VILNIUS TECH
Vilniaus Gedimino technikos universitetas

Mindaugas JANKAUSKAS

INVESTIGATION OF RECURRENT NEURAL NETWORKS-BASED METHODS FOR EARLY FAULT DETECTION AND SHORT-TERM POWER FORECASTING IN WIND ENERGY APPLICATIONS

DOCTORAL DISSERTATION

TECHNOLOGICAL SCIENCES,
ELECTRICAL AND ELECTRONIC ENGINEERING (T 001)

Vilnius, 2026

2026-027-M

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Mindaugas JANKAUSKAS

INVESTIGATION OF RECURRENT NEURAL NETWORKS-BASED METHODS FOR EARLY FAULT DETECTION AND SHORT-TERM POWER FORECASTING IN WIND ENERGY APPLICATIONS

DOCTORAL DISSERTATION

TECHNOLOGICAL SCIENCES,
ELECTRICAL AND ELECTRONIC ENGINEERING (T 001)

Vilnius, 2026

The doctoral dissertation was prepared at Vilnius Gediminas Technical University in 2021–2026.

Supervisor

Prof. Dr Artūras SERACKIS (Vilnius Gediminas Technical University, Electrical and Electronic Engineering – T 001).

The Dissertation Defence Council of the Scientific Field of Electrical and Electronic Engineering of Vilnius Gediminas Technical University:

Chairman

Prof. Dr Vaidotas BARZDĖNAS (Vilnius Gediminas Technical University, Electrical and Electronic Engineering – T 001).

Members:

Prof. Dr Anouar BELAHCEN (Aalto University, Finland, Electrical and Electronic Engineering – T 001),

Dr Vidas RAUDONIS (Kaunas University of Technology, Electrical and Electronic Engineering – T 001),

Prof. Dr Voitech STANKEVIČ (Vilnius Gediminas Technical University, Electrical and Electronic Engineering – T 001),

Assoc. Prof. Dr Eldar ŠABANOVIČ (Vilnius Gediminas Technical University, Electrical and Electronic Engineering – T 001).

The dissertation will be defended at the public meeting of the Dissertation Defence Council of the Scientific Field of Electrical and Electronic Engineering in the Aula Doctoralis Meeting Hall of Vilnius Gediminas Technical University at **10 a. m. on 5 June 2026**.

Address: Saulėtekio al. 11, LT-10223 Vilnius, Lithuania.
Tel.: +370 5 274 4956; fax +370 5 270 0112; e-mail: doktor@vilniustech.lt

A notification on the intended defense of the dissertation was sent on 4 May 2026. A copy of the doctoral dissertation is available for review at the Vilnius Gediminas Technical University repository (https://etalpykla.vilniustech.lt), the Library of Vilnius Gediminas Technical University (Saulėtekio al. 14, LT-10223 Vilnius, Lithuania), and the Wroblewski Library of the Lithuanian Academy of Sciences (Žygimantų g. 1, LT-01102 Vilnius, Lithuania).

Vilnius Gediminas Technical University book No. 2026-027-M
https://doi.org/10.20334/2026-027-M

© Vilnius Gediminas Technical University, 2026
© Mindaugas Jankauskas, 2026
m.jankauskas@vilniustech.lt

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Mindaugas JANKAUSKAS

REKURENTINIAIS NEURONINIAIS TINKLAIS GRĮSTŲ METODŲ TYRIMAS SIEKIANT ANKSTI APTIKTI GEDIMUS IR ATLIKTI TRUMPALAIKES GALIOS PROGNOZES VĖJO ENERGETIKOJE

DAKTARO DISERTACIJA

TECHNOLOGIJOS MOKSLAI,
ELEKTROS IR ELEKTRONIKOS INŽINERIJA (T 001)

Vilnius, 2026

Disertacija rengta 2021–2026 metais Vilniaus Gedimino technikos universitete.

Vadovas

prof. dr. Artūras SERACKIS (Vilniaus Gedimino technikos universitetas, Elektros ir elektronikos inžinerija – T 001).

Vilniaus Gedimino technikos universiteto Elektros ir elektronikos inžinerijos mokslo krypties disertacijos gynimo taryba:

Pirmininkas

prof. dr. Vaidotas BARZDĖNAS (Vilniaus Gedimino technikos universitetas, Elektros ir elektronikos inžinerija – T 001).

Nariai:

prof. dr. Anouar BELAHCEN (Aalto universitetas, Suomija, Elektros ir elektronikos inžinerija – T 001),

dr. Vidas RAUDONIS (Kauno technologijos universitetas, Elektros ir elektronikos inžinerija – T 001),

prof. dr. Voitech STANKEVIČ (Vilniaus Gedimino technikos universitetas, Elektros ir elektronikos inžinerija – T 001),

doc. dr. Eldar ŠABANOVIČ (Vilniaus Gedimino technikos universitetas, Elektros ir elektronikos inžinerija – T 001).

Disertacija bus ginama viešame Elektros ir elektronikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje **2026 m. birželio 5 d. 10 val.** Vilniaus Gedimino technikos universiteto Aula Doctoralis posėdžių salėje.

Adresas: Saulėtekio al. 11, LT-10223 Vilnius, Lietuva. Tel.: (0 5) 274 4956; faksas (0 5) 270 0112; el. paštas doktor@vilniustech.lt

Pranešimai apie numatomą ginti disertaciją išsiųsti 2026 m. gegužės 4 d.

Disertaciją galima peržiūrėti Vilniaus Gedimino technikos universiteto talpykloje (https://etalpykla.vilniustech.lt), Vilniaus Gedimino technikos universiteto bibliotekoje (Saulėtekio al. 14, LT-10223 Vilnius, Lietuva) bei Lietuvos mokslų akademijos Vrublevskių bibliotekoje (Žygimantų g. 1, 01102 Vilnius, Lietuva).

Abstract

The increasing role of wind energy in modern power systems creates a growing need for reliable turbine operation, accurate short-term power forecasting, and computationally efficient data-driven methods. This dissertation addresses two related problems: early fault detection in wind turbines using supervisory control and data acquisition (SCADA) time-series data, and short-term wind farm power forecasting using meteorological forecasts. The dissertation aims to develop and investigate data-driven methods that improve the accuracy, efficiency, and practical applicability of short-term wind power forecasting and early wind turbine fault detection using SCADA and meteorological forecast data.

The first part of the dissertation develops and investigates a virtual-sensor-based method for condition monitoring and early fault detection in wind turbines using SCADA time-series data, including the selection of the most informative features and the evaluation of factors affecting prediction accuracy.

The second part of the dissertation analyzes and optimizes recurrent neural-network structures for the virtual sensor by evaluating feature-sequence formation, training schemes, and alternative activation functions to increase accuracy and reduce the computational cost relevant for practical deployment.

The third part of the dissertation develops and investigates a bidirectional long short-term memory (BiLSTM) based method for short-term wind farm power forecasting using meteorological forecast data, and evaluates the impact of different numerical weather prediction (NWP) sources and the suitability of an objective function with a normalized Nord Pool price multiplier for day-ahead energy production forecasts.

The dissertation contributes to the fields of wind energy and artificial intelligence by proposing and validating data-driven methods for virtual sensing, residual-based early fault detection, recurrent-model optimization, computationally efficient activation-function selection, and economically meaningful short-term wind power forecasting. The research results have been published in three peer-reviewed scientific journals and one conference proceeding, and were presented at seven conferences and seminars.

Reziumė

Didėjantis vėjo energijos vaidmuo šiuolaikinėse elektros energetikos sistemose lemia augantį patikimo vėjo jėgainių veikimo, tikslaus trumpalaikio galios prognozavimo ir skaičiavimo požiūriu efektyvių duomenimis grįstų metodų poreikį. Šioje disertacijoje sprendžiamos dvi tarpusavyje susijusios problemos: ankstyvas gedimų aptikimas vėjo jėgainėse, naudojant valdymo, priežiūros ir duomenų surinkimo (SCADA) laike kintančių rodmenų duomenis, ir trumpalaikis vėjo jėgainių parko generuojamos galios prognozavimas, naudojant meteorologines prognozes. Tyrimo tikslas – sukurti ir ištirti duomenimis grįstus metodus, kurie pagerintų būsenos stebėsenos ir prognozavimo tikslumą, efektyvumą bei praktinį pritaikomumą vėjo energetikos sistemose.

Pirmojoje disertacijos dalyje kuriamas virtualiu jutikliu grįstas metodas, skirtas būsenai stebėti ir ankstyviems gedimams aptikti, kai neįprastas veikimas nustatomas pagal skirtumo tarp išmatuotų ir prognozuotų jutiklio reikšmių nuokrypį. Tyrime nagrinėjama, kaip įvesties duomenų pateikimas, mokymo parametrų parinkimas, rekurentinio modelio struktūra ir aktyvavimo funkcijos veikia virtualaus jutiklio tikslumą ir praktinį pritaikomumą.

Antrojoje disertacijos dalyje analizuojamos ir optimizuojamos virtualiajam jutikliui taikomos rekurentinių neuroninių tinklų struktūros, vertinant įvesčių sekų sudarymą, mokymo parametrų parinkimą ir alternatyvias aktyvavimo funkcijas, siekiant padidinti tikslumą ir sumažinti praktiniam taikymui svarbias skaičiavimo sąnaudas.

Trečiojoje disertacijos dalyje nagrinėjamas dvikrypčiu ilgos trumpalaikės atminties modeliu (BiLSTM) pagrįstas trumpalaikio vėjo jėgainių parko galios prognozavimo metodas, naudojantis skaitinių orų prognozių (NWP) duomenis. Tyrime analizuojama skirtingų meteorologinių prognozių šaltinių įtaka ir vertinamas tikslo funkcijos, papildytos normalizuotu „Nord Pool“ kainos daugikliu, tinkamumas paros į priekį energijos gamybos prognozėms.

Disertacija prisideda prie vėjo energetikos ir dirbtinio intelekto sričių, pasiūlydama ir validuodama duomenimis grįstus metodus virtualiam jutikliui sukurti, prognozuojamos ir matuojamos reikšmės skirtumu grįstiems ankstyviems gedimams aptikti, rekurentiniams modeliams optimizuoti, skaičiavimo požiūriu efektyvioms aktyvavimo funkcijoms parinkti ir trumpalaikei vėjo generuojamai galiai prognozuoti, vertinant ne tik pagal statistinę paklaidą, bet ir pagal rinkos rezultatą. Tyrimo rezultatai paskelbti trijuose recenzuojamuose mokslo žurnaluose ir viename konferencijos straipsnių rinkinyje, taip pat pristatyti septyniose konferencijose ir seminaruose.

Notations

Symbols

a_t – binary anomaly indicator (liet. *dvejetainis anomalijos indikatorius*);

C – imbalance cost functional (liet. *disbalanso sąnaudų funkcionalas*);

C_{1cell} – total cost of the single-cell recurrent reference model (liet. *vieno langelio rekurentinio etaloninio modelio bendrosios sąnaudos*);

$C_{\mathrm{aff}}^{(\ell)}, C_{\mathrm{nonlin}}^{(\ell)}$ – affine and nonlinear cost terms of recurrent layer ℓ (liet. *rekurentinio sluoksnio ℓ afininės ir netiesinės sąnaudų dedamosios*);

$C_{\mathrm{aff,cell}}, C_{\mathrm{nonlin,cell}}$ – affine and nonlinear cost terms of a single recurrent cell (liet. *vieno rekurentinio langelio afininės ir netiesinės sąnaudų dedamosios*);

C_g, C_s – scalar weighted costs of the gate and state activation functions (liet. *vartų ir būsenos aktyvavimo funkcijų svertinės skaliarinės sąnaudos*);

$C_g^{\mathrm{acc}}, C_s^{\mathrm{acc}}$ – accelerated scalar costs of the gate and state activation functions (liet. *pagreitintos vartų ir būsenos aktyvavimo funkcijų skaliarinės sąnaudos*);

$C_{\mathrm{LSTM}}, C_{\mathrm{BiLSTM}}, C_{\mathrm{2BiLSTM}}$ – preliminary cost functions of the LSTM, BiLSTM, and two-layer BiLSTM models (liet. *LSTM, BiLSTM ir dviejų sluoksnių BiLSTM modelių preliminariosios sąnaudų funkcijos*);

$\overrightarrow{c}_t, \overleftarrow{c}_t$ – forward and backward cell states in BiLSTM (liet. *dvikrypčio LSTM tiesioginė ir atgalinė langelio būsenos*);

D, D_ℓ – input dimension in the reference case and in layer ℓ (liet. *įvesties dimensija etaloniniu atveju ir sluoksnyje ℓ*);

d – dimension of the input vector (liet. *įvesties vektoriaus dimensija*);

$\Delta f_j(t)$ – first difference of feature j at sample t (liet. *požymio j pirmasis skirtumas imtyje t*);

$f_\theta(\cdot)$ – virtual sensing or predictive maintenance model (liet. *virtualaus jutiklio arba gedimų prognozės modelis*);

$f_j(t), \bar{f}_j$ – value and sample mean of feature j (liet. *požymio j reikšmė ir jo imties vidurkis*);

$g_\phi(\cdot)$ – wind power forecasting model (liet. *vėjo galios prognozavimo modelis*);

g_t, c_t – cell candidate and cell state of the LSTM cell (liet. *LSTM langelio kandidatas ir langelio būsena*);

h – forecast horizon (liet. *prognozės horizontas*);

H_ℓ – number of hidden units in layer ℓ (liet. *paslėptųjų vienetų skaičius sluoksnyje ℓ*);

$\overrightarrow{h}_t, \overleftarrow{h}_t$ – forward and backward hidden states in BiLSTM (liet. *dvikrypčio LSTM tiesioginė ir atgalinė paslėptosios būsenos*);

h_t^{bi} – BiLSTM output at time t (liet. *BiLSTM išvestis laiko momentu t*);

