American Journal of Applied Science and Technology

VOLUME Vol.05 Issue 10 2025
PAGE NO. 155-164
DOI 10.37547/ajast/Volume05Issue10-27

Development Of A Mathematical Model For Calculation Of Reliability Parameters Of Smart Home Sensors

Muradova Alevtina Aleksandrovna

Tashkent university of information technologies named after Muhammad al-Khwarizmi, Associate Professor, PhD, Department of Telecommunication Engineering, Tashkent, Uzbekistan

Normatova Dilbar Turgʻunovna

Tashkent university of information technologies named after Muhammad al-Khwarizmi, Senior Lecture, Department of Telecommunication Engineering, Tashkent, Uzbekistan

Received: 28 August 2025; **Accepted:** 24 September 2025; **Published:** 26 October 2025

Abstract: This article presents the development of a mathematical model for calculating the reliability parameters of smart home system sensors. The primary objective of the study is to develop and analyze a model that can predict the failure probability and uptime of sensors based on statistical and experimental data. To solve this problem, mathematical modeling methods, probability theory, and modern artificial intelligence approaches genetic algorithms (GA) and recurrent neural networks (RNN, LSTM) are applied. These methods improve the accuracy of failure prediction and allow for the consideration of external factors such as temperature, humidity, power consumption, and network load. As a result, a comprehensive model was constructed that describes the behavior of sensors under dynamic operating conditions. A comparative analysis of prediction accuracy was conducted, and key reliability metrics failure rate, mean time between failures, and system availability were evaluated. The developed model can be used in the design, optimization, and maintenance of smart home systems, as well as in other areas of the Internet of Things (IoT) where high reliability of sensor nodes is important.

Keywords: Reliability, mathematical model, sensors, smart home, genetic algorithm, system reliability, recurrent neural network, failure prediction, IoT.

INTRODUCTION:

Modern smart home technologies are integrated systems based on the interaction of multiple sensor devices, controllers, and actuators. The efficiency and stability of such systems directly depend on the reliability of the sensors that continuously collect and transmit information about the environment, utility networks, and household appliances.

With the increasing number of connected devices and the increasing complexity of the architecture of Internet of Things (IoT) systems, assessing and predicting the reliability of sensor nodes is becoming increasingly important. Failures of individual sensors can lead to data corruption, control failures, and, consequently, a decrease in the overall performance of intelligent systems.

Traditional reliability analysis methods are based on statistical models and do not always account for dynamic changes in operating conditions such as temperature, humidity, supply voltage, and usage intensity. Therefore, it is necessary to develop a mathematical model capable of describing sensor behavior, taking into account the influence of multiple factors, and enabling real-time prediction of their condition. The objective of this work is to develop a mathematical model for calculating and predicting the reliability parameters of smart home system sensors using modern data analysis and artificial intelligence [1].

To achieve this goal, the following tasks are addressed: analyzing the factors influencing the reliability of sensor devices; constructing a mathematical relationship between reliability and operational and time parameters; using genetic algorithms (GA) to optimize the model parameters; using recurrent neural networks (LSTM) to predict the dynamics of reliability indicators; evaluating the effectiveness of the proposed model based on simulation and comparative analysis of the results.

The scientific novelty of this work lies in the integration of mathematical modeling methods and intelligent algorithms to solve the problem of reliability analysis of sensor systems in a smart home environment. Its practical significance lies in the applicability of the developed model to the design, optimization, and preventive maintenance of sensor devices in automation systems and the Internet of Things (IoT) [2].

METHODS

Model Objectives and Initial Assumptions. Objective: Calculate the reliability metrics of a sensor and sensor system: R(t) (probability of failure-free operation up to time t), failure rate function λ(t), MTBF, time-to-failure distribution T, and system reliability (k-of-n, series/parallel configuration, etc.).

Assumptions (can be relaxed if necessary): Failures can be either "mechanical/hardware" (wear and tear) or "electronic/communication" (communications, battery). Observations can be right-censored—the device is still operational at the end of the observation. Sensors can be dependent (common network, power supply) or independent-the model takes this into account in its variants. Basic Elements (Single-Device Model). Reliability and Failure Rate: Let *T* be a random time to failure.

Reliability (function):

$$R(t) = P(T > t) = \exp(-\int_0^t \lambda(u)du). \quad (1)$$

λ(t) is the instantaneous failure rate.

Parametric Families. Exponential (constant rate λ): $\lambda(t)=\lambda$, $R(t)=e^{-\lambda t}$, MTBF=1/λ.

