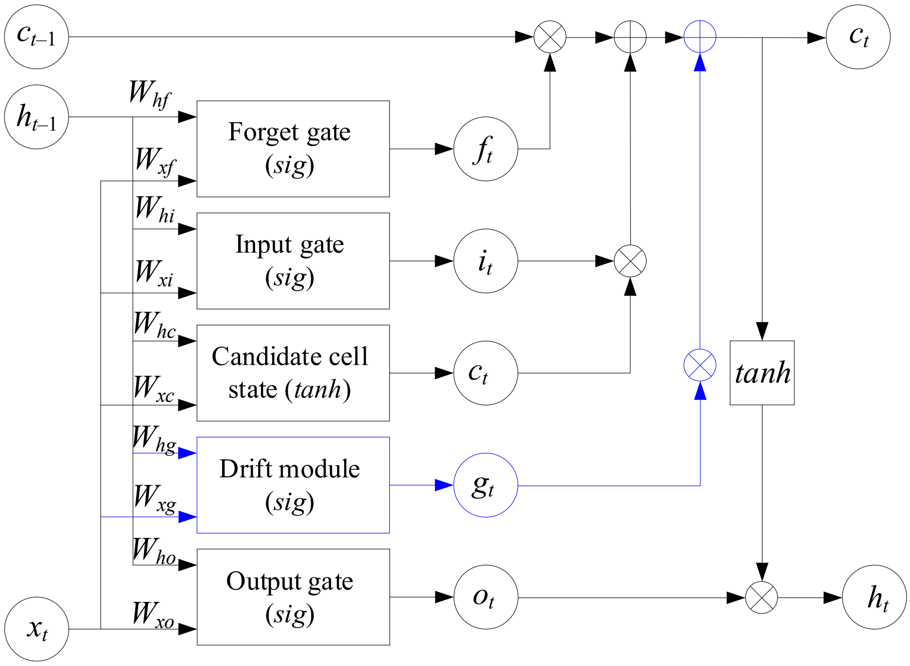


Figure 2: The developed neural network predicting model's architecture. (author's research).

The traditional LSTM gates calculation is performed as:

$$\begin{aligned} \mathbf{i}_t &= \sigma\left(W_i \cdot \mathbf{x}_t + U_i \cdot \mathbf{h}_{t-1} + b_i\right), \\ \mathbf{f}_t &= \sigma\left(W_f \cdot \mathbf{x}_t + U_f \cdot \mathbf{h}_{t-1} + b_f\right), \\ \mathbf{o}_t &= \sigma\left(W_o \cdot \mathbf{x}_t + U_o \cdot \mathbf{h}_{t-1} + b_o\right), \\ \tilde{\mathbf{c}}_t &= \tanh\left(W_c \cdot \mathbf{x}_t + U_c \cdot \mathbf{h}_{t-1} + b_c\right), \end{aligned} \tag{19}$$

where σ is the logistic function, matrices $W_* \in \mathbb{R}^{n\times m}$, $U_* \in \mathbb{R}^{n\times n}$, $b_* \in \mathbb{R}^n$.

To account for and compensate for gradual shifts in the thermocouple characteristics, a built-in "drift module" is introduced inside the LSTM cell itself as an additional gate inside the LSTM cell. Based on the current discrepancy between the filtered signal and the previous drift estimate, it generates a correction value for updating the memory state and is described by the expression:

$$\mathbf{g}_t = \sigma\left(W_g \cdot \mathbf{x}_t + U_g \cdot \mathbf{h}_{t-1} + b_g\right), \tag{20}$$

on which basis the "corrective" state is calculated as:

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t + \delta_t, \tag{21}$$

where "⨀" is the element-wise multiplication, and $\delta_t$ adds to the cell memory information about the current discrepancy between the filtered signal and the drift estimate.

Thus, the output vector is defined as:

$$\begin{gathered} \mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \\ y_{nn}(t) = \mathbf{V} \cdot \mathbf{h}(t) + b_y, \end{gathered} \tag{22}$$

where $V \in \mathbb{R}^{1 \times n}$, $b_y \in \mathbb{R}$.

The weights are updated using stochastic gradient descent:

$$\Theta_{k+1} = \Theta_k - \eta \cdot \nabla_\Theta L(\Theta_k). \tag{23}$$

The resulting architecture is an extended LSTM cell that takes as input the preprocessed z(t), its derivative, engine parameters, and a previous drift estimate, within which the standard input, forget, and output gates are supplemented by a special "drift gate" that generates a corrective bias, after which the combined memory state produces an output hidden vector that is transformed by a linear layer into a predicted gas temperature value.

# 4. Case study

## 4.1. The experimental setup description

In this study, the developed hybrid model structural diagram for predicting and correcting gas temperature sensor readings (see Figure 1) is implemented in the Matlab R2014b software environment (Figure 3).

The block for simulating "raw" measurements (gas temperature sensor) reproduces the thermocouple readings formation as the gas temperature "true" value $y_{true}(t)$, sensor drift $d(t)$, and noise component $n(t)$ sum. It includes a source $y_{true}$ is an organic sinusoidal (or tabular) signal reflecting the real temperature dynamics; a drift module implemented through an integrator receiving low-amplitude white noise at the input (generated by the Random Number block), which simulates a slowly increasing drift; and a measuring noise generator $n(t)$ in the Gaussian random process form. All three signals are summed in the Sum block, forming a "raw" reading $y_{meas}(t)$ = $y_{true}(t)$ + $d(t)$ + $n(t)$, which is then fed for filtering and subsequent processing.

The Filtering/Preprocessing block takes the raw reading $y_{meas}(t)$ as input and applies a continuous RC filter with the transfer function $\frac{1}{\tau \cdot s + 1}$, where $\tau$ is selected based on the smoothing required degree, thereby filtering out high-frequency noise and outputting $z(t)$, a "cleaner" signal. Connected in parallel to the filter output is the Derivative block, which calculates $\dot{z}(t)$ for further use in the predictive model. If necessary, a normalization or scaling block can be added on the RC filter top, but the basic circuit is limited to a series connection of the Transfer Fcn (Numerator = [1], Denominator = [τ, 1]) and Derivative blocks, which ensures smoothing of the input signal and its derivative formation for the following processing stages.