$\tilde{h}_t, h_t$ – candidate hidden state and hidden state in the GRU model (liet. *kandidatinė paslėptoji būsena ir paslėptoji būsena GRU modelyje*);

i_t, f_t, o_t – input, forget, and output gates of the LSTM cell (liet. *LSTM langelio įvesties, užmiršimo ir išvesties vartai*);

j, M – candidate-feature index and number of candidate features (liet. *kandidatinio požymio indeksas ir kandidatinių požymių skaičius*);

k – sensitivity coefficient used in thresholding (liet. *jautrumo koeficientas slenksčiui apskaičiuoti*);

L – sequence length or input window length (liet. *sekos arba įvesties lango ilgis*);

L_j – Laplacian score of feature j (liet. *požymio j Laplaso įvertis*);

ℓ – layer index in the recurrent architecture (liet. *sluoksnio indeksas rekurentinėje architektūroje*);

$\mathrm{Mon}(f_j)$ – monotonicity criterion of feature j (liet. *požymio j monotoniškumo kriterijus*);

μ, σ – mean and standard deviation used in the fixed Gaussian-type threshold (liet. *vidurkis ir standartinis nuokrypis, naudojami nustatant fiksuotą Gauso tipo slenkstį*);

$\mu_{r,t}$ – rolling mean of residuals (liet. *slenkantis nuokrypių vidurkis*);

μ_W, σ_W – mean and standard deviation in window W (liet. *vidurkis ir standartinis nuokrypis lange W*);

N – number of samples in a window or evaluation set (liet. *imčių skaičius lange arba vertinimo aibėje*);

N_+, N_- – numbers of positive and negative first differences (liet. *teigiamų ir neigiamų pirmųjų skirtumų skaičiai*);

$\odot$ – Hadamard product used in element-wise multiplications (liet. *Hadamardo sandauga, taikoma elementinėse daugybose*);

$P_i, \hat{P}_i$ – measured and predicted power values in the evaluation set (liet. *išmatuotos ir prognozuotos galios reikšmės vertinimo aibėje*);

P_t – measured generated power at time t (liet. *išmatuota generuojama galia laiko momentu t*);

$\bar{P}$ – mean measured power (liet. *vidutinė išmatuota galia*);

$\hat{P}_{t+h}$ – forecasted power at horizon h (liet. *prognozuojama galia horizontui h*);

ϕ – trainable parameters of model g (liet. *apmokomi modelio g parametrai*);

π_t – normalized market-price weight derived from the Nord Pool price of the forecasted hour (liet. *normalizuotas rinkos kainos svoris, sudarytas iš prognozuojamos valandos „Nord Pool“ kainos*);

$\psi(\cdot), \phi(\cdot)$ – gate and state activation functions (liet. *vartų ir būsenos aktyvavimo funkcijos*);

$\mathbb{R}$ – set of real numbers (liet. *realiųjų skaičių aibė*);

R^2 – coefficient of determination (liet. *determinacijos koeficientas*);

r_t – residual between measured and predicted values; in the GRU equations, reset gate (liet. *nuokrypis tarp išmatuotos ir prognozuotos reikšmės; GRU lygtyse - nustatymo iš naujo vartai*);

s – SCADA signal sample (liet. *SCADA signalo imtis*);

S_{pq} – similarity weight between samples p and q (liet. *panašumo svoris tarp imčių p ir q*);

$\sigma_{r,t}$ – rolling standard deviation of residuals (liet. *slenkantis nuokrypių standartinis nuokrypis*);

$s_W^{\min}, s_W^{\max}$ – minimum and maximum signal values in window W (liet. *minimali ir maksimali signalo reikšmė lange W*);

t – time index (liet. *laiko indeksas*);

T – sequence length in the computational-cost analysis (liet. *sekos ilgis, taikomas skaičiavimo sąnaudų analizėje*);

τ – fixed anomaly threshold (liet. *pastovus anomalijos slenkstis*);

τ_t – adaptive anomaly threshold (liet. *adaptyvus anomalijos slenkstis*);

θ – trainable parameters of model f (liet. *apmokomi modelio f parametrai*);

$\mathrm{Var}(f_j)$ – variance criterion of feature j (liet. *požymio j dispersijos kriterijus*);

$\mathrm{Var}_D(f_j)$ – degree-weighted variance of feature j (liet. *pagal laipsnį svertinė požymio j dispersija*);

W – time window of samples (liet. *laiko langas iš imčių*);

$W_\bullet, R_\bullet, b_\bullet$ – input weights, recurrent weights, and bias terms of the recurrent layer (liet. *rekurentinio sluoksnio įvesties svoriai, rekurentiniai svoriai ir poslinkio nariai*);

$w_{t:t+h}$ – meteorological forecast feature sequence (liet. *meteorologinės prognozės požymių seka*);

x – scalar pre-activation value (liet. *skaliarinė priešaktyvacinė reikšmė*);

x_t – SCADA input vector at time t (liet. *SCADA įvesties vektorius laiko momentu t*);

y_t – monitored target variable (liet. *stebimasis tikslinis kintamasis*);

$\hat{y}_t$ – model estimate of the target variable (liet. *modelio įvertinta tikslinio kintamojo reikšmė*);

z_t – update gate in the GRU equations; in the forecasting model, historical measurement vector (liet. *GRU lygtyse - atnaujinimo vartai; prognozavimo modelyje - istorinių matavimų vektorius*);

Abbreviations

Acc. – Accuracy (liet. *tikslumas*);

API – Application Programming Interface (liet. *taikomųjų programų sąsaja*);

BiLSTM – Bidirectional Long Short-Term Memory (liet. *dvikryptis ilgos trumpalaikės atminties modelis*);

CNN – Convolutional Neural Network (liet. *konvoliucinis neuroninis tinklas*);

CPU – Central Processing Unit (liet. *centrinis procesorius*);

DT – Decision Tree (liet. *sprendimų medis*);

ECMWF – European Centre for Medium-Range Weather Forecasts (liet. *Europos vidutinės trukmės orų prognozių centras*);

EDP – Energias de Portugal (liet. *Energias de Portugal*);

FLOPs – Floating-Point Operations (liet. *slankiojo kablelio operacijos*);

FN – False Negative (liet. *klaidingai neigiamas rezultatas*);

FP – False Positive (liet. *klaidingai teigiamas rezultatas*);

FP32 – 32-bit Floating-Point Format (liet. *32 bitų slankiojo kablelio formatas*);

GBT – Gradient Boosted Tree (liet. *gradientinio stiprinimo medis*);

GEM – Global Environmental Multiscale (liet. *globalus aplinkos daugiamastis modelis*);

GFS – Global Forecast System (liet. *globalioji prognozavimo sistema*);

GRU – Gated Recurrent Unit (liet. *sklendžių rekurentinių vienetų modelis*);

GWEC – Global Wind Energy Council (liet. *Pasaulinė vėjo energetikos taryba*);

ICON – Icosahedral Nonhydrostatic (liet. *ikosaedrinis nehidrostatinis modelis*);

IFS – Integrated Forecasting System (liet. *integruotoji prognozavimo sistema*);

LR – Linear Regression (liet. *tiesinė regresija*);

LSTM – Long Short-Term Memory (liet. *ilgos trumpalaikės atminties modelis*);

MAE – Mean Absolute Error (liet. *vidutinė absoliutinė paklaida*);

MAPE – Mean Absolute Percentage Error (liet. *vidutinė absoliutinė procentinė paklaida*);

MEPS – MetCoOp Ensemble Prediction System (liet. *„MetCoOp“ ansamblinė prognozavimo sistema*);

MLP – Multilayer Perceptron (liet. *daugiasluoksnis perceptronas*);

MSE – Mean Squared Error (liet. *vidutinė kvadratinė paklaida*);

NMAE – Normalized Mean Absolute Error (liet. *normuota vidutinė absoliutinė paklaida*);

NWP – Numerical Weather Prediction (liet. *skaitmeninis orų prognozavimas*);

Prec. – Precision (liet. *preciziškumas*);

RF – Random Forest (liet. *atsitiktinis miškas*);

RMSE – Root Mean Square Error (liet. *vidutinės kvadratinės paklaidos šaknis*);

RNN – Recurrent Neural Network (liet. *rekurentinis neuroninis tinklas*);

SCADA – Supervisory Control and Data Acquisition (liet. *dispečerinio valdymo ir duomenų surinkimo sistema*);

Sens. – Sensitivity (liet. *jautris*);

Spec. – Specificity (liet. *specifiškumas*);
STD – Standard Deviation (liet. *standartinis nuokrypis*);
SVM – Support Vector Machine (liet. *atraminių vektorių mašina*);
TN – True Negative (liet. *teisingai neigiamas rezultatas*);
TP – True Positive (liet. *teisingai teigiamas rezultatas*);
UTC – Coordinated Universal Time (liet. *koordinuotasis pasaulinis laikas*);

Contents

Introduction

Problem Formulation

This dissertation addresses how to improve the accuracy, robustness, computational efficiency, and practical usefulness of data-driven methods for two wind-energy applications: early fault detection in wind turbines and short-term wind farm power forecasting. Modern wind energy systems generate large volumes of Supervisory Control and Data Acquisition (SCADA) and meteorological time-series data, but limited fault labels, wide operating variability, incomplete observability of individual turbines, and uneven numerical weather prediction input quality reduce the effectiveness of conventional threshold-based monitoring, manual input selection, and standard error-minimization objectives. The core problem investigated in this dissertation is therefore how to construct and optimize sequence-based recurrent models that learn informative temporal representations from SCADA and weather data, support virtual sensing and residual-based anomaly detection, identify suitable meteorological inputs for day-ahead forecasting, and reduce computational burden while improving practical economic relevance.

Relevance of the Dissertation

The rapid expansion of wind energy increases the need for reliable turbine operation and accurate short-term power forecasting, because both directly affect maintenance costs, grid stability, and participation in electricity markets. As wind farms generate large volumes of SCADA and meteorological time-series data, data-driven methods have become a natural choice for early fault detection and power forecasting.

In Lithuania, related research has mainly addressed local wind resource assessment and forecasting (Gecevičius et al. 2019; Jankevičienė 2024; Katinas et al. 2017). This dissertation extends that context by combining SCADA-based early fault detection and short-term wind farm power forecasting, while also considering recurrent-model efficiency, embedded deployment constraints relevant to electronics engineering, and the economic impact of forecast errors. Therefore, improved feature extraction, model optimization, and economically meaningful forecasting strategies remain relevant for modern wind energy systems.

Object of Research

The object of this doctoral research is recurrent neural-network-based methods for early fault detection and short-term power forecasting.

Aim of the Dissertation

The aim of the dissertation is to develop and investigate data-driven methods that improve the accuracy, efficiency, and practical applicability of short-term wind power forecasting and early wind turbine fault detection using SCADA and meteorological forecast data.

Tasks of the Dissertation

The following research tasks were defined to achieve the main aim of the dissertation:

1. To develop and investigate a virtual-sensor-based method for condition monitoring and early fault detection in wind turbines using SCADA time-series data, including the selection of the most informative features and the evaluation of factors affecting prediction accuracy.

2. To analyze and optimize recurrent neural-network structures for the virtual sensor by evaluating feature-sequence formation, training schemes, and alternative activation functions to increase accuracy and reduce the computational cost relevant for practical deployment.
3. To develop and investigate a BiLSTM-based method for short-term wind farm power forecasting using meteorological forecast data, and to evaluate the impact of different NWP sources and the suitability of an objective function with a normalized Nord Pool price multiplier for day-ahead energy production forecasts.

Research Methodology

This dissertation applies data-driven time-series analysis, recurrent neural networks, and residual-based anomaly detection methods to wind energy applications. The methodology includes SCADA and meteorological data preprocessing, feature selection, temporal aggregation, recurrent modeling with LSTM, GRU, and BiLSTM architectures, and comparative experimental evaluation of virtual-sensor-based early fault detection and short-term wind power forecasting. Models were trained and evaluated using statistical metrics such as RMSE, MAE, R^2, and residual standard deviation, together with economically relevant criteria where needed, while hyperparameter tuning and model comparison were performed experimentally, including the analysis of numerical weather prediction sources and alternative activation functions.

Scientific Novelty of the Dissertation

The scientific novelty of the dissertation is defined by the following main contributions:

- A virtual-sensor-based method is developed for early wind turbine fault detection from deviations between measured and predicted values, enabling monitoring of dynamically varying signals with rare failures.
- An adaptively determined threshold for these deviations enables prediction of sensor-monitored component failure 20–30 days in advance.
- Bounded non-exponential activation alternatives are proposed for recurrent neural networks, reducing computational burden without losing prediction accuracy.

- For a Lithuanian wind farm, an aggregated weather-forecast source yields larger short-term forecasting error when used with a BiLSTM-based recurrent model.
- An alternative objective function incorporating the Nord Pool electricity price is proposed for recurrent wind-energy forecasting models, reducing financial losses caused by forecast-production mismatch.

Practical Value of the Research Findings

The virtual-sensor-based method supports earlier detection of abnormal SCADA signal changes and timely maintenance planning, while the developed guidelines for feature selection, temporal aggregation, and recurrent-model selection support more accurate and computationally efficient models.

The short-term forecasting results improve the use of meteorological and operational data for 24-hour forecasts and show that models should be evaluated not only by statistical accuracy but also by the financial losses caused by forecast errors.

The dissertation results were used in the project “Industrial Internet methods for electrical energy conversion systems monitoring and diagnostics”, contract No. S-BMT-21-5 (LT08-2-LMT-K-01-040).

Defended Statements

The following scientific statements are proposed for defense:

1. Using an adaptive threshold based on a moving median to evaluate the deviation between the measured value and the virtual sensor prediction makes it possible to extend the warning interval before a fault by about 50%.
2. The accuracy of recurrent time-series models depends strongly on input representation and training configuration, and appropriate choices of temporal aggregation, sequence length, and training history can reduce prediction error by more than 20% compared with other investigated configurations.
3. The computational efficiency of recurrent models can be improved by replacing standard exponential-based activation functions with properly selected bounded alternatives, reducing nonlinear computation cost by more than 30% in small models and by more than 60% in the activation component of larger models without degrading prediction accuracy.