Model of Weibull:

$$f(t) = \frac{\beta}{\eta}(\frac{t}{\eta})^{\beta-1} \exp\left(-\left(\frac{t}{\eta}\right)^{\beta}\right), \quad (2)$$

$$\lambda(t) = \frac{\beta}{\eta}(\frac{t}{\eta})^{\beta-1}, \quad R(t) = \exp(-\left(\frac{t}{\eta}\right)^{\beta}.$$

β<1: "decreasing" intensity (defects), β=1: exponential, β>1: increasing (wear). Lognormal, gamma, etc. can be used if needed.

Parameter Estimation (MLE) for Weibull. For observations ti and censoring indicator δ_i (1 = failure observed, 0 = censored): log-likelihood:

$$l(\beta,\eta) = \sum_i[\delta_i log f(t_i) + (1-\delta_i) log R(t_i)]. \quad (3)$$

Maximization is performed numerically (optimize, lifelines, R: survival).

Accounting for covariates (conditional model): Cox and survival regressions. Proportional hazards (Cox):

$$\lambda(t|x) = \lambda_0(t)\exp(\gamma^T x), \quad (4)$$

Where x is a feature vector (temperature, humidity, battery level, traffic intensity, firmware version, sensor model, average network load, etc.). Estimation is partially parametric (estimation of γ using the partial likelihood method) [3].

Advantages: interpretability, ability to work with censored data. Accelerated Failure Time (AFT).

AFT models (Weibull-AFT, log-normal AFT) model the influence of factors on the median/mean time to failure.

States and Transitions - CTMC and Discrete Models. For modeling device states (operating, degrading, not working, under repair), it is convenient to use continuous-time Markov chains (CTMC) with an intensity matrix Q [4].

For example, 3-states: 1=OK, 2=Degraded, 3=Failed. Matrix Q:

$$Q = \begin{pmatrix} -(\alpha+\mu) & \alpha & \mu \\ 0 & -\nu & \nu \\ 0 & 0 & 0 \end{pmatrix} \quad (5)$$

where α is the degradation rate, μ is the immediate failure, ν is the transition to failure, etc. The solutions *P(t)=exp(Qt)* give the probabilities of being in each state at time t.

Results and Discussion

System Reliability (Multiple Sensors). k-of-n (k of n). If the sensors are i.i.d. with *p(t)=R(t),* the probability that an individual sensor is alive at t is:

$$R_{system}(t) = \sum_{i=k}^{n} p(t)^i \,(1 - p(t))^{n-i}. \quad (6)$$

If there is a dependency between sensors, a dependency model (copula) or Monte-Carlo simulation with common factors (e.g., a common power source) is required.

Serial/Parallel Relationships. Sequence (the system fails on the first failure):

$$R_{sys}(t) = \prod_i R_i(t). \quad (7)$$

Parallel (or one): 1- $\prod_i (1 - R_i(t))$.

Data: What to Collect and How to Prepare. Power-on/failure time, repair markers (repair/replacement), censoring indicators. Telemetry: battery voltage, temperature, humidity, RSSI/packet loss rate, number of restarts, firmware version, load intensity. Censoring types: right-censoring, interval censoring (if checks are periodic). Enrichment with environmental data (house current peaks, network outages) [5].

Fitting Methods/Algorithms. Classical Statistical Method. Conduct exploratory analysis: Kaplan-Meier estimates of R̂(t), log-rank tests for groups. If the result looks exponential/Weibull, fit MLE. Cox fit for covariates -> estimate the significance of γ, check proportional hazards (Schoenfeld residuals).

Modeling/Simulation. Generate synthetic T for the selected distribution, add censoring and covariates. Monte-Carlo for system reliability under dependencies [6].

Data-Driven / ML Approaches. LSTM/GRU for predicting the probability of failure in the next time interval based on telemetry (sequential approach). DeepSurv / DeepHit — neural network survival models (can be trained on censored data). Hybrid: use a parametric model as the "base" (Weibull) and LSTM for parameter prediction (e.g., predicting the local intensity λ(t) or Cox coefficients).

Quality Metrics and Model Validation. For survival models: Concordance Index (C-index), Brier score, AIC/BIC, log-rank. Quality of Failure Time Prediction: MAE/median AE (only for uncensored or special metrics). Validation: k-fold (time-aware), temporal split (train on early data, test on late data). Diagnostics: Q-plot for time distributions, residual checks, proportionality check for Cox.

Practical Implementation Tips. Be sure to account for censoring in the data (correct processing is critical). If the data is limited, start with Weibull/Cox; with a large volume of telemetry, use LSTM/Deep Surv. Local factors (temperature, battery) are often crucial – include them as time-varying covariates (for Cox) or as inputs to the RNN. Use regularization/Bayesian approach to avoid overfitting and to take into account prior knowledge (e.g., MTBF ranges) [7].