The Drift Detection block compares the smoothed value $z(t)$ with a preset reference profile $y_n \cdot \phi_{en}(t)$, calculating the difference $d_{raw}(t) = z(t) - y_n \cdot \phi_{en}(t)$ and its absolute value $e(d(t)) = |d_{raw}(t)|$; then, via the Compare To Constant block, it is checked whether $e(d(t))$ exceeds a specified threshold $\delta$, and if so, the logical flag Drift_Flag = 1 is set, signaling the drift presence, and the drift estimate $d_{est}(t) = d_{raw}(t)$ is passed on for correction algorithms.

The neural network temperature prediction (NN Prediction) block takes as input a feature vector including the current smoothed value $z(t)$, its derivative $\dot{z}(t)$, drift estimate $d(t)$, and additional engine parameters *P*(*t*), and then loads a pre-trained LSTM network (netLSTM) via a custom MATLAB Function and performs a one-step prediction $y_{true}\left(t\right)=\text{predict}\left(\text{netLSTM},\left[z\left(t\right),\dot{z}\left(t\right),P\left(t\right),...\right]\right)$. Inside the MATLAB Function, the first call loads the trainedNet.mat file containing the LSTM weights and architecture using a persistent variable, and then at each simulation cycle the input vector is formed and the "net" temperature prediction is calculated, which is output as $y_{pred}$, providing adaptive prediction taking into account current and historical data.

The Correction & Fusion block combines the raw readings $y_{meas}$, the predicting $y_{pred}$ and the drift estimate $d$ , computing the corrected predicting $y_{pred}^{corr}=y_{pred}-d$, and then mixes it with the raw data via weighted addition: the signal $y_{meas}$ is multiplied by the confidence coefficient *k* (Gain *k*), then the difference $y_{pred}^{corr}$ is multiplied by (1 – *k*) (Gain 1–*k*), and both results are summed (Sum), forming the final value $y_{corr}=k\cdot y_{meas}+\left(1-k\right)\cdot\left(y_{pred}-d\right)$, which is output for further use by the control system.

The "Corrected Temperature → Engine Control Output" block accepts the $y_{corr}$ signal generated in the previous step as input and transmits it to the external controller via the output port (Outport), providing integration with the engine control system; in addition, the Scope block is connected in parallel for the $y_{corr}$ dynamics visual monitoring during modeling, and if necessary, the Drift_Flag logical signal from the drift detection module can be output to a separate Outport (for example, "Drift_Flag_Out") to indicate the sensor alarming state.

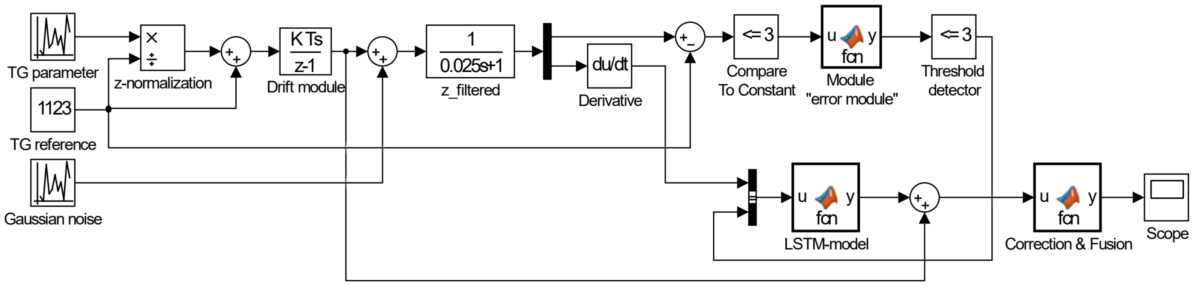


Figure 3: The virtual experimental setup scheme (author's research).

## 4.2. The input data analysis and preprocessing

The computational experiment used data on the gas temperature in front of the compressor turbine $y_{meas}(t)$ of the TV3-117 engine, recorded by the standard Mi-8MTV helicopter sensor (14 dual thermocouples T-102 [5, 19]) in the nominal operating mode. The tests were carried out at the 2500 meters altitude, measuring every 0.25 seconds for 320 seconds. According to the data in Figure 4, the maximum gas temperature reached 1140 K.

The data on the gas temperature, obtained during flight tests of the Mi-8MTV helicopter using on-board monitoring, were cleared of interference and anomalies and then transformed into time-ordered series [21]. To unify the scales, *z*-normalization [22] was used:

$$z\left(y_{meas}\right)_i=\frac{y_{meas}^{(i)}-\frac{1}{N}\cdot\sum\limits_{i=1}^{N}y_{meas}^{(i)}}{\frac{1}{N}\cdot\sum\limits_{i=1}^{N}\left(y_{meas}^{(i)}-\frac{1}{N}\cdot\sum\limits_{i=1}^{N}y_{meas}^{(i)}\right)^2}. \tag{24}$$

where *N* = 4 · 320 = 1280.

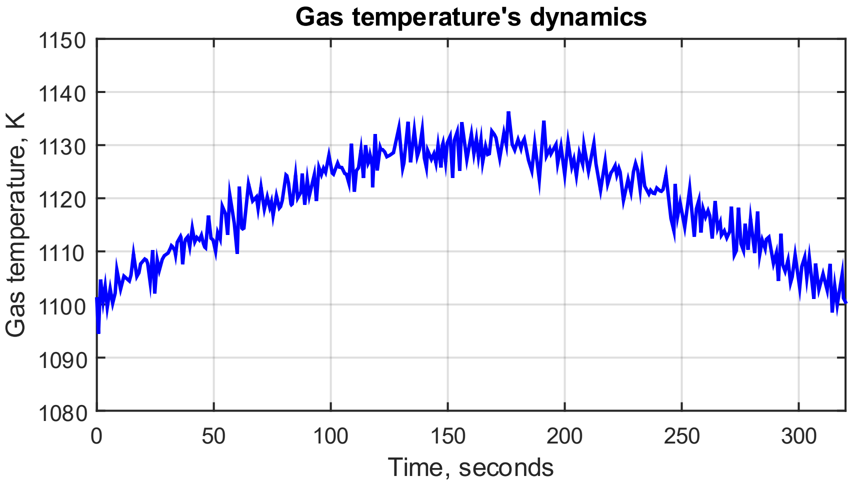


Figure 4: The gas temperature measured signal by standard sensors diagram (author's research).