4. In short-term wind farm power forecasting, selecting the ICON meteorological forecast source instead of GFS Global or GEM Global makes it possible to reduce forecasting error by more than 12%, while applying an objective function with a normalized Nord Pool price multiplier makes it possible to improve the economic outcome by more than 18%.

Approval of the Research Findings

The research results were published in three scientific articles in Clarivate Analytics Web of Science journals with an impact factor and one conference proceeding. The author gave seven presentations at conferences and seminars:

- Faculty of Electronics Scientific Seminar. 2025. Vilnius, Lithuania.
- Modelling, Data Analytics and AI in Engineering. 2024. Porto, Portugal.
- UiA to the Renewable Energy Group Seminar. 2024. Grimstad, Norway.
- Data Analysis Methods for Software Systems. 2024. Druskininkai, Lithuania.
- The 10th Jubilee IEEE Workshop on Advances in Information, Electronic, and Electrical Engineering. 2023. Vilnius, Lithuania.
- Data Analysis Methods for Software Systems. 2022. Druskininkai, Lithuania.
- Electronics and Electrical Engineering. 25th Lithuanian Conference of Young Scientists. 2022. Vilnius, Lithuania.

Structure of the Dissertation

The dissertation consists of an introduction, three main chapters, general conclusions, and a list of references.

The dissertation comprises 114 pages, including 12 figures, 67 equations, and 37 tables. A total of 102 bibliographic sources are cited in the text.

Acknowledgements

I thank Prof. Dr Artūras Serackis for his supervision, the University of Agder, and Prof. Dr Joao Leal for his collaboration, and my family for their support throughout this dissertation.

1

Data-Driven Maintenance and Forecasting for Wind Energy Systems

This chapter presents a review of the literature on two data-driven problems in wind energy systems. First, the literature on predictive maintenance in wind turbines is reviewed, with an emphasis on observability, feature preparation, virtual sensing, warning logic, and deployment constraints. Then, the literature on short-term wind power forecasting is reviewed, with emphasis on the forecast horizon, meteorological and NWP inputs, the model and training strategy, and an economically meaningful evaluation. The review presented in this chapter is published in scientific publications related to the dissertation (Jankauskas et al., 2023a; Jankauskas et al., 2024; Jankauskas et al., 2026).

1.1. Predictive Maintenance

This section reviews the predictive maintenance literature through four recurring decisions: what information can be observed in routinely available measurements, how SCADA time series are turned into informative features, which decision logic yields operationally meaningful warnings, and whether the final model can be deployed under realistic computational constraints (Campoverde-Vilela et al. 2023; de Azevedo et al. 2016; Gonzalez et al. 2016; Liu et al. 2021; Roelofs et al. 2021; Tidriri et al. 2021; Turnbull et al. 2021). In data-driven maintenance workflows, SCADA measurements are first prepared and transformed into features and then passed to a decision model that produces a health indicator, a fault probability,

or an early warning (Eriksson 2020; Garan et al. 2022; Tautz-Weinert, Watson 2017a). In the reviewed maintenance literature, four decisions recur throughout: observability, feature preparation, warning logic, and deployment cost.

1.1.1. Observability and Failure Constraints in Supervisory Control and Data Acquisition-Based Maintenance

The first question is whether the available measurements are informative enough to support reliable early warning. Modern wind farms tend to be equipped with SCADA systems that monitor operating parameters such as wind speed, rotor speed, generator power, temperatures, pressures, and alarms. Although SCADA was initially created for monitoring and control, it now serves as a useful data source to monitor large-scale conditions and predict failures (Gonzalez et al. 2016; Liu et al. 2021; Tautz-Weinert, Watson 2017a; Turnbull et al. 2021).

Condition monitoring reviews emphasize that no sensing approach is universally optimal. Instead, the preferred technique depends on the criticality of the components, the available instrumentation, and the warning horizon required for maintenance planning (Becker, Poste 2006; de Azevedo et al. 2016; Hameed et al. 2009; Stetco et al. 2019; Tautz-Weinert, Watson 2017b). Vibration analysis, in particular, remains a powerful diagnostic tool for mechanical problems such as gear tooth damage, but its sampling rates and infrastructure requirements differ significantly from SCADA-based systems (Iorgulescu, Beloiu 2008; Serrato et al. 2007). This is one of the main reasons why SCADA-based maintenance remains appealing for fleet-level deployment.

Figure 1.1 provides a simplified overview of the internal structure of a wind turbine and anchors the discussion to the main subsystems that recur throughout the predictive maintenance literature. In published SCADA studies, the gearbox, generator, bearings, transformer, pitch system, and hydraulic system are often treated as critical because they contribute substantially to downtime and repair costs (Liu et al. 2021; Tidriri et al. 2021; Turnbull et al. 2021).

These studies also show that SCADA-based maintenance observes failure processes only indirectly. Maintenance modeling is therefore often linked to related analytical tasks that provide operational context, including power curve monitoring for underperformance detection and sensor consistency checks (Kusiak et al. 2009), as well as broader wind energy analysis tasks such as wind speed prediction and power optimization (El-Fouly et al. 2008; Lin, Liu 2020a). Research on healthy state behavior over time and feature selection strategies further indicates that the usefulness of SCADA-based maintenance depends strongly on how data are summarized and compared over time, rather than on fault labels alone (Jastrzebska et al. 2021; Marti-Puig et al. 2019). These connections matter because the same data quality problems can bias both maintenance and forecasting models.

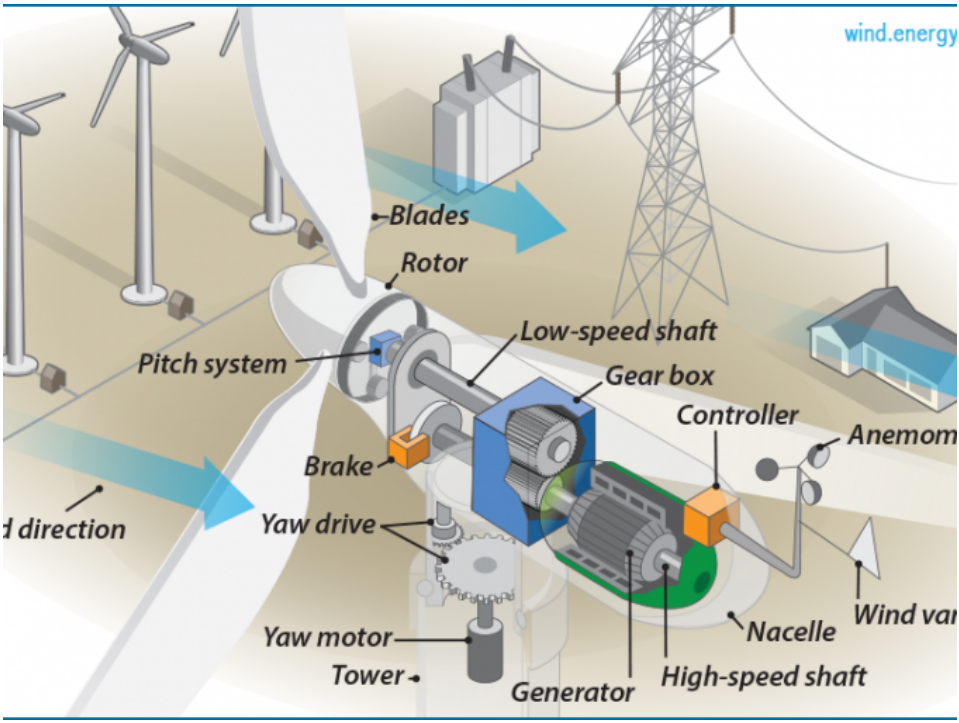


Fig. 1.1. Inside of a wind turbine (source: U.S. Department of Energy, energy.gov (2022); public domain according to the Office of Energy Efficiency and Renewable Energy copyright policy)

From an experimental design perspective, predictive maintenance based on SCADA data is constrained by a limited and imbalanced failure history, uncertainty in failure timestamps caused by late reporting or maintenance delays, and nonstationarity caused by sensor drift, firmware updates, or seasonal effects (Eriksson 2020; Garan et al. 2022; Tidriri et al. 2021). These constraints determine which feature extraction strategies remain feasible and which validation protocols can be defended (Berscheid 2021; Garan et al. 2022; Ng 2021). The literature, therefore, frames SCADA-based maintenance as a partially observable decision problem with limited data rather than as a standard supervised classification task.

1.1.2. Feature Preparation, Virtual Sensing, and Data Quality

The next question is how raw SCADA sequences should be prepared so that early signs of degradation are not masked by noise, operating regime changes, or label-

ing artifacts. The reviewed literature groups maintenance features into four main families:

- *Manually engineered statistical features* computed over time windows (mean, variance, quantiles, slopes, and correlations) (Eriksson 2020; Marti-Puig et al. 2019; Tidriri et al. 2021);
- *Virtual sensing and residual features*, where a model predicts a target (e.g., temperature), and the prediction deviation is used as an anomaly indicator (Azzam et al. 2021; Qian et al. 2019; Yan et al. 2021);
- *Multivariate health indicators* constructed from multiple SCADA channels (e.g., distance-based indices, reconstruction error) (Jastrzebska et al. 2021; Miele et al. 2022; Wen et al. 2022; Zheng et al. 2023);
- *Learned feature extraction* using autoencoders and CNN-based encoders to reduce the reliance on domain-specific feature design (Jalayer et al. 2021; Khan et al. 2023; Miele et al. 2022).

The literature does not treat preprocessing as a neutral step. Figure 1.2, recreated from the same EDP challenge data used by Eriksson (2020), is kept here because it directly shows why preprocessing becomes a design variable: different scaling methods alter the distribution of the same ambient temperature series and therefore change the learning conditions for downstream models. Standard scaling and min-max scaling preserve the overall temporal shape while changing magnitudes, whereas power transforms, and can substantially reshape the distribution, which may improve variance stabilization but can also distort physically meaningful relationships depending on the variable (Eriksson 2020; Garan et al. 2022).

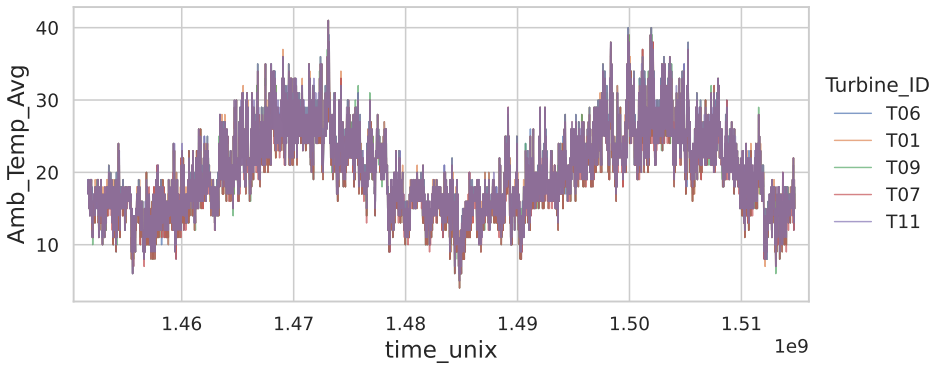


Fig. 1.2. Ambient temperature over time (example variable used in model training; source: recreated from the EDP Wind Turbine Failure Detection Challenge SCADA data used by Eriksson (2020))

These observations lead to two practical requirements that recur throughout the maintenance literature: preprocessing must preserve physically meaningful relationships such as thermal response to load, and the resulting set of features must remain robust to outliers, missing values, and changes in operating regimes (Eriksson 2020; Garan et al. 2022; Marti-Puig et al. 2019). In practice, preprocessing is therefore paired with explicit handling of anomalies and data artifacts because SCADA streams contain both genuine signs of degradation and communication dropouts or sensor glitches (Khan et al. 2023; Roelofs et al. 2021; Zhang et al. 2022; Zheng et al. 2023).

Recent studies have improved maintenance models in two main ways. Some focus on model architecture and hyperparameter tuning, while others focus on improving data by refining data quality, target definitions, and feature robustness (Berscheid 2021; Garan et al. 2022; Ng 2021). This distinction matters here because later chapters treat temporal aggregation, training data composition, and target construction as experimental factors rather than fixed preprocessing details.

Garan et al. (2022) proposed a data-focused methodology for modeling wind turbine maintenance in which performance was improved through iterative refinement of the data set rather than repeated redesign of the model itself. Figure 1.3 illustrates one of the main reasons why data-centric treatment is necessary: failure events can be recorded late or with ambiguous start times, directly affecting the label construction and validation logic (Garan et al. 2022; Tidriri et al. 2021).

Figure 1.4 shows the workflow proposed by Garan et al. (2022), in which preprocessing, feature selection, and model evaluation are refined iteratively rather than fixed in advance. This is important here because it treats data preparation and target construction as part of the modeling process, rather than as a separate preliminary step.

Figure 1.5 places this workflow in a broader data-centric machine learning perspective, in which data cleaning, label quality, and target definition are treated as primary design decisions (Berscheid 2021; Garan et al. 2022; Ng 2021). Unlike the more application-specific workflow in Figure 1.4, this broader view makes explicit that improvements can come from revisiting the data and labels themselves rather than only from redesigning the model.

Together, these figures show why target quality and data consistency should be treated as major contributors to maintenance performance rather than as secondary preprocessing details (Marti-Puig et al. 2019). Feature preparation, temporal aggregation, and target construction are therefore treated here as explicit experimental factors. In practice, these choices affect both what the model can learn from the data and how reliable the subsequent validation results are.