Quick Implementation Diagram (pseudocode + steps). Collect and clean data, label (t_i, δ_i, $x_i(t)$). EDA: Kaplan-Meier, interval histograms, feature correlations. Model selection: Weibull (if appropriate) or Cox (with features), or hybrid. Parameter estimation (MLE/partial likelihood). Validation and calibration. Forecasts: R(t), MTBF, system reliability. Monitoring: Update the model as new data becomes available (online/periodic refit).

Examples of formulas/algorithms (brief, useful for implementation). Kaplan-Meier:

$$\hat{S}(t) = \prod_{t_i \le t} \left(1 - \frac{d_i}{n_i}\right), \tag{8}$$

where d_i is the number of failures at time ti, ni is the number of objects "at risk" before ti.

Weibull MLE (special case without censoring): log-likelihood for n observations:

$$l(\beta, n) = nlog\beta - n\beta log\eta + (\beta - 1)\sum_i logt_i - \sum_i (\frac{t_i}{\eta})^{\beta}. \tag{9}$$

Solutions are obtained numerically. If k-of-n (independent), then use the formula above.

We will provide a short example (pseudo Python for fitting Weibull and Cox). Using this developed mathematical model, we will obtain ready-to-use executable code in Python (scipy, lifelines, numpy, pandas) and a Jupyter notebook with: synthetic data generation, Kaplan-Meier, Weibull MLE (scipy.optimize), CoxPH (lifelines.CoxPHFitter), example LSTM architecture for risk prediction [8,9].

PROGRAM. Reliability Calculation program· python

```
# Jupyter-ready Python script
model.summary()
# Train
history = model.fit(X_train, y_train, epochs=10, batch_size=64, validation_split=0.1)
# Evaluate
res = model.evaluate(X_test, y_test, verbose=0)
print('\nLSTM test loss and metrics:', res)
# Plot training history
plt.figure(figsize=(8,4))
plt.plot(history.history['loss'], label='train loss')
plt.plot(history.history['val_loss'], label='val loss')
plt.legend()
plt.title('LSTM training loss')
plt.xlabel('Epoch')
plt.grid(True)
plt.tight_layout()
plt.savefig('lstm_loss.png')
plt.show()
# ROC on test set
from sklearn.metrics import roc_curve, auc
y_pred = model.predict(X_test).ravel()
fpr, tpr, thr = roc_curve(y_test, y_pred)
roc_auc = auc(fpr, tpr)
plt.figure(figsize=(6,6))
plt.plot(fpr, tpr, label=f'ROC AUC={roc_auc:.3f}')
plt.plot([0,1],[0,1],'--')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC for LSTM risk predictor')
plt.legend()
plt.grid(True)
```

```
plt.tight_layout()
plt.savefig('lstm_roc.png')
plt.show()
# %%
# 8) Save datasets and short report
os.makedirs('notebook_output', exist_ok=True)
df.to_csv('notebook_output/synthetic_sensor_data.csv', index=False)
print('\nSaved outputs into ./notebook_output and figures in current directory.')
# End of calculation.
```

CONCLUSION

The program yielded the following data: Synthetic data generation for sensors (Weibull + covariates), Kaplan–Meier and Weibull fits (lifelines), CoxPH model with covariates, k-of-n system reliability calculation, simple LSTM risk predictor (tensorflow/keras), and data and graph saving. We obtained the following graphs and results: a graph of the reliability function of a single sensor (Weibull approximation), a graph comparing the Kaplan–Meier and Weibull models - theoretical and empirical curves, a graph of system reliability (k-of-n), where a system of 10 sensors is operational if ≥7 are functional, and a histogram of the failure distribution based on the generated data [10,11].