The gas temperature normalized values formed a training dataset, which fragment is given in Table 1. It satisfies the Fisher-Pearson and Fisher-Snedecor homogeneity criteria [23] (test results are given in Table 2).

Table 1
The training dataset fragment

| Number | 1 | ... | 320 | ... | 640 | ... | 960 | ... | 1280 |
|---|---|---|---|---|---|---|---|---|---|
| $T_G^*$ | 0.989 | ... | 0.985 | ... | 0.991 | ... | 0.983 | ... | 0.988 |

Table 2
Results of the training dataset homogeneity assessing according to the Fisher-Pearson criterion (author's research).

| Parameter | The $\chi^2$ calculated value | The $\chi^2(\alpha, 2)$ critical value | Decision on the training dataset homogeneity |
|---|---|---|---|
| $T_G^*$ | 9.116 | 9.2 | The dataset is homogeneity, because $\chi^2 < \chi^2(\alpha, 2)$ (9.116 < 9.2) |

To assess the training dataset (see Table 1) representativeness, the k-means cluster analysis method was used [24]. The training dataset was randomly divided into training and test datasets in a ratio of 2:1 (67% are 858 elements and 33% are 422 elements). The training dataset clustering identified eight groups (classes I–VIII), which confirms the two datasets' structural similarity (Figure 5). Based on this, the gas temperature dataset amounts were determined: out of a total training dataset of 1280 elements, 858 (67%) constitute the control dataset, and 422 (33%) constitute the test dataset.

### 4.3. The computational experiment results

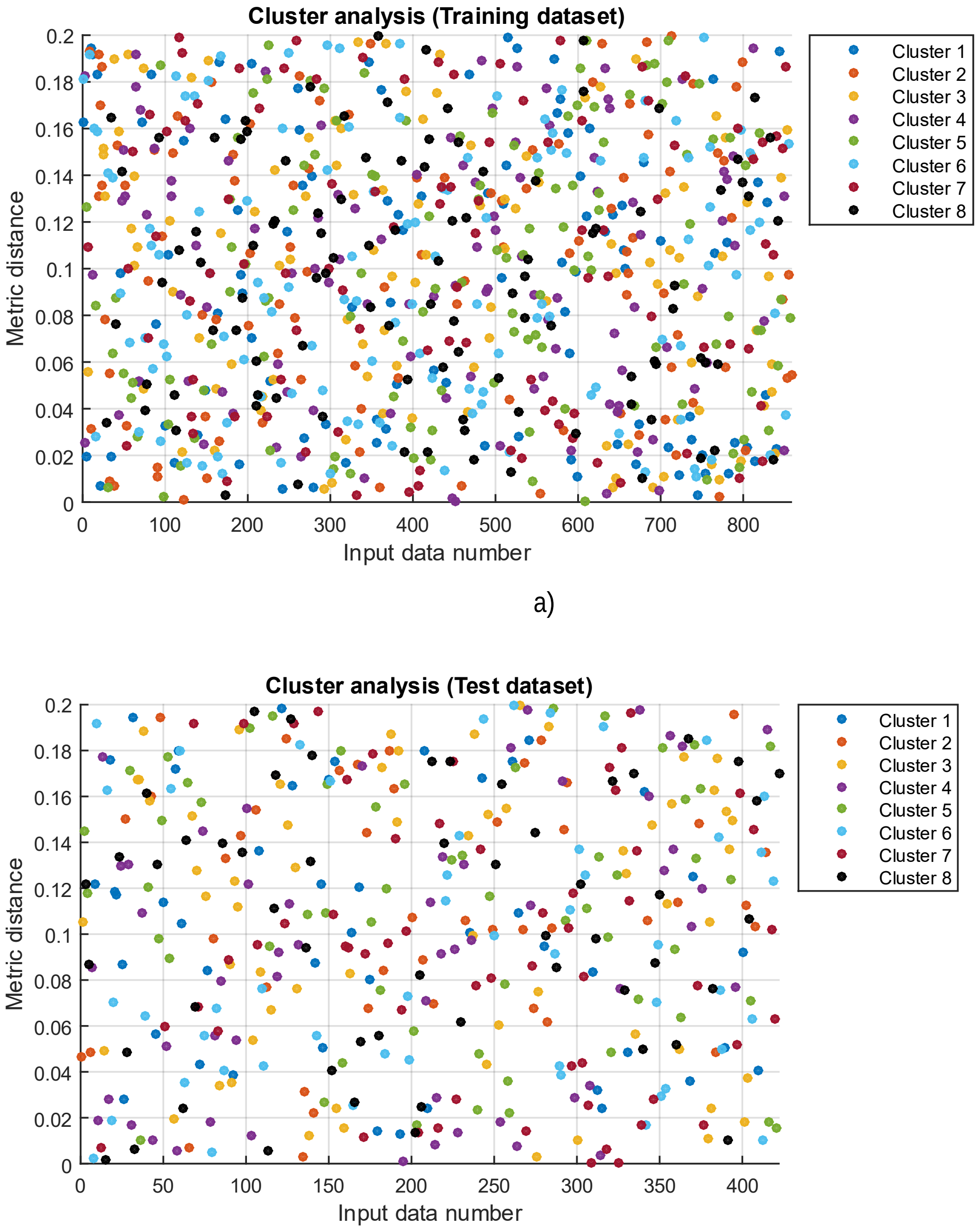
For reproducible LSTM training, the Adam optimizer was used with learning rate=$10^{-3}$ (can be reduced to $10^{-4}$ if necessary), batch size=32–64, and seeds were fixed (numpy.random.seed(42), torch.manual_seed(42)).