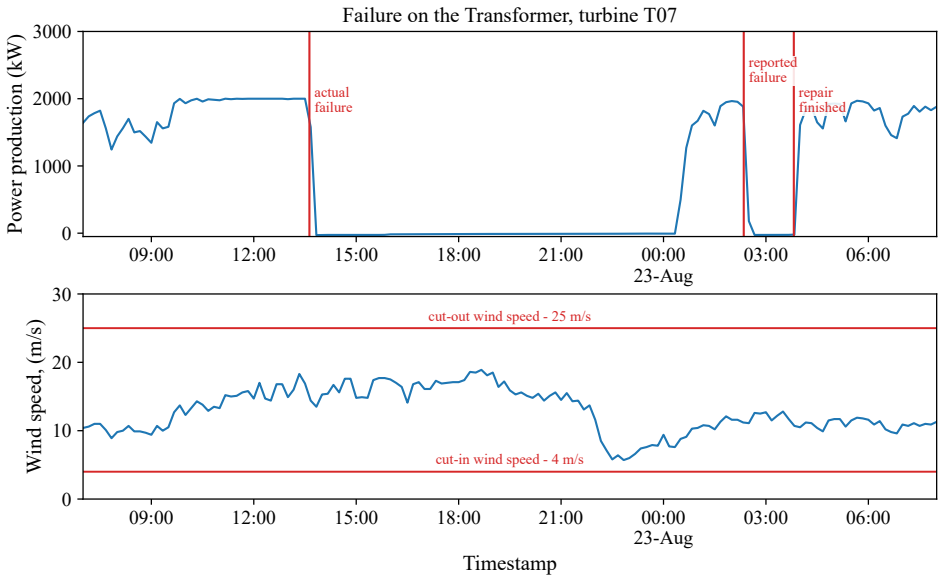


Fig. 1.3. Example of late reporting of a failure event (source: recreated from the EDP Wind Turbine Failure Detection Challenge SCADA signals, failure annotations, and event logs used by Garan et al. (2022))

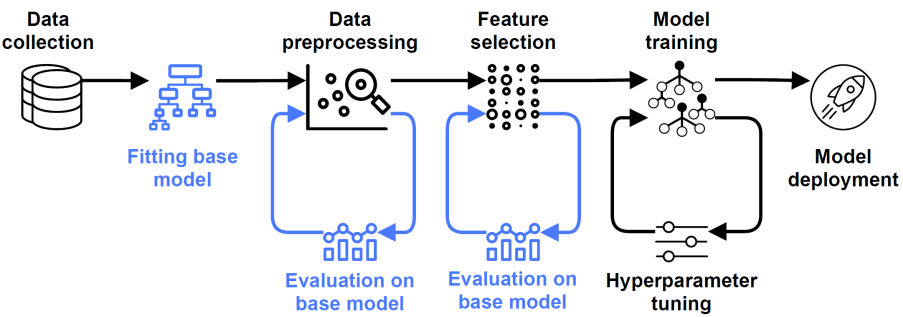


Fig. 1.4. Pipeline of the proposed data-centric ML methodology (source: Garan et al. (2022))

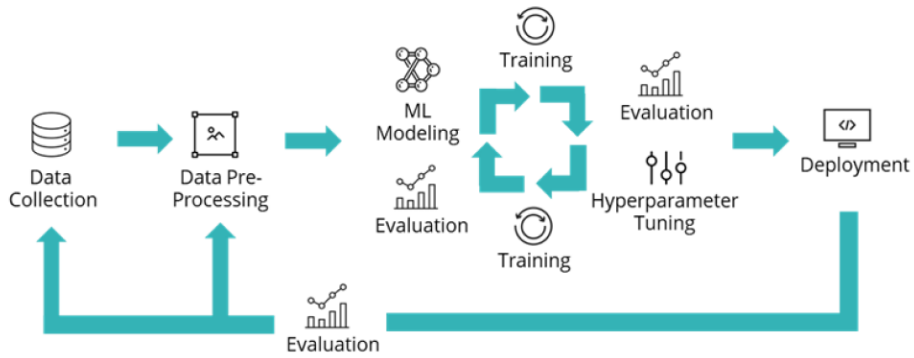


Fig. 1.5. Example of a data-centric ML project (source: Berscheid (2021))

1.1.3. Decision Models, Early-Warning Logic, and Maintenance Trade-offs

After the features and targets are defined, predictive maintenance still requires a decision logic that turns the model output into a fault probability, component condition, or early warning (Campoverde-Vilela et al. 2023; Tidriri et al. 2021). The reviewed literature shows that choosing this decision model is not simply a matter of selecting the highest reported score. Instead, the decision layer must be interpreted together with the class imbalance, the cost of false alarm, and the practical meaning of the warning delay (Campoverde-Vilela et al. 2023; Eriksson 2020; Liu et al. 2020; Tidriri et al. 2021).

Many studies benchmark classical machine learning models due to their interpretability and lower computational cost (Eriksson 2020; Tidriri et al. 2021), while deep sequence models such as LSTM and BiLSTM are increasingly being used for the diagnosis of gearbox and bearings under changing operating conditions (Bharatheedasan et al. 2023; Cao et al. 2019; Jalayer et al. 2021; Zheng et al. 2023). For example, Eriksson (2020) compared the following algorithms for the prediction of failure based on SCADA data:

- Multilayer Perceptron (MLP),
- Linear Regression (LR),
- Random Forest (RF),
- Decision Tree (DT),
- Support Vector Machine (SVM),
- Gradient Boosted Tree (GBT).

Table 1.1. Classification results for different stacked models (Eriksson 2020)

Models	TP	FP	TN	FN	Prec.	Sens.	Spec.	Acc.	Time
					%	%	%	%	s
MLP/MLP	8291	20119	130073	8950	29.18	48.09	86.60	82.64	7272
MLP/RF	4866	14969	135223	12375	24.53	28.22	90.03	83.67	6954
MLP/DT	5057	16670	133520	12184	23.28	29.33	88.90	82.77	5470
MLP/LR	6985	19386	130806	10256	26.49	40.51	87.09	82.30	9456
RF/RF	4991	14919	135273	12250	25.07	28.95	90.07	83.77	4358
RF/DT	4799	14375	135817	12442	25.03	27.83	90.43	83.98	3240
RF/LR	4579	14275	135917	12662	24.29	26.56	90.50	83.91	12081
DT/DT	5598	15413	143779	11643	26.64	32.47	90.32	84.67	860
DT/LR	5838	19352	130840	11403	23.18	33.86	87.12	81.63	16426

The same study also evaluated stacking strategies in which multiple base learners are combined. Table 1.1 shows that the choice of the model changes the entire error structure, not only a single summary metric: some combinations improve robustness by increasing the number of correctly detected cases and reducing false negatives, but the results remain strongly affected by the class imbalance and validation design (Eriksson 2020).

The main point of Table 1.1 is that predictive maintenance models cannot be judged only by headline accuracy: a configuration with similar aggregate performance may still generate an unacceptable false positive burden or miss too many rare failures. The table shows this directly. DT/DT reported the highest accuracy (84.67%) and the shortest execution time (860 s), but MLP/MLP achieved much higher sensitivity (48.09% versus 32.47%) at the cost of 20,119 false positives. This is why several studies complement classification metrics with maintenance cost analysis (Eriksson 2020; Tidriri et al. 2021).

In addition to statistical metrics, Eriksson (2020) also reported estimated economic results for each component. Table 1.2 reveals that excessive maintenance, prompted by high false positive rates, can actually result in financial losses. This finding supports existing research on maintenance decisions that consider costs (Eriksson 2020; Tidriri et al. 2021). In this table, positive values signify net savings, while negative values indicate a net economic detriment stemming from overly frequent interventions or undetected failures. A window filtering strategy (Table 1.3) reduces false positives but can increase false negatives, illustrating the operational trade-off between early warning aggressiveness and maintenance cost (Eriksson 2020). The totals make this trade-off explicit: filtering reduces false positives from 112 to 20, but false negatives increase from 2 to 9, and the overall net savings deteriorate from -295,128 EUR to -348,614 EUR. These results show that maintenance models must be interpreted jointly in statistical and economic terms because a technically plausible classifier may still be impractical if it triggers too many unnecessary maintenance actions.

Table 1.2. Net savings and classification metrics by component using unfiltered predictions (Eriksson 2020)

Component	Net savings (EUR)	TP	FP	FN
Gearbox	-4157	2	23	0
Generator	-170000	0	22	1
Generator Bearing	-82778	2	23	0
Transformer	-59535	1	20	1
Hydraulic Group	21342	6	24	0
Sum	-295128	11	112	2

Table 1.3. Filtered net savings and classification metrics by component (Eriksson 2020)

Component	Net savings (EUR)	TP	FP	FN
Gearbox	-64712	1	6	1
Generator	-85000	0	5	1
Generator Bearing	-10516	2	6	0
Transformer	-103000	0	2	2
Hydraulic Group	-85386	1	1	5
Sum	-348614	4	20	9

Table 1.4. Comparison of the studies by Tidriri et al. (2021) and Eriksson (2020) (Accuracy %; Cost kEUR)

Failure	Accuracy	Accuracy	Cost	Cost
Gearbox	89.01	82.64	+115	-64.712
Gen.	98.71	80.79	+10.75	-85
Gen. bearing	94.31	81.11	-35	-10.516
Transf.	93.43	88.21	+22.5	-103
Hydr.	79.95	70.65	-132	-85.386
Avg	91.08	80.68	-18.75	-348.614

Reliability can also be improved by training separate models for specific failure types. In this approach, a separate model is trained for each component or failure mode (Tidriri et al. 2021). Tidriri et al. (2021) followed this strategy by training a dedicated model for each type of failure, as illustrated in Figure 1.6. The figure makes an important methodological point: improvement is often achieved not only by changing the algorithm, but by restructuring the maintenance task around component-specific failure mechanisms.

Table 1.4 illustrates the same point numerically. Although the underlying studies are not fully comparable in data, labeling, and validation protocol, the comparison remains useful as a qualitative overview because it shows that the reported maintenance performance can change substantially when the prediction strategy is aligned more closely with the type of failure and intervention logic. Even with that comparability limit, one pattern remains clear: the strategy tailored to specific failure types reported higher accuracy for all listed failure types, and its costs at the component level are positive for gearbox, generator, and transformer, while the

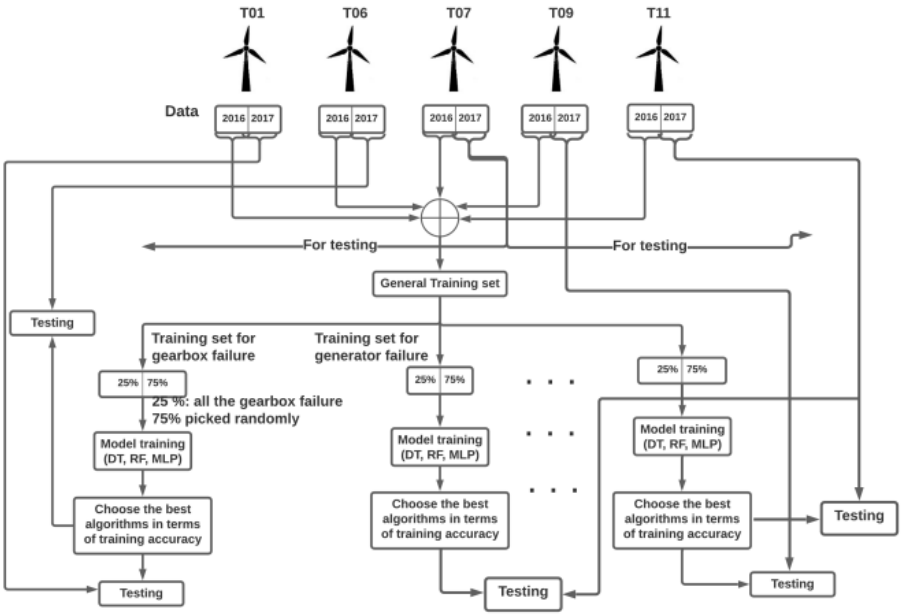


Fig. 1.6. Predictive models tailored to specific failure types in wind turbine components (source: Tidriri et al. (2021))

corresponding values in Eriksson (2020) remain negative. The model structure and evaluation protocol, therefore, have to be treated as experimental factors, because they change what kind of maintenance decision is actually being tested.

Anomaly detection is frequently employed when labeled failure data is scarce, extending beyond the realm of supervised fault classification (Campoverde-Vilela et al. 2023; Khan et al. 2023; Liu et al. 2022; Roelofs et al. 2021; Wen et al. 2022; Zhang et al. 2022; Zheng et al. 2023). In certain semi-supervised scenarios, models are trained exclusively on healthy data, which mitigates the necessity of historical failures and facilitates alerts well in advance of breakdowns, while maintaining low false positive rates (Campoverde-Vilela et al. 2023). Furthermore, unsupervised deep learning frameworks that integrate autoencoders and graph neural networks have been suggested to discern dependencies among SCADA channels, thus increasing the lead time for early warnings (Miele et al. 2022). Together, these studies show that useful early warning can be formulated even when labeled failures are sparse, and the decision model must operate close to the boundary between anomaly detection and prognostics.

The threshold design then becomes part of the decision logic rather than a separate processing detail. Comparative research indicates that the implementation

of adaptive thresholds, as opposed to static limits, can improve the warning time in regression-based monitoring systems (Liu et al. 2020). Likewise, sequence models have been employed to address particular anomaly classes, including blade icing, by leveraging the combined context of SCADA and environmental data (Wang et al. 2022b). A further practical challenge is generalization under nonstationarity: even models that perform well in controlled evaluation may degrade when sensor drift, turbine retrofits, or firmware updates shift data distributions, leading to false alarms or missed degradation (Garan et al. 2022; Gruhl et al. 2021).

Accordingly, the remaining maintenance analysis focuses on early warning and assesses the warning stage through prediction quality together with threshold behavior, warning stability, and robustness under distribution shift.