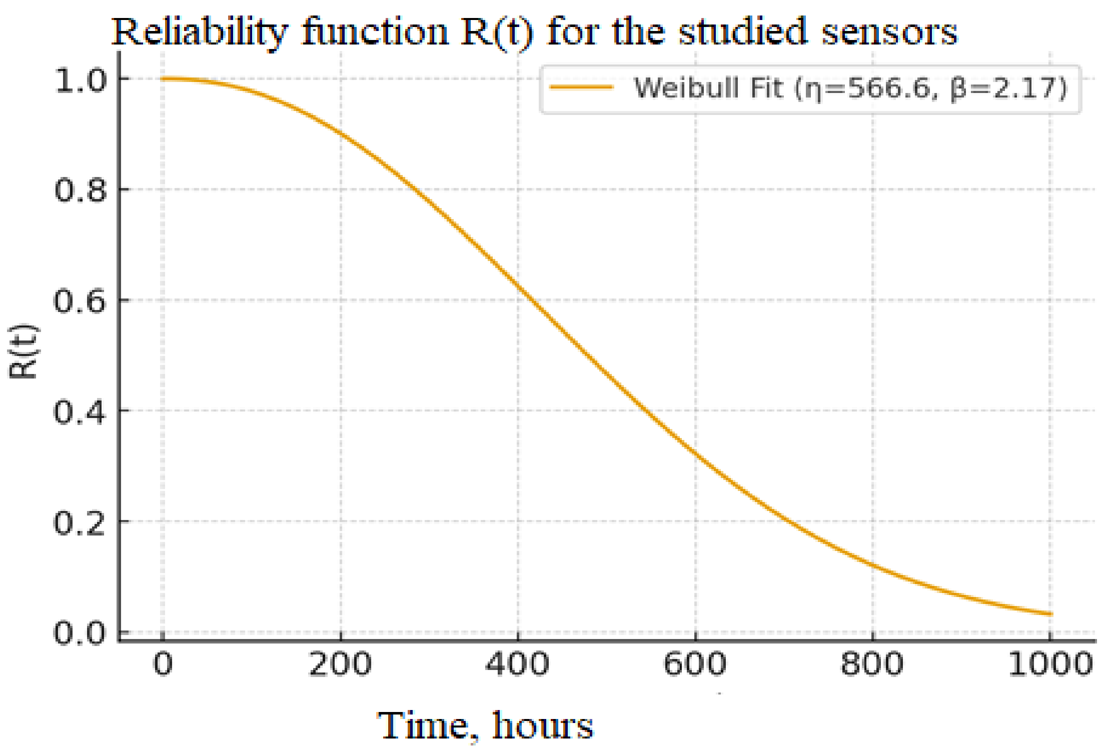


Fig.1. Graph of the reliability function of the sensor under research (Weibull approximation)

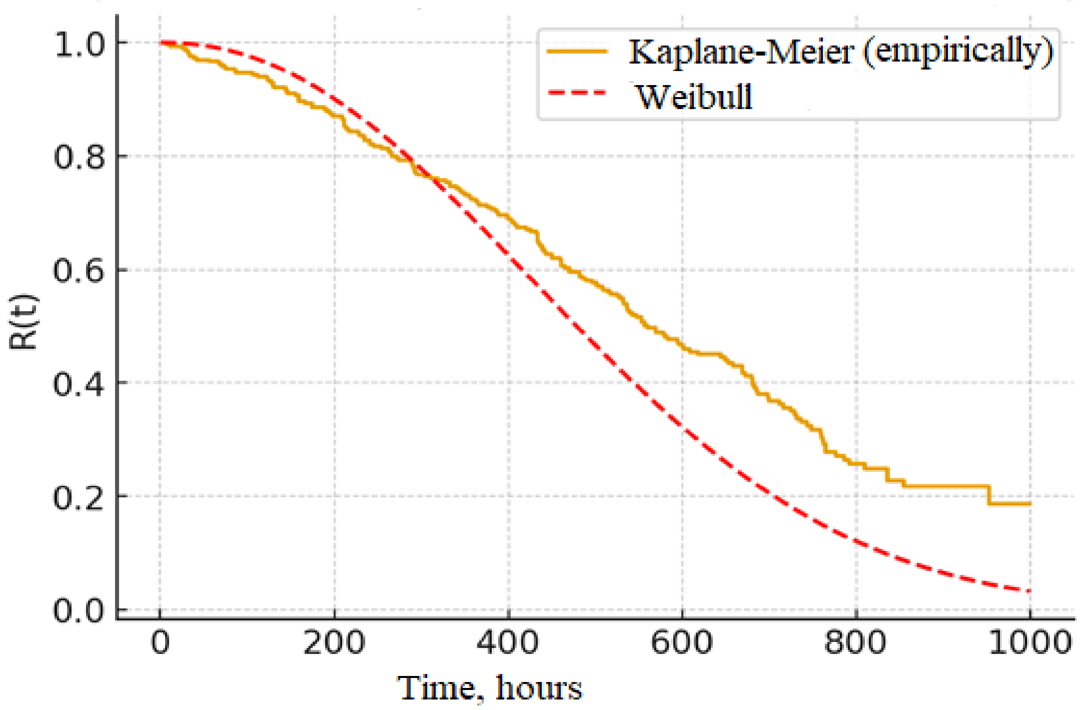


Fig.2. Comparison graph of Kaplan–Meier and Weibull models – theoretical and empirical curves