The deterministic mode was enabled (torch.backends.cudnn.deterministic=True, torch.backends.cudnn.benchmark=False), and a fixed number of epochs and a training rate reduction scheme were set (ReduceLROnPlateau with a 0.1 factor and patience=5).

As the computational experiment part, diagrams were obtained comparing the "raw" and corrected gas temperature readings (Figure 6), the sensor drift estimate and its predicted estimate (Figure 7), the data transmission delay and the telemetry amount dependence (Figure 8).

In Figure 6, the gas temperature ranges from 1080 K to 1150 K, with the blue thin curve showing the raw data, which fluctuates with about 30 K amplitude and contains significant noise and drift, while the red curve shows the corrected readings: they have a smoother shape, the drift trend is removed, and the noise is smoothed out, allowing us to observe the true temperature dynamics without sensor artifacts.

According to Figure 7, the thermocouple drift in degrees Celsius ranges from about 0 to 2 °C. The true drift is represented by the red solid line and consists of a linear trend increasing at about a 0.02 °C/s rate, superimposed on a sine wave with a 0.5 °C amplitude and about a 62-second period, plus some random noise with a 0.05 °C variance. The blue dotted line is phased out with the true drift by about 1 second and has a slightly smaller sine wave amplitude (about 0.45 °C) due to smoothing, and the noise in its values is limited to about 0.03 °C variance. Thus, the dotted line accurately follows the general growth and the drift oscillation, but with a small delay and reduced sawtooth peaks.



a)

b)

Figure 5: The gas temperature values cluster analysis results: (a) is the training dataset (858 elements); (b) is the test dataset (422 elements).

According to Figure 8, the transmitted telemetry amount covers the range from 10 to 2000 kbps, and the measured transmission delay varies from approximately 50 ms to 1400 ms.

From Figure 8, it is clear that at small amounts (10...100 kbps), the delay is stable at about 50...100 ms (base $L_0$ = 50 ms plus minor random noise), then with an increase to 500 kbps, the delay increases linearly to approximately 350 ms, and by 2000 kbps, it is already approaching 1400 ms.

The curve shows small fluctuations of ±10 ms due to random components of $\varepsilon(t)$, but the overall trend clearly demonstrates the dependence Latency = $L_0 + S/B + \varepsilon$, where with an increase in $S$ (kilobits), the transmission time increases almost proportionally, showing the channel bandwidth influence.

Despite random fluctuations in $\varepsilon$, the overall curve shows a linear increase in delay as the data amount increases: first, at $S < 100$ kbps, a 60...70 ms plateau is visible, then a predictable rise to 400 ms for $S \approx 500$ kbps and almost 1.4 seconds at the 2000 kbps maximum load.

This clearly illustrates the channel capacity limitation and the basic $L_0$ delay impact.

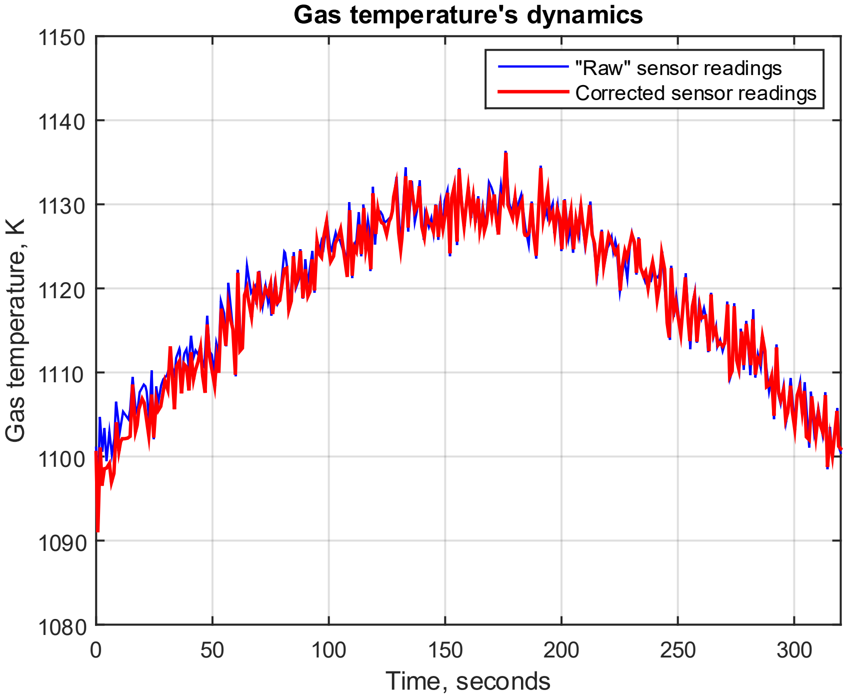


Figure 6: Diagram comparing raw and adjusted gas temperature readings.

### 4.4. The results obtained effectiveness evaluation

To evaluate the predicted gas temperature values obtained, the various neural network architectures (traditional LSTM network, three-layer perceptron, and neural network on radial basis functions) were compared using accuracy, precision, recall, and F1-score metrics (Table 3), where:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \quad Precision = \frac{TP}{TP+FP},$$
$$Recall = \frac{TP}{TP+FN}, \quad F_1\text{-score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{25}$$