1.1.4. Virtual Sensing as the Retained Maintenance Strategy

Within the maintenance literature, virtual sensing is especially relevant because it addresses the main observability gap of SCADA-based monitoring. Instead of attempting to classify rare failures directly from heterogeneous operating signals, a virtual sensor predicts an internal variable, typically a temperature or load, from other SCADA channels and then monitors the deviation between the measured and predicted value over time (Azzam et al. 2021; Qian et al. 2019; Yan et al. 2021).

This strategy appears in several complementary studies. Ensemble methods have been used to predict wind turbine bearing temperatures, prediction errors from Long Short-Term Memory (LSTM) models have been applied for condition monitoring, and virtual gearbox load sensors have been developed using multibody simulation and neural networks (Azzam et al. 2021; Qian et al. 2019; Yan et al. 2021). The common contribution of these studies is not only the use of regression, but also the conversion of a difficult fault detection problem into a deviation monitoring problem that remains compatible with routinely available SCADA data.

Under these constraints, virtual sensing becomes a coherent maintenance strategy because it links feature preparation, threshold design, and early-warning evaluation within the same monitoring framework.

1.1.5. Lightweight Inference and Embedded Deployment

Practical predictive maintenance often requires processing close to real time and deployment close to the asset, where computational resources and energy budgets are limited (Rybalkin et al. 2020; Silfa et al. 2020). For practical deployment, maintenance models must be evaluated not only by predictive performance, but also by inference cost, memory footprint, and energy consumption (Chong et al. 2021; Timmons, Rice 2020). This deployment viewpoint already matters at the literature-review stage because later chapters study simplified recurrent mod-

els and alternative activation functions under these resource constraints, in line with broader work on compact network design and efficient mobile deployment (Howard et al. 2019).

In edge deployments, these constraints are often associated with microcontroller devices and local inference pipelines, motivating both software accelerators and compact network designs (Banbury et al. 2021; Howard et al. 2019; Lane, Bhattacharya 2016).

For recurrent neural networks, a large portion of the computational cost is associated with gate nonlinearities (logistic sigmoid activation and hyperbolic tangent activation) and sequential processing (Chong et al. 2021; Rybalkin et al. 2020). Work on efficient recurrent inference therefore examines hardware accelerators and parallelism, numerical precision reduction, and approximation of activation functions (Chong et al. 2021; Rybalkin et al. 2020; Silfa et al. 2020; Timmons, Rice 2020). For example, optimized LSTM/BiLSTM accelerator architectures have been proposed to increase parallelism while still relying on standard logistic sigmoid and hyperbolic tangent activations (Rybalkin et al. 2020). Dynamic precision selection can reduce running time and energy without harming accuracy (Silfa et al. 2020). Efficient approximations of logistic sigmoid and hyperbolic tangent activations using piecewise polynomials or lookup tables reduce power consumption in dedicated hardware (Chong et al. 2021), and evaluations on CPUs also indicate that fast approximations can reduce training and inference time (Timmons, Rice 2020).

Another area of research investigates different activation functions or their configurations, with the goal of reducing computational costs while maintaining accuracy (Ali et al. 2022; Dubey et al. 2022; Elsayed et al. 2019; Gomes et al. 2011; Seidel et al. 2021; Wuraola, Patel 2022). Seidel et al. (2021) discussed how alternative activations can affect both accuracy and processing speed in neural networks. Ali et al. (2022) evaluated LSTM classifiers with multiple activation functions and compared their effects on datasets and training setups. Additional research efforts propose new activation units or alternative nonlinearities aimed at hardware efficiency (Joseph, Bindiya 2023; Parisi et al. 2021).

In embedded deployment contexts, the choice of activation function is increasingly discussed alongside lightweight architectures and other efficiency-oriented design measures. For example, efficient architectural modifications and activation choices have been explored in energy-efficient DL models, and surveys of activation functions highlight the trade-off between smoothness, accuracy, and computational cost (Chandra 2020; Ding et al. 2023; Howard et al. 2019; Vallés-Pérez et al. 2023; Wuraola, Patel 2022; Zhang et al. 2025). In this deployment literature, computational efficiency is treated as a design constraint rather than as a post hoc implementation issue. Recurrent cell structure and activation design, therefore,

have to be considered together with predictive performance when deployment near the asset is required.

Across the predictive maintenance studies reviewed in this section, five factors recur: SCADA-based observability under uncertain failure timing, feature preparation and target construction, data-centric treatment of labels and preprocessing, residual-based early warning logic with explicit threshold design, and computational efficiency for deployment. They are carried forward into the theoretical analysis of the Second Chapter and the experimental evaluation of the Third Chapter.

1.2. Wind Power Generation Forecasting

In this section, the forecasting literature is narrowed to the studies most relevant for later methodological choices. The review focuses on four questions: which forecast horizon is operationally meaningful, how meteorological and NWP inputs should be selected, which model and training choices remain justified under input uncertainty, and why forecast quality must be interpreted not only statistically but also economically (Choudhary et al. 2023; González-Sopeña et al. 2021b; Lagos et al. 2022a,b; Tsai et al. 2023; Ye et al. 2022). Wind energy is increasingly important in the transition to sustainable electricity generation. According to the Global Wind Energy Council (GWEC) market statistics, the cumulative installed global wind capacity reached approximately 906 GW by the end of 2022, and annual new installations exceeded 100 GW around 2023 (Global Wind Energy Council (GWEC) 2023). GWEC further estimates that annual new wind installations will increase from approximately 162 GW in 2026 to approximately 207 GW by 2030, resulting in approximately 1,060 GW of additional capacity deployed globally over 2025–2030, with cumulative capacity reaching approximately 2,196 GW by the end of 2030. However, GWEC also notes that achieving a 1.5°C aligned trajectory and the 2030 "tripling" ambition would require annual wind installations to accelerate to at least 320 GW (i.e., $\geq$2.7 TW cumulative capacity by 2030) (Global Wind Energy Council (GWEC) Market Intelligence 2025). Reliable wind power operation depends strongly on the quality of meteorological forecasts. Under variable wind conditions, forecast quality directly affects day-to-day plant scheduling, grid balancing, and overall operating costs (Georgilakis 2008; González-Sopeña et al. 2021b; Lagos et al. 2022a). Accurate forecasting of wind power is essential for grid integration, plant operation, and participation in deregulated electricity markets (González-Sopeña et al. 2021b; Lagos et al. 2022a). Because wind energy is unpredictable and behaves in a nonlinear way, forecasting errors can lead to imbalances and economically unfavorable dispatch decisions, particularly in the context of energy trading (González-Sopeña et al. 2021b; Wang et al. 2021). Re-

cent reviews also show that the forecasting field has expanded rapidly in terms of deep learning models, uncertainty treatment, and evaluation strategies (Choudhary et al. 2023; Lagos et al. 2022b; Tsai et al. 2023; Ye et al. 2022). Both application domains rely on SCADA and meteorological/NWP data that change over time, and in both cases, performance depends on data preparation and on whether the selected models can be executed efficiently (Gonzalez et al. 2016; Lagos et al. 2022a; Turnbull et al. 2021; Wang et al. 2021).

1.2.1. Forecast Setting, Observability, and Meteorological Inputs

The forecasting problem must first be defined in terms of horizon and observability. Forecasting models are commonly grouped by horizon: very short-term (seconds to 4 h), short-term (4–24 h), medium-term (1–7 days), and long-term (weeks to years) (Lagos et al. 2022a; Mundotiya et al. 2022a; Wang et al. 2021; Ye et al. 2022). Here, the key point is that the 6 h and 24 h horizons are not arbitrary reporting choices: they align with market trading and operational scheduling cycles, so the usefulness of the forecast cannot be separated from the selected horizon (González-Sopeña et al. 2021b; Lagos et al. 2022a; Tsai et al. 2023). For horizons beyond several hours, numerical weather prediction (NWP) output and meteorological forecasts are typically necessary (Aggarwal, Kumar 2013; Bouallègue et al. 2023; Heinermann, Kramer 2016; Lagos et al. 2022a).

Input features for wind power forecasting commonly include wind speed, wind direction, temperature, humidity, and pressure, often supplemented with lagged power measurements and engineered temporal descriptors (Chen et al. 2021; Demolli et al. 2019; Heinermann, Kramer 2016; Wang et al. 2021, 2023). Several enhancement strategies are reported in the literature.

- *Weather clustering*, where models are trained using subsets of historical days with conditions similar to the target day (Dimd et al. 2022);
- *Outlier detection and filtering*, such as Isolation Forest applied prior to training deep models (Kisvari et al. 2021);
- *Automated feature extraction* using deep encoders such as autoencoders, restricted Boltzmann machines, and CNN-based encoders, which can learn representations directly from raw sequences but may require a higher training cost (Higashiyama et al. 2018; Wang et al. 2021; Ying et al. 2023).

Forecasting accuracy depends not only on model architecture, but also on the quality, structure, and uncertainty of the meteorological input (Kisvari et al. 2021; Lagos et al. 2022a; Liu et al. 2023a, 2024; Wang et al. 2021). Operational factors such as power outages, operating constraints, and maintenance actions can distort the relation between inputs and outputs in a wind farm. Periods affected by out-

ages, operating constraints, or maintenance actions should therefore be identified and either removed or handled separately in the training data, depending on the forecasting objective, to obtain more robust forecasting models (Mundotiya et al. 2022b). This literature treats meteorological input preparation and NWP selection as design variables rather than as fixed external conditions, especially when multiple NWP products or correction strategies are available (Aggarwal, Kumar 2013; Bouallègue et al. 2023; Liu et al. 2024). Accordingly, NWP sources and input preparation choices are compared explicitly rather than treated as fixed background conditions.

1.2.2. Model and Training Choices under Input Uncertainty

Once the forecast horizon and input space are fixed, the next question is how to choose the prediction model and the training strategy. A wide range of machine learning models have been applied to forecast wind power generation. Classical approaches, such as regression and tree-based models, remain relevant because they offer interpretability and relatively low computational overhead (Demolli et al. 2019; Heinermann, Kramer 2016; Wang et al. 2023). Support Vector Regression is frequently reported as a strong baseline and has demonstrated robust generalization between sites in multilocation settings (Demolli et al. 2019; Liu et al. 2018). Ensemble models and hybrid optimization-based strategies are also commonly used, particularly when feature engineering is well designed (Demolli et al. 2019; Heinermann, Kramer 2016; Pan et al. 2019; Rahaman et al. 2020; Wang et al. 2020).

The main recent improvements in short-term forecasting are associated with deep learning models that capture temporal dependencies, including RNN, LSTM, GRU, BiLSTM, and hybrid CNN–RNN architectures (Chai 2023; Chen et al. 2021; Higashiyama et al. 2018; Kisvari et al. 2021; Lin, Liu 2020b; Liu et al. 2023b; Wang et al. 2022a, 2021; Yin et al. 2019; Zhang et al. 2020). However, the literature does not support the simplistic conclusion that deeper architectures are automatically superior. For example, Kisvari et al. (2021) showed that the application of Isolation Forest to filter input data improved deep learning performance, and GRU/LSTM achieved high precision only after the input data had been cleaned. Other architectures tackle sequence modeling in different ways; for example, decomposition-based BiLSTM models and attention-based BiLSTM models have both been proposed for short-term wind power forecasting (Chai 2023; Liu et al. 2023b).

Another strand of the literature treats input uncertainty explicitly rather than assuming that meteorological forecasts are equally reliable. Hybrid approaches that account for meteorological input uncertainty have therefore been proposed (Liu et al. 2023a; Mundotiya et al. 2022a; Sideratos, Hatziargyriou 2007). For example, Sideratos, Hatziargyriou (2007) used self-organizing maps to categorize predicted wind speed and estimate forecast quality, improving subsequent power

predictions by incorporating input reliability. NWP correction and uncertainty-aware forecasting frameworks further support this viewpoint (Liu et al. 2023a, 2024). Two implications matter here. First, the NWP source cannot be treated as a neutral background choice when different products and correction strategies are available. Second, once forecast quality is interpreted in operational terms, the training objective also becomes part of the modeling decision. Within the broader model, families surveyed here, recurrent sequence models remain a justified family for the dissertation's short-term forecasting experiments without implying universal superiority.

1.2.3. Economic Interpretation and Practical Forecasting Constraints

The main practical question is what should count as a useful forecast. Despite strong results in published studies, forecasting systems used in practice remain sensitive to data quality and preprocessing choices, redundancy and collinearity of input variables, site-specific effects that reduce generalization, and the computational cost and limited interpretability of complex deep models (Demolli et al. 2019; González-Sopeña et al. 2021a; Piotrowski et al. 2022; Wang et al. 2021). This means that forecast quality cannot be defended only by reporting a low error on a single data split.

In large-scale centralized forecasting systems, data privacy is an additional constraint because pooling operational data from multiple wind farms can expose sensitive information (Li et al. 2023; Yang et al. 2023). At the same time, from an operational standpoint, the quality of the forecast must also be assessed in economic terms. In deregulated electricity markets, forecast errors translate into imbalance costs, and the same magnitude of error can have a different financial impact depending on market conditions and penalty structures (González-Sopeña et al. 2021b). Reviews of wind power forecast metrics also show that the choice of assessment strongly shapes the interpretation of model quality (González-Sopeña et al. 2021a; Piotrowski et al. 2022). For horizons used in market participation, the literature increasingly links prediction errors to economically meaningful outcomes rather than to purely statistical scores alone (González-Sopeña et al. 2021a,b). Training objectives, therefore, cannot be chosen only as technical optimization devices; they also determine how forecast errors are valued in practice.

In the reviewed forecast studies, four factors recur: operational horizon definition, meteorological input quality and NWP source selection, model and training strategy under input uncertainty, and economic interpretation of forecast error (Choudhary et al. 2023; González-Sopeña et al. 2021b; Lagos et al. 2022b; Ng 2021; Tsai et al. 2023; Wang et al. 2021). Based on that, the forecasting analy-

sis focuses on short-term prediction with BiLSTM models, explicit comparison of NWP sources, and an economically oriented training objective.