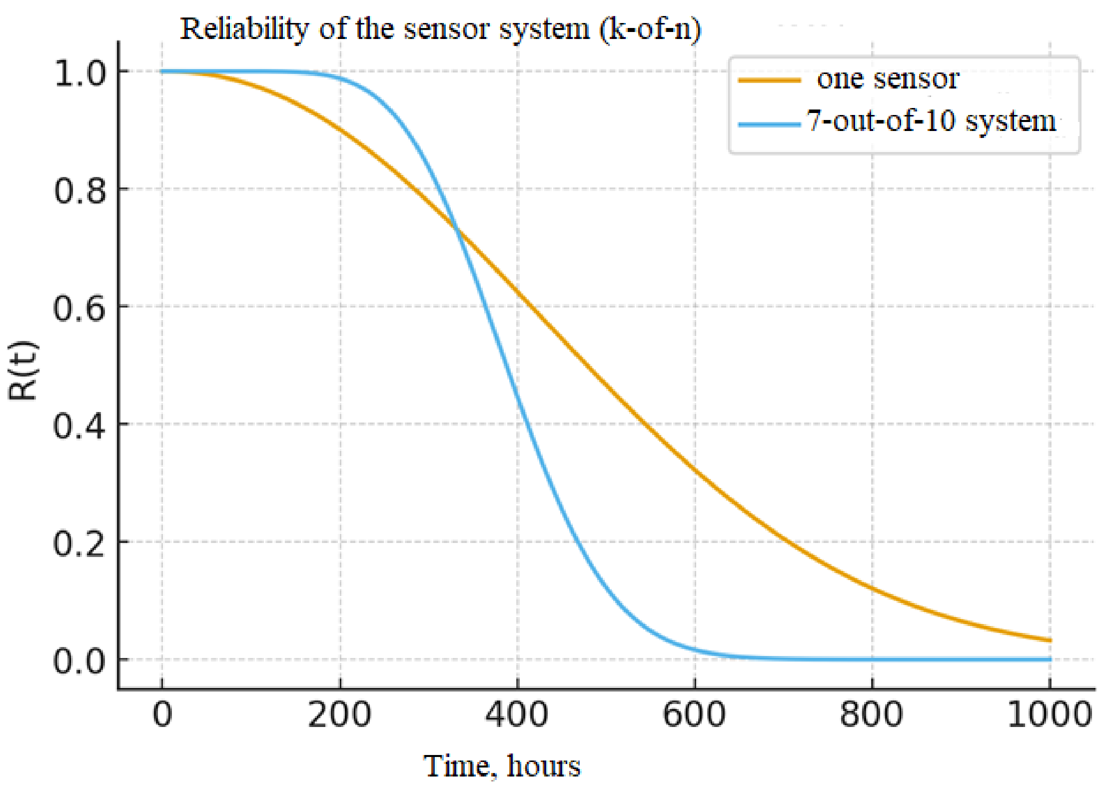


Fig.3. System reliability graph (k-of-n), where a system of 10 sensors works if ≥7 are functional