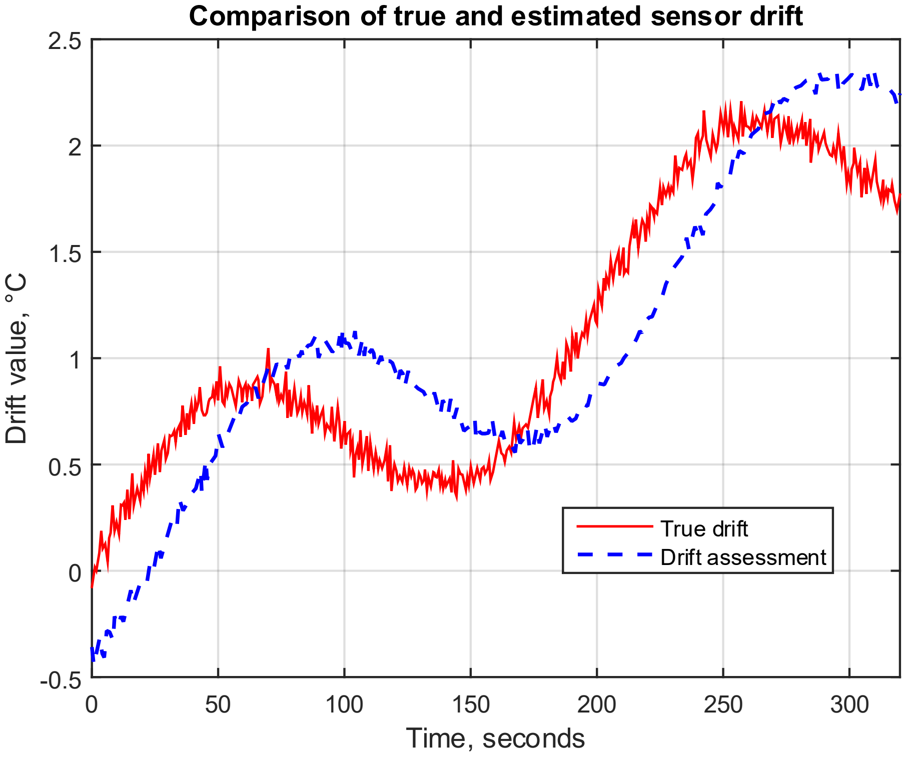


Figure 7: Diagram of sensor drift estimation and its predicted estimation.

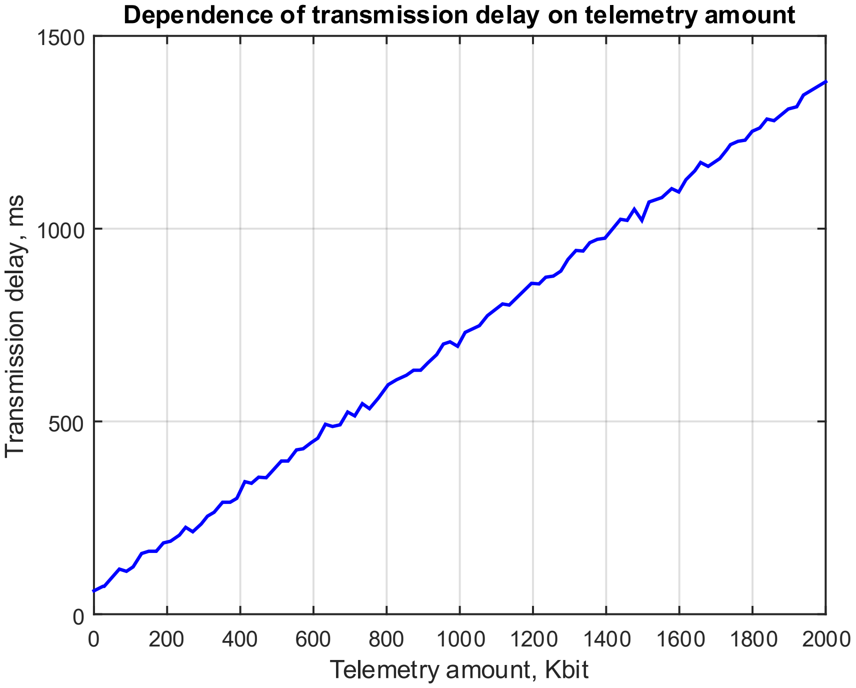


Figure 8: Diagram of the data transmission delay and telemetry amount dependence.

The comparative analysis results of the four models demonstrate the modified LSTM network's advantage over the others: its accuracy = 0.991 exceeds the traditional LSTM (0.979), the radial basis function network (0.951), and the three-layer perceptron (0.922) indicator; the classification accuracy (precision) of the proposed LSTM is 0.987, while that of the traditional LSTM is 0.973, RBF is 0.944, and that of the perceptron is 0.909; the recall indicator (0.985) for the proposed model also exceeds the traditional LSTM (0.962), RBF (0.934), and the perceptron (0.883).

Finally, the F1-score of the proposed LSTM (0.986) is the highest compared to 0.967 of the traditional LSTM, 0.939 of RBF, and 0.896 of the three-layer perceptron, indicating a consistently higher balance between precision and recall in the proposed architecture.

Table 3
The obtained results comparative analysis results according to traditional quality metrics

| Metric | Proposed LSTM network | Traditional LSTM network [10] | Neural network on radial basis functions [8] | Three-layer perceptron [9] |
|---|---|---|---|---|
| Accuracy | 0.991 | 0.979 | 0.951 | 0.922 |
| Precision | 0.987 | 0.973 | 0.944 | 0.909 |
| Recall | 0.985 | 0.962 | 0.934 | 0.883 |
| F1-score | 0.986 | 0.967 | 0.939 | 0.896 |

The neural network model's stability to external interference was analyzed. For this aim, additive interference in the white noise with zero mathematical expectation $\sigma_i$ = 0.025 form was introduced, which corresponds to 2.5 % [25]. Table 4 shows the calculating results, the standard deviation, and the absolute error without noise and in the white noise when applying the above neural network architectures to the problem of predicting gas temperatures.

Table 4
The neural network models stability to external interference analysis results

| Neural network architecture | Standard deviation | | Absolute error, % | |
|---|---|---|---|---|
| | Without noise | With noise | Without noise | With noise |
| Proposed LSTM network deviation | 0.008 | 0.135 | 0.9 | 1.623 |
| Traditional LSTM network [10] | 0.021 | 0.216 | 2.1 | 3.779 |
| Neural network on radial basis functions [8] | 0.063 | 0.335 | 4.9 | 8.117 |
| Three-layer perceptron [9] | 0.119 | 0.519 | 7.8 | 14.035 |

From Table 4 it can be seen that the modified LSTM network demonstrates the best performance among the considered architectures: without noise its standard deviation is 0.008, and with noise it is only 0.135, while the absolute error without noise is 0.9%, and with noise it is 1.623%; the traditional LSTM is inferior with deviations of 0.021 (without noise) and 0.216 (with noise) and errors of 2.1% and 3.779%, respectively; the network on radial basis functions shows higher deviations of 0.063/0.335 and errors of 4.9%/8.117%.