1.3. Conclusions of the First Chapter and Formulation of the Dissertation Tasks

1. The reviewed literature shows that wind energy systems generate large-scale SCADA and meteorological time-series data suitable for data-driven predictive maintenance and wind power forecasting; however, both application domains are constrained by incomplete observability, uncertain failure timing, and changing operating conditions.
2. In predictive maintenance, feature preparation, preprocessing, and target construction are key design decisions: the literature shows that they directly determine which signs of degradation remain visible to the model and which validation results are defensible.
3. For SCADA-based maintenance, virtual sensing combined with deviation monitoring and threshold design provides a practical way to build early-warning models when failure history is limited, because it supports healthy-state modeling and warning generation within the same framework.
4. For practical deployment, computational efficiency is a primary design factor. This makes recurrent model simplification, activation function selection, and other measures that balance predictive performance with execution cost relevant research directions.
5. For short-term wind power forecasting, the literature points to four recurring factors: operational horizon definition, meteorological input quality and NWP source selection, model and training choices when input quality varies, and the economic interpretation of forecast errors. For this reason, forecast quality has to be treated not only as a statistical regression problem, but also as a decision problem shaped by market conditions.

Based on this literature review and the identified research gap, the following hypotheses are formulated for empirical verification in this dissertation:

1. An approach based on a virtual sensor and selected SCADA time series features can provide accurate modeling of normal sensor behavior and enable early fault detection in wind turbines through residual analysis.
2. The prediction accuracy and practical suitability of the virtual sensor can be improved by optimizing the structure of the recurrent neural network, the representation of the temporal input, the training configuration, and

the activation functions, while reducing the computational complexity for deployment.

3. The accuracy and practical usefulness of BiLSTM-based short-term wind farm power forecasting depend on the selection of meteorological forecast inputs and the training objective, and an economically oriented loss function can improve day-ahead forecasting performance under market conditions.

The following dissertation tasks are defined to verify these hypotheses:

1. To develop and investigate a virtual-sensor-based method for condition monitoring and early fault detection in wind turbines using SCADA time-series data, including the selection of the most informative features and the evaluation of factors affecting prediction accuracy.
2. To analyze and optimize recurrent neural-network structures for the virtual sensor by evaluating feature-sequence formation, training schemes, and alternative activation functions to increase accuracy and reduce the computational cost relevant for practical deployment.
3. To develop and investigate a BiLSTM-based method for short-term wind farm power forecasting using meteorological forecast data, and to evaluate the impact of different NWP sources and the suitability of an objective function with a normalized Nord Pool price multiplier for day-ahead energy production forecasts.

2

Theoretical Foundations of Virtual Sensing and Wind Power Forecasting

This chapter establishes the theoretical foundation of the dissertation and introduces the principal concepts used in the subsequent experimental investigation. It considers two applied problems. The first is data-driven condition monitoring and early fault detection based on a virtual sensor constructed from SCADA data. The second is short-term wind farm power generation forecasting under limited observability of individual turbines. The chapter further presents the recurrent modeling framework, the main factors influencing prediction accuracy, and the theoretical considerations related to the reduction of computational cost for embedded deployment. Part of the material presented in this chapter has been published in two scientific publications (Jankauskas et al., 2023a; Jankauskas et al., 2026).

During the preparation of this dissertation, OpenAI Codex was used for supporting technical functions, including LaTeX formatting, language editing support, and consistency checks of tables, captions, references, and layout. The tool was not used as a primary source of scientific information; all generated suggestions were critically reviewed, verified, and edited by the author (OpenAI 2026).

2.1. Data-Driven Condition Monitoring and Early Fault Detection Using Supervisory Control and Data Acquisition Data

This section defines the practical problem of data-driven condition monitoring and early fault detection addressed in the dissertation and explains the motivation for the proposed virtual sensing approach. The considered task is not formulated as remaining useful life estimation, because the objective is not to predict the exact amount of time left before component failure. Instead, the aim is to detect the onset of abnormal behavior at an early stage by comparing the measured value of a physical sensor with the output of its virtual counterpart trained to represent normal operation. Under this logic, the monitored component is assumed to operate normally as long as the residual between the real and predicted sensor values remains within the expected range, whereas a systematic increase in this residual is treated as an indication of anomalous behavior that may precede failure.

For experimental validation of the proposed data-driven condition monitoring and early fault detection approach, this dissertation uses the public EDP wind turbine dataset from the Wind Turbine Failure Detection Challenge (Portugal 2019). The dataset contains two years of SCADA records collected from five wind turbines with a temporal resolution of 10 minutes, together with a list of registered component failures. Each SCADA record includes the turbine identifier, the measurement timestamp, and a broad set of operational, thermal, electrical, and environmental variables. The available measurements include generator speed statistics, generator bearing temperatures, generator phase temperatures, slip ring chamber temperature, hydraulic oil temperature, gearbox oil temperature, gearbox bearing temperature, nacelle temperature and direction, rotor speed statistics, blade pitch angle statistics, ambient and estimated wind speed, relative and absolute wind direction, ambient temperature, active and reactive power by generator state, transformer temperatures, inverter temperatures, controller temperatures, spinner temperature, grid power variables, possible power variables, power factor, grid frequency, grid voltage, grid current, and busbar temperature. In this dissertation, the target signal is the single SCADA channel listed in the dataset as *gearbox bearing temperature*. The fault annotations in the dataset cover gearbox, generator, generator bearing, transformer, and hydraulic group failures. Table 2.1 summarizes the recorded failures across turbines and components.

Table 2.1 shows that failures are sparse and unevenly distributed across both turbines and component types. This characteristic has direct methodological consequences. Although the dataset is rich in terms of operating measurements, the number of registered failures is limited, and the fault events are not uniformly represented across the monitored fleet. This makes straightforward supervised fault

Table 2.1. Failure list based on Portugal (2019)

Component	T01		T06		T07		T09		T11	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Gearbox	1	0	0	1	0	0	1	1	0	0
Generator	0	0	5	0	0	1	0	0	1	0
Generator Bearing	0	0	0	0	1	1	3	1	0	0
Transformer	0	1	0	0	2	0	0	0	0	0
Hydraulic Group	0	0	1	1	0	2	0	1	1	2

classification less reliable and motivates the use of normal-behavior modeling instead, where the objective is to learn the expected sensor response under healthy operating conditions and then detect abnormal deviations from that learned behavior.

In this dissertation, the virtual sensor is constructed from a selected subset of SCADA variables that describe the operating state of the turbine and the environmental excitation acting on it. The actual model inputs are generator RPM represented by maximum, minimum, mean, and standard deviation values; rotor RPM represented by maximum, minimum, and mean values; ambient wind speed represented by maximum, minimum, mean, and standard deviation values; wind direction represented by relative and absolute values; and ambient temperature represented by its mean value. These variables were selected because they provide a compact but physically meaningful description of the loading and environmental conditions that influence the thermal response of the monitored component.

The need for a virtual-sensor-based formulation is also supported by the behavior of the target thermal signal itself. The SCADA channel listed in the dataset as *gearbox bearing temperature* varies over a wide operating range, even under healthy conditions, which means that early fault development cannot be identified reliably from the absolute sensor value alone. In the healthy state, the median temperature is 52.61 °C, the interquartile range extends from 44.17 to 59.89 °C, and the full observed range spans from 20.00 to 71.50 °C. By comparison, the pre-failure temperature increase is relatively small: the median reaches 52.83 °C at 30–60 days before failure and 53.50 °C at 0–30 days before failure, which corresponds to median shifts of only 0.22 °C and 0.89 °C, respectively. These values show that the natural variability of the sensor signal is substantially larger than the early pre-failure change. Impending failure, therefore, cannot be assessed robustly using a fixed temperature threshold or direct inspection of temperature trajectories alone. A more suitable approach is to evaluate the deviation of the measured signal from its expected normal behavior under the given operating conditions.

The general idea of the proposed monitoring approach is therefore to model the expected sensor behavior conditioned on the current operating context and then analyze the deviation of the real measurement from this context-dependent esti-

mate. If the virtual sensor reproduces healthy behavior with sufficient accuracy, then an increasing residual can be interpreted as evidence that the physical relationship between the monitored sensor and the remaining SCADA variables is no longer consistent with normal operation. In this sense, the virtual sensor serves as a reference model of healthy behavior, while the residual serves as the anomaly indicator. This formulation provides the theoretical basis for the recurrent modeling approach developed later in the dissertation, where sequence-based neural networks are used to capture the temporal dependencies between SCADA inputs and the target sensor signal.

2.2. Virtual Sensor Concept and Training Principle

This section presents the theoretical basis of the virtual sensor used in the dissertation. In the proposed framework, the virtual sensor is defined as a data-driven model that estimates the value of a physical sensor from other SCADA variables available in the wind turbine monitoring system. Its main purpose is to reproduce the expected behavior of the real sensor during healthy turbine operation as accurately as possible. This requirement is important because the diagnostic value of the residual depends directly on the prediction error achieved under normal conditions. If the virtual sensor remains sufficiently accurate across different operating regimes, then a systematic deviation between the measured and predicted sensor values can later be interpreted as an indication of abnormal behavior rather than as a modeling artifact.

Let $x_t \in \mathbb{R}^d$ denote the SCADA input vector at time t (e.g., wind speed, rotor speed, generator speed, power, and ambient temperature), and let $y_t \in \mathbb{R}$ denote the monitored target variable (e.g., the SCADA channel listed as *gearbox bearing temperature*). Using a sequence length L, the forecasting model $f_\theta(\cdot)$ estimates:

$$\hat{y}_t = f_\theta(x_{t-L+1}, \ldots, x_t)\,. \tag{2.1}$$

The residual signal is defined as:

$$r_t = y_t - \hat{y}_t. \tag{2.2}$$

An anomaly decision can be defined using a fixed threshold:

$$a_t = \begin{cases} 1, & |r_t| > \tau, \\ 0, & \text{otherwise}, \end{cases} \tag{2.3}$$

or a threshold that varies over time, estimated from a rolling residual distribution:

$$\tau_t = \mu_{r,t} + k\sigma_{r,t}, \tag{2.4}$$

where $\mu_{r,t}$ and $\sigma_{r,t}$ are the rolling mean and standard deviation of the residuals, and k controls the sensitivity.

In this expression, $\hat{y}_t$ denotes the model prediction of the monitored target, r_t is the residual between the measured and predicted values, and $a_t \in \{0, 1\}$ is the binary anomaly indicator. The thresholds τ and τ_t have the same physical unit as the monitored variable y_t (for example, °C when the target is bearing temperature), whereas k is a dimensionless sensitivity coefficient.

2.2.1. Feature Extraction from Supervisory Control and Data Acquisition Time Series

Feature extraction transforms raw SCADA signals into representations suitable for learning. One common way to do this is to compute summary statistics over predefined time windows. For a SCADA signal s sampled within a window $W = \{s_1, \ldots, s_N\}$, a range of statistical characteristics can be extracted:

$$\begin{aligned} \mu_W &= \frac{1}{N}\sum_{i=1}^{N} s_i, \\ \sigma_W &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i - \mu_W)^2}, \\ s_W^{\min} &= \min_i s_i, \\ s_W^{\max} &= \max_i s_i. \end{aligned} \tag{2.5}$$

In these expressions, W is a time window containing N samples of the SCADA signal s, while μ_W, σ_W, $s_W^{\min}$, and $s_W^{\max}$ denote the mean, standard deviation, minimum, and maximum over that window. All four statistics inherit the physical unit of the original signal (e.g., wind speed in m/s, temperature in °C, or power in kW/MW). Considering several window lengths, such as intervals spanning several hours, helps the model capture both short-term dynamics and slower operational trends. Unlike manually constructed feature sets, automatic feature extraction reduces the need for hand-crafted design and can improve reliability by systematically generating informative statistics or learning representations directly from sequences.

The virtual sensor is trained using historical SCADA data under the assumption that a subset of the available records corresponds to healthy component operation. Under this assumption, the model learns the normal relationship between the target sensor and the remaining operating variables. This training principle is consistent with the purpose of the method, because the objective is not to classify predefined fault categories directly, but to detect departures from normal behavior by analyzing the residual between the physical and virtual sensor outputs. In this sense, the virtual sensor acts as a reference model of expected healthy-state operation and forms the basis for subsequent residual-based anomaly detection.

2.2.2. Candidate Recurrent Network Structures and Cell Formulation

This dissertation considers four candidate recurrent neural network structures for virtual sensor construction. They are ordered below from the simplest to the most complex architecture:

- a single-layer GRU architecture;
- a single-layer BiLSTM architecture;
- a two-layer stacked LSTM–LSTM architecture;
- a two-layer stacked LSTM–BiLSTM architecture.

All candidate structures are used for sequence-based regression, where the target is to estimate the value of the monitored physical sensor from a multivariate SCADA input sequence. The structures differ in the way temporal dependencies are represented and in the resulting model complexity. Since the implementation used in this dissertation is based on MATLAB recurrent layers, the mathematical notation below follows the MATLAB convention, where $W_\bullet$ denotes input weights, $R_\bullet$ denotes recurrent weights, and $b_\bullet$ denotes bias terms.

The simplest candidate is the gated recurrent unit (GRU) model. It is included because it offers a compact recurrent formulation with fewer gating operations than LSTM-based models. In MATLAB, the default GRU implementation uses the reset gate after recurrent matrix multiplication. Its dynamics can therefore be written as

$$r_t = \psi(W_r x_t + R_r h_{t-1} + b_r)\,; \tag{2.6}$$

$$z_t = \psi(W_z x_t + R_z h_{t-1} + b_z)\,; \tag{2.7}$$

$$\tilde{h}_t = \phi(W_h x_t + r_t \odot (R_h h_{t-1}) + b_h)\,; \tag{2.8}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t, \tag{2.9}$$

where r_t is the reset gate, z_t is the update gate, $\tilde{h}_t$ is the candidate hidden state, and h_t is the hidden state at time step t. The function $\psi(\cdot)$ denotes the gate activation

function and $\phi(\cdot)$ denotes the state activation function. In the default MATLAB configuration, $\psi(\cdot)$ is the sigmoid function and $\phi(\cdot)$ is the hyperbolic tangent function.