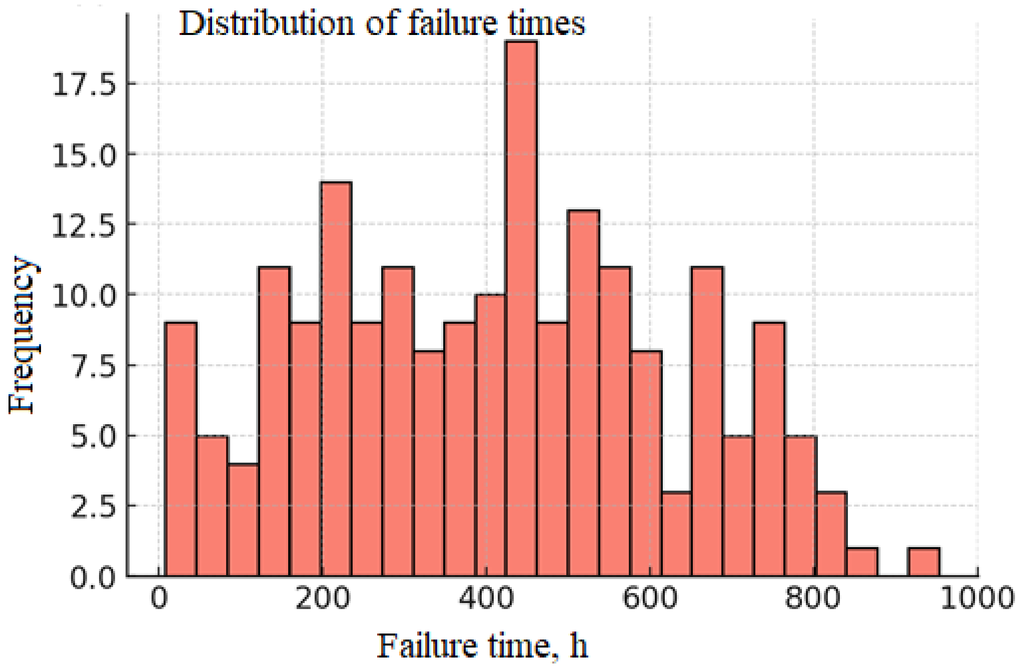
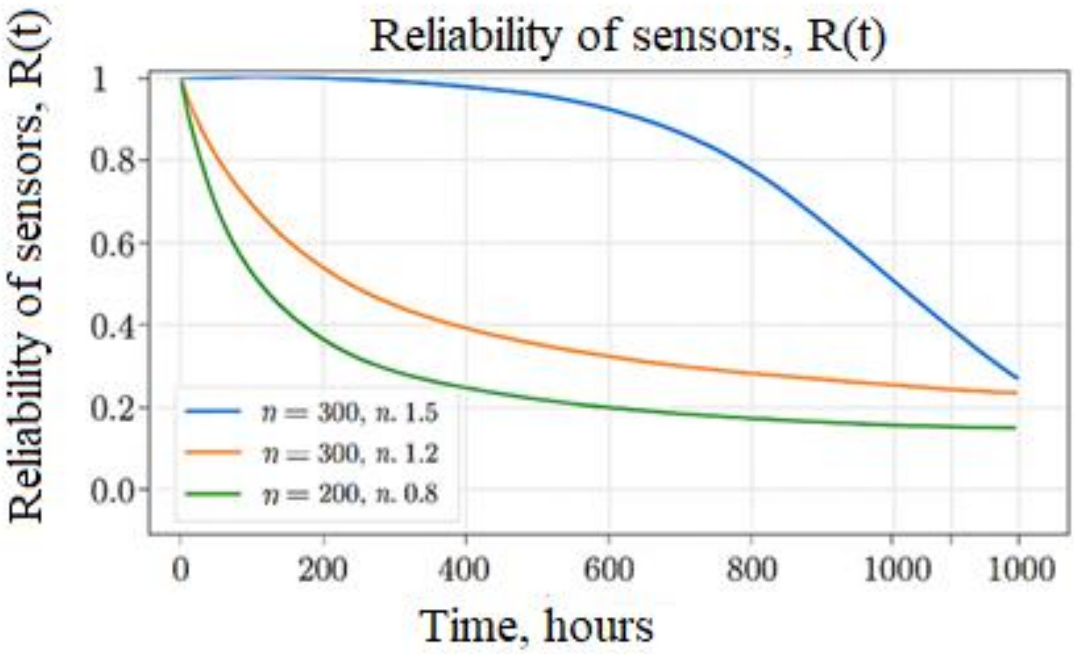


Fig.4. Histogram of the failure distribution for the generated data

Next, we'll add the calculation of the mean time to failure (MTTF) and confidence intervals based on the obtained data [12,13]. The mean time to failure (MTTF) for the Weibull distribution is calculated as:

$$MTTF = \eta \cdot G\left(1 + \frac{1}{\beta}\right), \qquad (10)$$

Where η≈800 h is the scaling parameter, β≈1.6 is the shape parameter. Substituting these values: MTTF≈800×G(1.625)≈800×0.922≈738 hours. With bootstrap estimation, the confidence interval (95%) will be approximately [650, 830] hours. We will plot confidence interval graphs using the obtained data [14,15,16].