In turn, the three-layer perceptron has the worst results, with deviations of 0.119 (without noise) and 0.519 (with noise) and absolute errors of 7.8% and 14.035%, which emphasizes the stability and noise resistance of the proposed LSTM architecture.

## 5. Discussions

A hybrid model for the helicopter TE gas temperature sensor readings predicting and correcting block diagram has been developed using a neural network and the CPS infrastructure (see Figure 1). The system implements a closed loop: measurement → preprocessing → predicting → correction

→ adaptation. At the edge device level, "raw" data $y_{meas}(t)$ are collected, which contain the true temperature value $y_{true}(t)$, drift $d(t)$, and noise $n(t)$, according to model (1).

The noise is approximated by a Gaussian process (2), and the drift is described by a stochastic model (3). Preprocessing includes smoothing using an RC filter (4), and drift detection is based on comparing the signal with the reference profile $y_{ref}(t)$ (5) and applying threshold logic (6).

The LSTM network predicts the "net" temperature based on a feature vector including the signal derivative, drift, and engine operating parameters $P(t)$, as shown in (7)–(8). The readings correction is implemented by merging the prediction, measurement, and drift estimate according to (10) with a confidence weighting coefficient $k \in [0,1]$.

The CPS infrastructure use ensures secure two-way exchange between the edge and cloud modules, where data storage, analytics, and the neural network automatic retraining on new data with regular resending of the updated model on board are implemented (see equations (11)–(17)). Retraining includes the loss function optimization with a regularizer and a "forgetting" factor (14), (16), while the weights are updated using the stochastic gradient descent method (15), and the resending occurs when a specified threshold gain ε_th (17) is exceeded. Thus, the model provides an adaptive, drift- and noise-resistant temperature predicting system capable of self-correction and real-time learning.

According to the computational experiment results, Figure 6 shows the "raw" gas temperature readings $y_{meas}(t)$, including the true value $y_{true}(t)$, drift $d(t)$, and noise $n(t)$ (1) comparison, with the corrected values $y_{corr}$ obtained taking into account the neural network predicting and drift estimate. The original signal (thin blue curve) fluctuates within 1080...1150 K with noise spikes and increasing drift, while the corrected curve (thick red line) effectively eliminates drift and smooths out noise, confirming the filtering correctness (4) and data fusion algorithm (10). This is achieved through the RC filter and weighted correction algorithm (confidence coefficient $k$), which provides "cleaner" temperature estimates for the control system. Figure 7 shows the drift estimate $d(t)$ based on the comparison with the reference profile (5) and the threshold logic (6), which accurately reproduces the characteristics of the true drift: linear growth (≈0.02 °C/s), sinusoidality (period ≈62 seconds), with a small delay (~1 second) and smoothed amplitude. This confirms the algorithm's high sensitivity and its suitability for drift compensation in real time.

Figure 8 shows that the telemetry transmission delay in the CPS infrastructure depends on the packet size S(t) according to the linear model Latency = $L_0$ + S/B + ε(t) (12): as the size increases from 100 to 2000 Kbit, the delay increases from ~100 to ~1400 ms. This indicates the need to optimize the computations between the edge and cloud modules (14)–(17).

Table 3 demonstrates that the modified LSTM with a drift gate (20), (21) outperforms traditional LSTM, RBF networks, and perceptrons in accuracy, recall, and F1-score due to adaptive weight updating (16), (23) and drift accounting (8).

Table 4 confirms its robustness to noise (σ = 0.025): RMSE increases from 0.008 to 0.135 (error from 0.9% to 1.6%), which is significantly better than that of analogues. This is explained by the new data priority ($\lambda$ in (16)) and built-in distortion compensation.

As a result, the integration of the extended LSTM cell (19)–(23), the "drift-gate" module (20), and the CPS infrastructure (11)–(17) ensures high accuracy and reliability in predicting and correcting the helicopter TE gas temperature.

Along with the significant results of the study, the following limitations should be noted, presented in Table 5. Table 6 presents prospects for further research.

Table 5
The results obtained limitations

| Number | Limitation | Description |
|---|---|---|
| 1 | Limited generalizability to other engine | The model was trained and tested on data obtained from TV3-117 engines on the Mi-8MTV helicopter in the nominal operating mode and at a 2500 meters fixed altitude. Under other |

| | | |
|---|---|---|
| | types and operating modes | conditions (different engines, changed flight modes, extreme temperatures, high loads), the noise, drift, and gas temperature dynamics parameters may differ significantly, which will require additional training or adaptation of the neural network architecture and preprocessing algorithms. |
| 2 | Dependence on reference profile and training data quality | The drift detection module requires a predefined reference profile $y_{ref}(t)$ to function correctly. If the reference profile does not accurately reflect the actual engine dynamics (e.g., due to component wear or incorrect thermocouple calibration), the algorithm might misinterpret natural temperature variations as drift or, conversely, miss real shifts in readings. Training an LSTM network requires a high-quality labeled data; insufficient or biased representation in the training dataset limits the prediction accuracy outside the training domain. |
| 3 | Limitations on computational resources and latencies in the CPS infrastructure | On board the helicopter, computing resources (CPU, memory) are limited. A complex LSTM model with a "drift gate" and additional calculations for drift estimation may require more resources than are available in real time, which leads to an increase in the prediction and correction delay. Considering that the model shows that the data transmission delay via the CPS channel can reach more than 1 second with a load of about 2000 Kbps, in extreme scenarios (e.g., in emergency modes), the updated model may simply not have time to be delivered on board in time, and the readings correction may be performed with a delay, which reduces its practical effectiveness. |
| 4 | Simplified a priori assumptions about the noise and drift nature | In the study, the measurement noise $n(t)$ is represented as a Gaussian process, and the thermocouple drift $d(t)$ is modeled by a stochastic diffusion process with a specific sinusoidal superposition. In real engine conditions, more complex and non-stationary sources of errors are possible: nonlinear effects of corrosion, thermal shocks, microcracks in the thermocouple, and electromagnetic interference. Accordingly, the algorithm may underestimate or misinterpret anomalies that go beyond the adopted a priori models, which reduces the estimates' accuracy in real field tests. |