The next candidate is a single-layer bidirectional LSTM (BiLSTM) model. This structure is more expressive because it processes the sequence in both temporal directions and combines information from past and future samples within the analyzed input window. In MATLAB, the `bilstmLayer` is most naturally represented as two standard LSTM recurrences, one in the forward direction and one in the backward direction. The underlying LSTM cell is therefore written as

$$i_t = \psi(W_i x_t + R_i h_{t-1} + b_i)\,; \tag{2.10}$$

$$f_t = \psi(W_f x_t + R_f h_{t-1} + b_f)\,; \tag{2.11}$$

$$g_t = \phi(W_g x_t + R_g h_{t-1} + b_g)\,; \tag{2.12}$$

$$o_t = \psi(W_o x_t + R_o h_{t-1} + b_o)\,; \tag{2.13}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t; \tag{2.14}$$

$$h_t = o_t \odot \phi(c_t), \tag{2.15}$$

where i_t, f_t, and o_t are the input, forget, and output gates, g_t is the cell candidate, c_t is the cell state, and h_t is the hidden state. In the default MATLAB configuration, the gate activation is sigmoid, and the state activation is hyperbolic tangent. For the bidirectional layer, the two directional states are

$$\left(\overrightarrow{h}_t, \overrightarrow{c}_t\right) = \mathrm{LSTM}_f\left(x_t, \overrightarrow{h}_{t-1}, \overrightarrow{c}_{t-1}\right); \tag{2.16}$$

$$\left(\overleftarrow{h}_t, \overleftarrow{c}_t\right) = \mathrm{LSTM}_b\left(x_t, \overleftarrow{h}_{t+1}, \overleftarrow{c}_{t+1}\right); \tag{2.17}$$

$$h_t^{\mathrm{bi}} = \begin{bmatrix} \overrightarrow{h}_t \\ \overleftarrow{h}_t \end{bmatrix}. \tag{2.18}$$

The BiLSTM output at each time step is the concatenation of the forward and backward hidden states.

A more complex candidate is the stacked LSTM–LSTM architecture. In this architecture, the first LSTM layer processes the input sequence, and the second processes the hidden-state sequence produced by the first. The first layer is defined as

$$\left(h_t^{(1)}, c_t^{(1)}\right) = \mathrm{LSTM}^{(1)}\left(x_t, h_{t-1}^{(1)}, c_{t-1}^{(1)}\right), \tag{2.19}$$

and the second layer is defined as

$$\left(h_t^{(2)}, c_t^{(2)}\right) = \mathrm{LSTM}^{(2)}\left(h_t^{(1)}, h_{t-1}^{(2)}, c_{t-1}^{(2)}\right). \tag{2.20}$$

In this case, both recurrent layers follow the same LSTM cell formulation, but the second layer operates on the hidden representation produced by the first rather than on the raw input sequence. This allows the model to learn a more hierarchical temporal representation.

The most complex candidate considered in this stage of the dissertation is the stacked LSTM–BiLSTM architecture. In this model, the first recurrent layer is a unidirectional LSTM, and the second recurrent layer is bidirectional. The first layer is written as

$$\left(h_t^{(1)}, c_t^{(1)}\right) = \mathrm{LSTM}^{(1)}\left(x_t, h_{t-1}^{(1)}, c_{t-1}^{(1)}\right), \tag{2.21}$$

while the second layer is written as

$$\left(\overrightarrow{h}_t^{(2)}, \overrightarrow{c}_t^{(2)}\right) = \mathrm{LSTM}_f^{(2)}\left(h_t^{(1)}, \overrightarrow{h}_{t-1}^{(2)}, \overrightarrow{c}_{t-1}^{(2)}\right); \tag{2.22}$$

$$\left(\overleftarrow{h}_t^{(2)}, \overleftarrow{c}_t^{(2)}\right) = \mathrm{LSTM}_b^{(2)}\left(h_t^{(1)}, \overleftarrow{h}_{t+1}^{(2)}, \overleftarrow{c}_{t+1}^{(2)}\right); \tag{2.23}$$

$$h_t^{(2,\mathrm{bi})} = \begin{bmatrix} \overrightarrow{h}_t^{(2)} \\ \overleftarrow{h}_t^{(2)} \end{bmatrix}. \tag{2.24}$$

This arrangement first transforms the original SCADA sequence into a latent temporal representation and then refines that representation using bidirectional context. For this reason, it is treated as the most expressive and computationally demanding candidate among the initially considered structures.

In all equations above, x_t denotes the input vector at time step t, $\odot$ denotes the Hadamard product, and the states are dimensionless neural-network variables after preprocessing rather than direct physical measurements. The gate activation function is denoted by $\psi(\cdot)$ and the state activation function by $\phi(\cdot)$ so that the same mathematical form can later be reused when alternative activation functions are investigated in the dissertation.

2.3. Factors Affecting Virtual Sensor Prediction Accuracy

This section discusses the main factors that may influence the accuracy of the virtual sensor and, therefore, the reliability of residual-based anomaly detection. The prediction error depends not only on the type of model selected, but also on the quality of the input features, the temporal representation of the data, the composition of the training data, and the chosen hyperparameters. For this reason, several aspects are identified in this dissertation as theoretically relevant and are later examined in the experimental part.

2.3.1. Input Feature Preparation and Selection

The predictive accuracy of the virtual sensor depends on the amount of useful information contained in the input sequence. The SCADA data set includes a wide range of operational, electrical, thermal, and environmental measurements, but not all available variables are equally useful for predicting the target sensor. Some variables describe the turbine operating regime and loading conditions in a physically meaningful way, while others are redundant, only weakly informative, or too closely tied to the target response itself. For this reason, the input space was treated as a design variable, and a separate feature-selection stage was performed before the main comparative modeling experiments.

In the feature-selection stage, 82 SCADA candidate features were ranked using three complementary criteria: monotonicity, Laplacian score, and variance. The purpose of this ranking was to identify variables that are dynamically informative, sufficiently variable, and potentially useful for describing changes in turbine operating behavior. Let $f_j(t)$ denote the value of the candidate characteristic j in sample t, where $j = 1, \ldots, M$ and $M = 82$. For a time series of length N, the variance of feature j is defined as

$$\mathrm{Var}(f_j) = \frac{1}{N-1} \sum_{t=1}^{N} \left(f_j(t) - \bar{f}_j \right)^2, \tag{2.25}$$

where $\bar{f}_j$ is the sample mean of the feature. This criterion reflects the degree of amplitude variation present in the signal and helps exclude nearly constant variables with limited descriptive value.

The monotonicity criterion was used to evaluate whether a feature exhibits a consistent directional tendency over time. In general form, monotonicity can be written as

$$\mathrm{Mon}(f_j) = \left| \frac{N_+ - N_-}{N-1} \right|, \tag{2.26}$$

where N_+ is the number of positive first differences $\Delta f_j(t) = f_j(t) - f_j(t-1)$ and N_- is the number of negative first differences. A larger value indicates a more directionally consistent change pattern. The Laplacian-score-based criterion was used to evaluate the locality-preserving discriminative ability of each feature, and can be expressed as

$$L_j = \frac{\sum_{p,q} \left(f_j(p) - f_j(q)\right)^2 S_{pq}}{\mathrm{Var}_D(f_j)}, \tag{2.27}$$

where S_{pq} is the similarity weight between samples p and q, and $\mathrm{Var}_D(f_j)$ denotes the degree-weighted variance of the feature. Since the ranking in this dissertation follows the implementation used in the design worksheet, the numerical interpretation is tied to that specific implementation.

The ranking results (Table 2.2) show that the highest-ranked variables according to the selected criteria were dominated by power-related and possible-power-related signals. According to the monotonicity-based ranking, the strongest candidates were the minimum possible capacitive reactive power (0.1211), the average gearbox oil temperature (0.1057), the average possible capacitive reactive power (0.1012), the average possible inductive reactive power (0.1005), and the maximum possible inductive reactive power (0.1002). According to the Laplacian-score-based ranking used in the worksheet, the leading features were the average grid active power (0.9956), the minimum possible inductive reactive power (0.9730), the standard deviation of grid reactive power (0.8492), the minimum possible capacitive reactive power (0.7843), and the minimum blade pitch angle (0.7838). According to the variance criterion, the highest-ranked features were the minimum possible inductive reactive power (0.2014), the maximum possible capacitive reactive power (0.1943), the average possible inductive reactive power (0.1218), the average possible capacitive reactive power (0.1171), and the maximum possible active power (0.0920). These results indicate that a considerable part of the SCADA feature space is strongly related to turbine operating load and power-conversion behavior.

However, the final input subset used for virtual sensor modeling was not obtained by simply selecting the numerically highest-ranked variables. Instead, the ranking results were combined with engineering judgment and with the modeling objective of reconstructing the expected target-sensor behavior from compact, physically interpretable, and broadly available operating descriptors. In particular,

Table 2.2. Top-ranked SCADA features according to the three selection criteria

Rank	Monotonicity		Laplacian score		Variance	
	Feature	**Score**	**Feature**	**Score**	**Feature**	**Score**
1	Minimum possible capacitive reactive power	0.1211	Average grid active power	0.9956	Minimum possible inductive reactive power	0.2014
2	Average gearbox oil temperature	0.1057	Minimum possible inductive reactive power	0.9730	Maximum possible capacitive reactive power	0.1943
3	Average possible capacitive reactive power	0.1012	Standard deviation of grid reactive power	0.8492	Average possible inductive reactive power	0.1218
4	Average possible inductive reactive power	0.1005	Minimum possible capacitive reactive power	0.7843	Average possible capacitive reactive power	0.1171
5	Maximum possible inductive reactive power	0.1002	Minimum blade pitch angle	0.7838	Maximum possible active power	0.0920

the final input set was chosen to represent the mechanical operating regime, aerodynamic excitation, inflow direction, and ambient thermal boundary conditions, while avoiding excessive dependence on variables that are too close to the target temperature response itself or that mainly reflect downstream power-conversion effects.

Based on this selection, the final neural-network input consists of 14 features: maximum, minimum, mean, and standard deviation of generator rotational speed; maximum, minimum, and mean of rotor rotational speed; maximum, minimum, mean, and standard deviation of ambient wind speed; relative and absolute wind direction; and mean ambient temperature. The final feature set is smaller than the original ranked feature pool, but it remains physically interpretable and suitable for sequence-based virtual sensing. Generator and rotor speed variables characterize the mechanical operating regime of the turbine, wind-speed variables describe the excitation level acting on the system, wind-direction variables provide contextual information related to aerodynamic alignment, and ambient temperature represents the external thermal condition. This combination was considered theoretically appropriate for estimating the expected target-sensor value under normal operating conditions, while keeping the input representation compact enough for systematic experimental comparison in the later chapters.

2.3.2. Selection of Temporal Aggregation Range

SCADA records in the analyzed dataset are available as 10-minute measurements, which already represent short-term summaries of turbine operation. However, even

after this first level of aggregation, the input signals may still contain short-duration fluctuations caused by transient operating conditions, local disturbances, and rapid changes in turbine loading. Since the target of the virtual sensor is a thermal signal, whose dynamics are typically slower than those of many electrical or mechanical variables, the temporal representation of the input data may have a direct effect on prediction accuracy. For this reason, temporal aggregation is treated in this dissertation not only as a preprocessing step, but also as an explicit design factor.

Additional aggregation over longer time windows may affect virtual sensor performance in two opposite ways. On the one hand, aggregation may reduce short-term variability, suppress high-frequency noise, and better align the input representation with the slower physical dynamics of the monitored temperature signal. In such a case, the recurrent model may learn a more stable mapping between turbine operating conditions and the expected sensor value. On the other hand, excessive aggregation may remove informative temporal variation and compress changes in the operating regime that are still relevant for predicting the target signal. The effect of aggregation, therefore, cannot be assumed in advance and has to be verified experimentally.

The aggregation setup is defined as follows. The basic input representation is formed from 10-minute SCADA records. In addition to this original temporal resolution, several longer aggregation intervals are considered to examine whether a second level of temporal summarization improves the suitability of the input data for virtual sensing. The investigated aggregation windows are selected to represent short, medium, and longer-term summaries of turbine operation. In the planned experimental investigation, the considered aggregation intervals are 3 h, 6 h, and 24 h. These intervals are sufficiently different to reveal whether the virtual sensor benefits more from preserving relatively local operating information or from using a more strongly smoothed operational context.

To avoid drawing conclusions from a single network configuration, the planned aggregation investigation is defined using several reference recurrent neural network structures. The investigated model set includes a single-layer LSTM architecture, a single-layer BiLSTM architecture, a single-layer GRU architecture, and a stacked LSTM–LSTM architecture. The purpose of this design is to evaluate the effect of temporal aggregation across models with different recurrent mechanisms and different temporal representation capacity. In this way, the analysis is intended to separate the influence of the aggregation interval from the influence of the model structure itself.