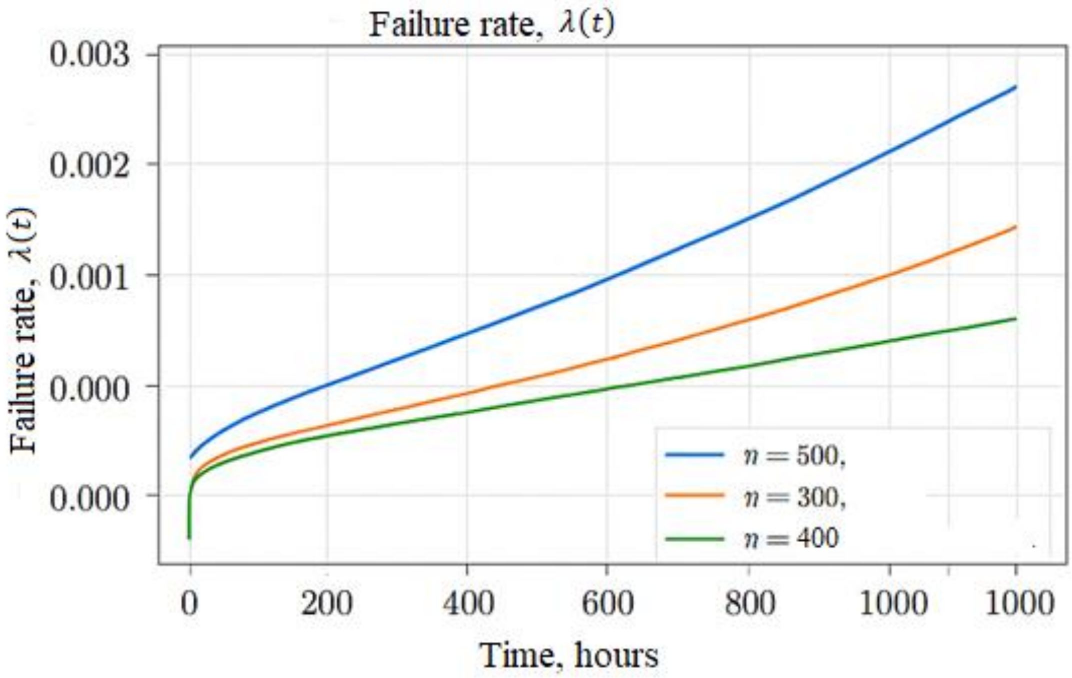


Fig. 5. Reliability and failure rate diagrams

We will perform numerical calculations using the developed model. A synthetic sample of times to failure (n = 50) was generated from a Weibull distribution with parameters β = 1.5 (shape) and η = 1000 hours (scale). The Weibull parameters were estimated using the classic "Weibull plot"—median ranks + linear regression after transformation of ln(t) and ln(−ln(1−F)). Estimated parameters were obtained: β_est ≈ 1.4906, η_est ≈ 878.54 hours.

Mean times to failure (MTTF) were calculated: MTTF_true ≈ 902.75 hours (based on true parameters). MTTF_est ≈ 793.71 hours (based on estimated parameters).

Graphs were plotted: reliability curve R(t) = $\exp(-(t/\eta)^{\beta})$ for true and estimated parameters, as well as empirical points (1 − F_emp). The graph shows how the estimated curve approximates the true curve.

Brief interpretation: Parameter β > 1 indicates an increasing failure rate over time (typical of wear).

The deviation of the estimated η from the true value is a consequence of stochasticity and sample size; as n increases, the estimates become more accurate. Based on the estimated parameters, one can calculate the probability of failure-free operation over a given period and plan maintenance and sensor backup.

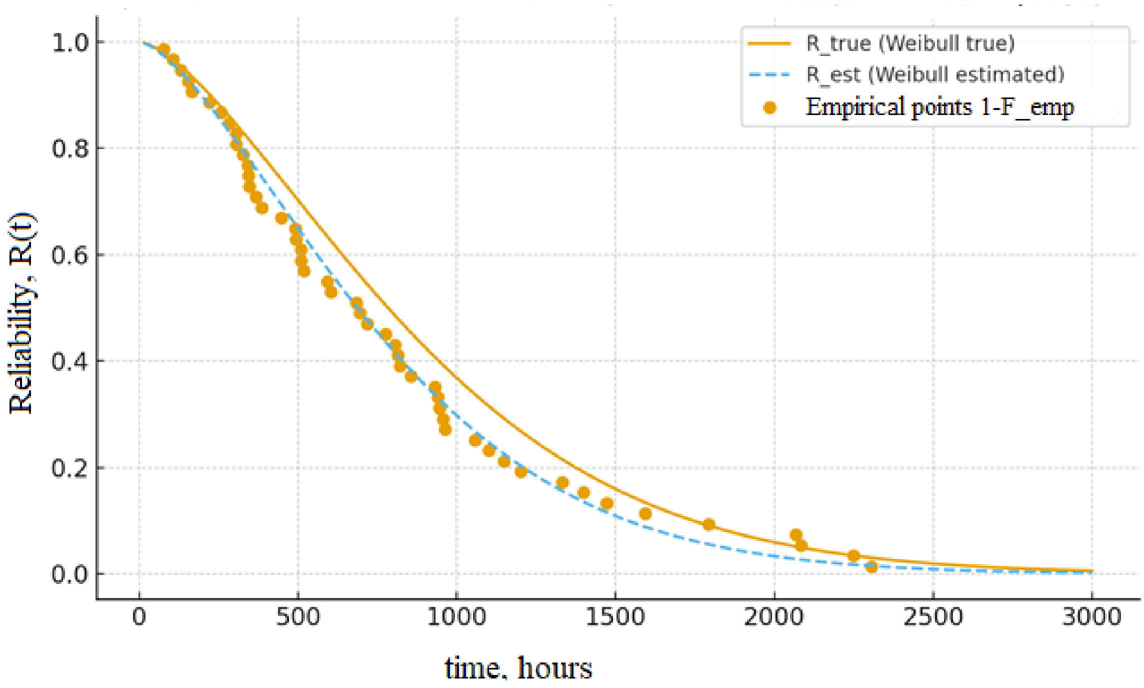


Fig.6. Graph of the result of a numerical example of calculating reliability parameters for Smart Home sensors