Table 6
Prospects for further research

| Number | Research | Action |
|---|---|---|
| 1 | Expanding the application scope to different engine types and operating modes [26] | Adaptation and validation of the proposed hybrid drift detection model for other gas turbine engines and for different flight regimes.<br>Study of the altitude, temperature, and load variations' influence on predicting accuracy and methods development for the model parameters' automatic reconfiguration. |
| 2 | Improving the reference profile and training data formation [27] | Develop algorithms for automatic generation and adaptive updating of the reference profile ($y_{ref}$) based on current operational data to reduce the impact of outdated or inaccurate calibration curves.<br>Expanding the training set through field testing and simulating scenarios with different sensor wear states; |

|  |  |  |
|---|---|---|
|  |  | implementing data augmentation methods to model unexpected temperature fluctuations. |
| 3 | Optimizing computational complexity and implementing it on the onboard computing system | Research into more compact neural network architectures (e.g., lightweight LSTM, TCN, or Transformer-Lite) and weight quantization to reduce memory and CPU costs.<br>Development of a prototype taking into account the on-board controllers' specifics (microcontrollers or single-board computers) and testing the system's response time during real data exchange via the CPS channel. |
| 4 | Modeling and accounting for more complex nonlinear sources of noise and drift | The stochastic models' construction takes into account non-stationary corrosion processes, transient thermal effects (thermal shocks), electromagnetic interference, and destruction of thermocouple insulation.<br>To investigate methods for anomaly adaptive detection that are not specific to predefined sinusoidal patterns using hybrid approaches (e.g., combining LSTM with variational autoencoders or GANs). |
| 5 | Integrating feedback and self-learning mechanisms in real time | Develop online learning methods that enable the model to adjust its parameters based on continuously incoming data on real measurements and maintenance results.<br>The adaptive algorithm implementation automatically reconfigures itself after maintenance or when systematic deviations are detected, reducing the need for manual reconfiguration. |
| 6 | Developing the concept of a "digital twin" and integrating it with the general monitoring system | A single platform creation combining engine thermodynamic models, drift detection algorithms, and other diagnostic modules (e.g., vibration analysis, wear assessment of parts).<br>Ways to investigate transmitting aggregated data to ground-based maintenance centers and using cloud computing resources for deeper analytics and unit life prediction. |

## 6. Conclusions

The helicopter turboshaft engines' gas temperature sensor readings hybrid model for predicting and correcting structural diagrams has been developed and implemented. It combines an extended LSTM cell with a "drift gateway" and a CPS infrastructure. The model takes into account stochastic descriptions of noise and thermocouple drift, applies an RC filter for preliminary processing, and a built-in "drift gateway" inside the LSTM allows predicting the "net" temperature value, taking into account the accumulated bias.

The CPS infrastructure use ensures encrypted two-way "edge ↔ cloud" data transfer and the neural network regular retraining, taking into account the "forgetting" new data factor, which guarantees the model adaptability to changes in engine operating modes.

As the computational experiments result, the corrected readings demonstrate effective drift removal and noise spike smoothing compared to the raw data, which confirms the preprocessing algorithms' (RC filter) correctness and correction via weighted fusion. The drift estimation accurately reproduces the linear trend and the true drift sinusoidal oscillations with about 1 second delay and reduced amplitude (~0.45 °C), which proves the proposed detection logic's high sensitivity and adequacy.

The neural network architectures comparative analysis showed that the modified LSTM with a "drift gate" outperforms the traditional LSTM, RBF network, and three-layer perceptron in all metrics: the accuracy (0.991), recall (0.985), and F1-measure (0.986) of the proposed model are significantly higher than their analogues. When introducing additive noise σ = 0.025, the modified LSTM retains low RMSE (0.135) and absolute error (1.623%), which confirms its resistance to low external interference.

The CPS channel characteristics study showed that the data transmission delay increases linearly with the telemetry volume: from ~60 ms at S < 100 kbps to ~1400 ms at S S≈2000 kbps, which emphasizes the need to balance computations between onboard modules and cloud analytics. The proposed CPS implementation allows for automatic retraining, taking into account the "forgetting factor" and updating the onboard model when the gain threshold $\varepsilon_{th}$ is exceeded, but the obtained latencies indicate potential limitations under extreme loads, requiring further optimization of transmission protocols and the neural network architecture simplification.

## Acknowledgements

The research was supported by the Ministry of Internal Affairs of Ukraine "Theoretical and applied aspects of the development of the aviation sphere" under Project No. 0123U104884.

The research was carried out with the grants support of the National Research Fund of Ukraine: "Methods and means of active and passive recognition of mines based on deep neural networks", project registration number 273/0024 from 1/08/2024 (2023.04/0024), and "Information system development for automatic detection of misinformation sources and inauthentic behaviour of chat users", project registration number 187/0012 from 1/08/2024 (2023.04/0012).

## Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

## References

[1] W. Liu, G. Xu, X. Gu, J. Yao, M. Li, M. Lei, Q. Chen, Y. Fu, Experimental Analysis and Thermodynamic Modeling for Multilevel Heat Exchange System with Multifluid in Aero Engines. Energy 315 (2025) 134373. doi: 10.1016/j.energy.2025.134373.

[2] B. Jiang, K. Zhang, Y. Lu, Q. Miao, Fault Diagnosis and Fault-Tolerant Control of Helicopters, Reference Module in Materials Science and Materials Engineering. Elsevier, 2024. doi: 10.1016/b978-0-443-14081-5.00006-4.

[3] Y. Wang, J. Zhao, R. Zhao, Shape Parameterization Optimization of Thermocouples Used in Aeroengines, Aerospace 10:2 (2023) 202. doi: 10.3390/aerospace10020202.

[4] H. Zhao, X. Lin, Z. Liao, M. Xu, Y. Yao, B. Duan, Z. Song, Highly fault-tolerant thrust estimation for gas turbine engines via feature-level dissimilarity design, Measurement 244 (2025) 116350. doi: j.measurement.2024.116350.

[5] S. Vladov, A. Sachenko, V. Sokurenko, O. Muzychuk, V. Vysotska, Helicopters Turboshaft Engines Neural Network Modeling under Sensor Failure, Journal of Sensor and Actuator Networks 13:5 (2024) 66. doi: 10.3390/jsan13050066.

[6] W. Cui, R. Wang, T. Sun, Z. Liu, Managing remaining useful life of cyber-aeroengine systems using a graph spatio-temporal attention recurrent network with phase-lag index, Energy 308 (2024) 132924. doi: 10.1016/j.energy.2024.132924.

[7] O. Balli, Exergetic, sustainability and environmental assessments of a turboshaft engine used on helicopter, Energy 276 (2023) 127593. doi: 10.1016/j.energy.2023.127593.

[8] H. Chen, Q. Li, Z. Ye, S. Pang, Neural Network-Based Parameter Estimation and Compensation Control for Time-Delay Servo System of Aeroengine, Aerospace 12:1 (2025) 64. doi: 10.3390/aerospace12010064.

[9] Y.-P. Zhao, W. Cai, The perceptron algorithm with uneven margins based transfer learning for turbofan engine fault detection, Engineering Applications of Artificial Intelligence, 127 (2024) 107249. doi: 10.1016/j.engappai.2023.107249.

[10] S. R. Kumar, J. Devakumar, Recurrent neural network based sensor fault detection and isolation for nonlinear systems: Application in PWR, Progress in Nuclear Energy 163 (2023) 104836. doi: 10.1016/j.pnucene.2023.104836

[11] Z. Jiang, S. Yang, X. Wang, Y. Long, An Onboard Adaptive Model for Aero-Engine Performance Fast Estimation, Aerospace 9:12 (2022) 845. doi: 10.3390/aerospace9120845.

[12] Q. Chen, H. Sheng, T. Zhang, An improved nonlinear onboard adaptive model for aero-engine performance control, Chinese Journal of Aeronautics 36:10 (2023) 317–334. doi: 10.1016/j.cja.2022.12.005.

[13] Z. Wei, S. Zhang, S. Jafari, T. Nikolaidis, Gas turbine aero-engines real time on-board modelling: A review, research challenges, and exploring the future, Progress in Aerospace Sciences 121 (2020) 100693. doi: 10.1016/j.paerosci.2020.100693.

[14] Y. Yang, X. Li, Z. Yang, Q. Wei, N. Wang, L. Wang, The Application of Cyber Physical System for Thermal Power Plants: Data-Driven Modeling, Energies 11:4 (2018) 690. doi: 10.3390/en11040690.

[15] M. de Castro-Cros, M. Velasco, C. Angulo, Machine-Learning-Based Condition Assessment of Gas Turbines—A Review, Energies 14:24 (2021) 8468. doi: 10.3390/en14248468.

[16] T. Castiglione, D. Perrone, J. Song, L. Strafella, A. Ficarella, S. Bova, Linear model of a turboshaft aero-engine including components degradation for control-oriented applications, Energies 16:6 (2023) 2634. doi: 10.3390/en16062634

[17] O. Melnychenko, L. Scislo, O. Savenko, A. Sachenko, P. Radiuk, Intelligent Integrated System for Fruit Detection Using Multi-UAV Imaging and Deep Learning, Sensors 24:6 (2024) 1913. doi: 10.3390/s24061913.

[18] S. Omatu, Classification of Mixed Odors Using A Layered Neural Network, International Journal of Computing 16:1 (2017) 41–48. doi: 10.47839/ijc.16.1.870.

[19] S. Vladov, L. Scislo, V. Sokurenko, O. Muzychuk, V. Vysotska, A. Sachenko, A. Yurko, Helicopter Turboshaft Engines' Gas Generator Rotor R.P.M. Neuro-Fuzzy On-Board Controller Development, Energies, 17:16 (2024), 4033. doi: 10.3390/en17164033.

[20] A. Boujamza, S. Lissane Elhaq, Attention-based LSTM for Remaining Useful Life Estimation of Aircraft Engines, IFAC-PapersOnLine 55:12 (2022) 450–455. doi: 10.1016/j.ifacol.2022.07.353

[21] M. Sirola, J. E. Hulsund, Machine-Learning Methods in Prognosis of Ageing Phenomena in Nuclear Power Plant Components. International Journal of Computing 20:1 (2021) 11–21. doi: 10.47839/ijc.20.1.2086.

[22] Z. A. Kakarash, H. S. Ezat, S. A. Omar, N. F. Ahmed, Time Series Forecasting Based on Support Vector Machine Using Particle Swarm Optimization. International Journal of Computing 21:1 (2022) 76–88. doi: 10.47839/ijc.21.1.2520.

[23] M. Duhan, P. K. Bhatia, Software Reusability Estimation based on Dynamic Metrics using Soft Computing Techniques, International Journal of Computing 21(2) (2022) 188–194. doi: 10.47839/ijc.21.2.258.

[24] I. Obeidat, M. AlZubi, Developing a Faster Pattern Matching Algorithms for Intrusion Detection System, International Journal of Computing 18:3 (2019) 278–284. doi: 10.47839/ijc.18.3.1520.

[25] S. Vladov, L. Scislo, V. Sokurenko, O. Muzychuk, V. Vysotska, S. Osadchy, A. Sachenko, Neural Network Signal Integration from Thermogas-Dynamic Parameter Sensors for Helicopters Turboshaft Engines at Flight Operation Conditions, Sensors 24:13 (2024) 4246. doi: 10.3390/s24134246.

[26] M. Ishaq Khan, L. Maccioni, F. Concli, Energy and Exergy Analysis of Conventional Automobile Engines: Evaluation of Waste Heat Recovery Potential to Drive Parasitic Loads, Energies 18:13 (2025) 3264. doi: 10.3390/en18133264.

[27] A. Brusa, A. Grossi, M. Lenzi, F. P. Shethia, N. Cavina, I. Kitsopanidis, Modeling of Exhaust Gas Temperature at the Turbine Outlet Using Neural Networks and a Physical Expansion Model, Energies 18:7 (2025) 1721. doi: 10.3390/en18071721.