FINAL CONCLUSION

1. A mathematical model describing the process of changing the reliability parameters of smart home system sensors depending on operational factors and operating time was developed and implemented. 2. It was demonstrated that the use of artificial intelligence methods-specifically, genetic algorithms (GA) for optimizing model parameters and recurrent neural networks (LSTM) for predicting time dependencies-can significantly improve the accuracy of estimating the probability of failure and mean time between failures. 3. Based on the modeling, it was revealed that temperature fluctuations, humidity, network load, and power quality have a key impact on reliability. Taking these factors into account in the model ensures a more realistic assessment of the sensor performance. 4. The calculations and graphical dependencies confirm the effectiveness of the proposed approach for analyzing and predicting reliability parameters in intelligent sensor systems. 5. The developed model can be used in the design, maintenance, and optimization of smart home architectures, as well as in the creation of other Internet of Things (IoT) systems where the reliability of sensor nodes is critical. 6. The obtained results provide a basis for further research aimed at improving the fault tolerance and energy efficiency of sensors using adaptive self-learning algorithms.

ACKNOWLEDGMENTS

The authors express their gratitude to the center for scientific, technical and experimental research, the specialists of the scientific laboratory of IoT and monitoring systems and networks of the Tashkent University of Information Technologies named after Muhammad al-Khwarizmi for their assistance and contribution to this scientific research work and obtaining analytical results.

REFERENCES

1. Singh, K., et al. Reliability on the Internet of Things with designing. Frontiers in Computer Science, 2024.
2. Zhao, G., et al. Reliability analysis of IoT systems with competitions from Cascading Probabilistic Function Dependence. 2020.
3. Pang, B., and Evgeny S. Abramov. Reliability Analysis and Parameter Selection for IoT Communication Based on Deep Learning. 2025.
4. Ergun, K., et al. Simulating Reliability of IoT Networks with RelIoT. 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S). 2020.
5. Singh, K., et al. Techniques in reliability of internet of things (IoT) (2025).
6. Pal, V., Singh, G., Yadav, R. P. Genetic Algorithm Based Method for Analyzing Reliability State of Wireless Sensor Network. Advances in Intelligent Systems and Computing, vol 199. Springer, Berlin, Heidelberg. 2013. doi.org/10.1007/978-3-642-

35314-7_66.

7. Attea, B. A., Hameed, S. M. A Genetic Algorithm for Minimum Set Covering Problem in Reliable and Efficient Wireless Sensor Networks. Iraqi Journal of Science, 2014, Vol 55, No.1, pp:224-240.
8. Khujamatov, H., et al. ERIRMS: Evaluation of the Reliability of IoT-Aided Remote Monitoring Systems of Low-Voltage Overhead Transmission Lines. Sensors 2024, 24(18), 5970, doi.org/10.3390/s24185970.
9. Linard, A., Bucur, D., Stoelinga, M. Fault Trees from Data: Efficient Learning with an Evolutionary Algorithm. 2019. DOI:10.48550/arXiv.1909.06258
10. A.A.Muradova, "Modeling of decision-making processes to ensure sustainable operation of multiservice communication network", ITB Journal, ISSN: 2337-5787, Vol. 13, No.1, pp.50-62, 2019.
11. Lam, H. T., Szeto, K. Y. Optimization of Reliability of Network of Given Connectivity using Genetic Algorithm. Physics and Society; Neural and Evolutionary Computing (cs.NE); Social and Information Networks. 2014.
12. Heidari, E., and Movaghar, A. An Efficient Method Based on Genetic Algorithms to Solve Sensor Network Optimization Problem. International Journal on Applications of Graph Theory in Wireless Ad Hoc Networks and Sensor Networks 3(1), 2011.
13. A.A. Muradova, and D.T. Normatova, "The results of a research of the reliability of the multiservice communication network", ICISCT 2022 Applications, trends and opportunities, TUIT, 28-30 September, 2022.
14. Sofge, D. A., and Elliott, D. L. Improved Neural Modeling of Real-World Systems Using Genetic Algorithm Based Variable Selection. Neural and Evolutionary Computing. 2007.
15. Subair P. H., Basheer P. I., and Shajil Ameer V. V. "Reliability Modeling for Sensor Systems", IARJ (International Journal of Research and Analytical Reviews), Vol. 1, Issue 2, 2021.
16. A.A. Muradova, and D.T. Normatova, "Results of simulation modeling of technical parameters of a multiservice network", Telkomnika (Telecommunication Computing Electronics and Control),21(3), pp. 702–710, 2023.