The experimental investigation planned for this factor is, therefore, organized as a controlled comparison in which the same selected SCADA input features are represented using different temporal aggregation intervals and evaluated with the same set of reference recurrent models (Table 2.3). The main objective of this experiment is to determine whether additional aggregation improves the predic-

Table 2.3. Planned experimental design for the investigation of temporal aggregation

Experimental factor	Tested values	Purpose of the comparison
Base SCADA input resolution	10-minute records	To define the original temporal representation used as the starting point for further aggregation.
Additional aggregation interval	3 h, 6 h, 24 h	To determine whether further temporal summarization improves the suitability of SCADA inputs for thermal virtual sensing.
Reference model structures	GRU, BiLSTM, stacked LSTM–LSTM and LSTM–BiLSTM	To evaluate the aggregation effect across several recurrent architectures rather than under a single model configuration.
Evaluation objective	Prediction error of the virtual sensor	To identify the aggregation interval that provides the most suitable balance between information preservation and suppression of irrelevant short-term fluctuations.
Role in later dissertation stages	Selection of one preferred aggregation interval	To propagate the selected aggregation setting to the later experimental investigation of sequence length, model structure, and training configuration.

tion accuracy of the virtual sensor and, if so, which aggregation interval provides the most suitable balance between information preservation and suppression of irrelevant short-term fluctuations. The comparison is carried out using the same train–test logic and the same prediction error criteria as in the later model comparison experiments, so that the selected aggregation interval can be consistently propagated to the subsequent stages of the dissertation.

In the theoretical framework of the dissertation, temporal aggregation is treated as an experimentally testable hypothesis. The working assumption is that the original 10-minute SCADA records may still contain variability that is not fully beneficial for thermal virtual sensing, but it remains an open question whether moderate or stronger additional aggregation leads to a more informative and robust input representation. The detailed numerical results of this comparison and the final choice of the aggregation interval are presented later in the experimental investigation chapter.

2.3.3. Training Data Composition

The behavior learned by the virtual sensor depends strongly on the composition of the training dataset. Even when the same network structure and the same input features are used, the prediction accuracy may change depending on which turbines are included in the training set, how much historical healthy-state data are used, and how far in time the trained model is applied without retraining. For this reason, the composition of the training data is treated in this dissertation as a separate experimental factor rather than as a fixed implementation detail.

Different training-data compositions may influence virtual sensor performance in different ways. If the model is trained on data from only one turbine, it may learn relationships that are specific to that machine. This can lead to high accuracy when the model is evaluated on the same turbine, but the transfer to other turbines may be weaker if their operating characteristics, sensor calibration, or local behavior differ. Training on healthy data collected from several turbines may help the model learn a broader picture of normal operation. This can improve robustness, although it may also make the model less sensitive to the behavior of an individual turbine. The effect of turbine composition in the training set has to be verified experimentally.

Another important aspect is the duration of historical healthy-state data used for training. A short historical period may not cover a sufficiently broad range of operating conditions, seasonal variations, and loading regimes. As a result, the trained virtual sensor may reproduce only a limited part of normal behavior and may become inaccurate when applied to later data. A longer training period may provide a more complete representation of healthy operation, but it may also include a greater diversity of operating states and increase the complexity of the learned mapping. The required amount of historical healthy-state data should also be treated as an experimentally testable factor.

In addition to the composition of the training set itself, the temporal distance between the training period and the later test period may influence prediction quality. Even if the model is trained on healthy historical data, its accuracy may gradually deteriorate when applied far beyond the training interval, for example, because of seasonal shifts, slow system drift, or changes in turbine operating patterns. For this reason, the present dissertation also defines an experimental comparison in which the virtual sensor is trained on different lengths of historical data and then evaluated on later unseen periods located at different distances from the training interval.

The planned experimental investigation of training data composition is organized into three complementary parts. First, a turbine-specific generalization experiment is planned, in which the virtual sensor is trained on healthy historical data from one turbine and then evaluated on several turbines. Second, a multi-turbine training experiment is planned, in which the virtual sensor is trained on a combined healthy dataset formed from several turbines and then evaluated separately on each turbine using data not included in training. Third, a training-history-length experiment is planned, in which the model is trained using different durations of healthy historical data and later evaluated on temporally separated unseen data.

In the current experimental design (Table 2.4), the training-history-length experiment is planned using 1-month, 3-month, and 6-month healthy historical records for model training. The trained virtual sensor is then planned to be evaluated on unseen data located 3 months and 5 months after the training period. This

Table 2.4. Planned experimental design for the investigation of training data composition

Experimental factor	Tested values	Purpose of the comparison
Training-set turbine composition: single-turbine training	Healthy data from one turbine; testing on several turbines	To determine whether the learned virtual-sensor mapping is turbine-specific or transferable across turbines of the same type.
Training-set turbine composition: multi-turbine training	Healthy data combined from several turbines; testing on each turbine separately using unseen data	To determine whether broader training diversity improves robustness and cross-turbine generalization.
Length of healthy historical data used for training	1 month, 3 months, 6 months	To determine how much historical healthy-state data are needed to build a stable and representative virtual sensor.
Temporal distance between training and testing	Testing on unseen data after 3 months and 5 months	To determine how long a trained virtual sensor remains accurate when applied later in time without retraining.
Evaluation objective	Prediction error of the virtual sensor on unseen data	To compare the influence of training-data composition on generalization, robustness, and temporal durability.
Role in later dissertation stages	Selection of a preferred training-data composition strategy	To define how the final virtual-sensor models should be trained in the later experimental investigation.

design makes it possible to examine not only whether longer training histories improve immediate prediction accuracy, but also whether they improve the temporal durability of the virtual sensor when the model is applied later in time.

In the theoretical framework of the dissertation, training data composition is treated as a structured experimental problem involving turbine diversity, historical coverage, and temporal transferability. The working assumption is that virtual sensor performance depends not only on the network architecture, but also on how representative the healthy training set is with respect to turbine-to-turbine variability and later operating conditions. The detailed numerical results of these comparisons and the final conclusions regarding the preferred training-data composition are presented later in the experimental investigation chapter.

2.3.4. Selection of Input Sequence Length and Number of Training Epochs

The predictive capacity of recurrent neural networks depends not only on the selected input variables and model structure, but also on how much temporal context is provided to the model and how thoroughly the model is optimized during training. In the virtual sensing problem considered in this dissertation, the target signal is a temperature-related variable whose behavior is influenced not only by the current turbine operating state, but also by its recent history. For this reason, the sequence length and the number of training epochs are treated as separate experimental factors in the experimental design.

Sequence length determines how many consecutive aggregated input vectors are made available to the recurrent model before predicting the target sensor value. A short input sequence provides only local temporal context and may be insufficient for representing slower thermal dynamics, delayed responses, or cumulative effects of previous operating regimes. A longer sequence can provide a broader view of recent turbine behavior and may allow the recurrent model to learn more informative temporal dependencies. However, increasing the sequence length also increases the dimensionality of the effective input representation, the memory requirements of the model, and the computational burden of training. In addition, excessively long sequences may include redundant information and may make optimization more difficult without providing a proportional gain in prediction accuracy. Sequence length should, therefore, be treated as a controllable trade-off between temporal coverage and model efficiency.

In the present dissertation, the sequence-length problem is defined after selecting the preferred temporal aggregation setting in the earlier stage of the dissertation. Once the aggregation interval is fixed, each element of the input sequence corresponds to one aggregated vector of selected SCADA features. The planned experimental investigation then evaluates several alternative sequence lengths to determine how much historical context is needed for accurate virtual sensing. In the current experimental design, the investigated sequence lengths are 1, 3, 6, 12, and 24 time steps. This set covers both minimal contextual input and substantially extended temporal context, making it possible to evaluate whether the prediction task benefits more from local or longer-horizon historical information.

The influence of the number of training epochs is considered together with the sequence-length investigation because both factors affect the final performance of the recurrent model, but in different ways. While the sequence length determines the amount of temporal information available at the input, the number of epochs determines how far the model parameters are adapted to the training data. Too few training epochs may lead to underfitting, meaning that the network does not learn the relevant relationship between the aggregated SCADA inputs and the target sensor. On the other hand, too many epochs may provide only marginal additional improvement, may increase training time unnecessarily, and may eventually reduce generalization if the model begins to fit details specific to the training set rather than the general pattern of healthy behavior. For this reason, the appropriate training duration cannot be fixed in advance and has to be examined experimentally.

For this experiment, the training duration is varied over a wide range. The tested values run from 100 to 2000 epochs, with increments of 100. Such a range is intended to reveal whether the model performance reaches a stable plateau after a limited number of optimization iterations or whether further training over substantially longer durations continues to produce meaningful gains. Since the

purpose of the dissertation is not only to improve prediction accuracy but also to consider the practical cost of model training and later embedded deployment, identifying a sufficiently effective rather than unnecessarily prolonged training regime is methodologically relevant.

The planned investigation of these two factors is organized as a controlled experiment in which one selected recurrent architecture is used while varying the sequence length and the number of training epochs. This design makes it possible to analyze the effect of temporal context and training duration without simultaneously changing the model type. The main objective of this experiment is to determine which sequence length provides the most informative historical context for virtual sensing and whether the selected training duration is sufficient for stable model convergence. In this way, the experiment serves to define a justified input-history configuration and a justified training schedule for the later stages of the dissertation.

In summary, input sequence length and number of training epochs are treated as experimentally testable design parameters. The working assumption is that an appropriate amount of historical context is necessary to represent the thermal dynamics of the monitored component, while an appropriate training duration is necessary to learn this relationship reliably without excessive optimization effort. The detailed numerical results of this comparison and the final choice of the sequence length and training duration are presented later in the experimental investigation chapter.

2.3.5. Selection of Alternative Model Structures for Investigation

The prediction accuracy of a virtual sensor depends not only on the selected input representation and training procedure, but also on the internal structure of the recurrent neural network itself. In sequence-to-one regression problems, the architecture determines how temporal dependencies are represented, how much latent information can be stored and transformed, and how complex the nonlinear mapping between the SCADA input sequence and the target sensor value can become. For this reason, model structure is treated in this dissertation as a separate experimental factor rather than as a fixed implementation choice.

Several structural properties of a recurrent network may influence its performance. One of the most important factors is the number of recurrent layers. A shallow architecture may be sufficient when the relationship between the input sequence and the target signal is relatively direct, but a deeper architecture may be more suitable when the model has to represent multiple levels of temporal abstraction. In a stacked recurrent network, lower layers may capture more local temporal patterns, whereas higher layers may encode more abstract or longer-range depen-

dencies. However, increasing the number of layers also increases the number of trainable parameters, makes optimization more difficult, and may increase the risk of overfitting. The influence of model depth therefore cannot be assumed in advance and has to be examined experimentally.

Another important structural factor is the number of recurrent cells in each layer. This choice sets the size of the hidden representation and, in practice, how much temporal structure the network can capture. With more cells, the model may be able to represent subtler nonlinear relationships between the selected SCADA features and the target thermal signal. The price of that added capacity is higher computational cost, greater memory use, and a higher chance that the network becomes larger than the task really requires. If too few cells are used, the model may miss relevant dynamics. If too many are used, the gain in prediction accuracy may be small compared with the added complexity. For this reason, the number of recurrent cells is treated as a controlled design variable in the experimental investigation.

The third structural factor is dropout. Although dropout is often treated as a training hyperparameter, it is closely related to the effective model structure because it modifies how the latent representation is regularized during learning. In recurrent sequence modeling, dropout may help reduce overfitting by discouraging excessive dependence on specific internal activations. Smaller dropout values may provide only weak regularization, especially in larger networks, whereas larger values may suppress useful information together with unwanted co-adaptation. The selected dropout level may, therefore, affect both the stability of training and the final generalization accuracy of the virtual sensor.

In the present dissertation, the planned experimental investigation of model structure is formulated as a controlled comparison of three structural factors: the number of recurrent layers, the number of recurrent cells per layer, and the dropout ratio. To avoid confounding the interpretation of the results, the comparison is defined using one recurrent cell type and varying the structural configuration around it. In the current experimental design, the selected recurrent cell type for this stage is BiLSTM, because it provides a flexible sequence representation and is suitable for examining how architectural complexity affects virtual sensor performance.

The planned depth experiment investigates models with one to four recurrent layers. The purpose of this comparison is to determine whether increasing the depth of the BiLSTM architecture improves sequence-to-one prediction accuracy for the selected SCADA input representation, or whether additional layers mainly increase complexity without a proportional gain in performance. The planned width experiment investigates a range of hidden-layer sizes, from 16 to 128 recurrent cells per layer. This comparison is intended to determine how much hidden-state capacity is required for accurate virtual sensing and whether the studied problem benefits from broader latent representations. The planned regulariza-

Table 2.5. Planned experimental design for the investigation of model structure

Experimental factor	Tested values	Purpose of the comparison
Recurrent cell type	BiLSTM	To fix one recurrent mechanism and isolate the effect of structural complexity.
Number of recurrent layers	1–4 layers	To determine whether increasing model depth improves sequence-to-one prediction accuracy or mainly increases architectural complexity.
Number of recurrent cells per layer	16, 32, 48, 64, 80, 96, 112, 128	To determine how much hidden-state capacity is required for accurate virtual sensing.
Dropout ratio	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	To determine the level of regularization that provides stable generalization without excessive information loss.
Evaluation objective	Prediction error of the virtual sensor on unseen data	To compare the effect of structural complexity and regularization on generalization quality.
Role in later dissertation stages	Selection of a preferred recurrent architecture	To define the model structure used in the subsequent experimental and computational-efficiency investigations.

tion experiment investigates dropout values from 0.2 to 0.8. The purpose of this comparison is to determine how strongly the model must be regularized to achieve stable generalization without suppressing too much useful temporal information.

The full plan of the structural comparison is summarized in Table 2.5. This experiment is designed to identify a model structure that provides a suitable balance between prediction accuracy, architectural complexity, and practical feasibility for later deployment-oriented analysis. In this sense, the structural investigation forms a bridge between the earlier investigation of input representation and the later investigation of computational simplification.

Within the theoretical framework of the dissertation, model structure is treated as an experimentally testable design space rather than as a single predefined architecture. The working assumption is that prediction accuracy depends on the interaction between model depth, hidden-state capacity, and regularization strength, and that these factors must be evaluated jointly to identify a practically justified recurrent architecture for virtual sensing. The detailed numerical results of this comparison and the final choice of the preferred model structure are presented later in the experimental investigation chapter.

2.3.6. Preliminary Theoretical Cycle Count for the Retained Activation Functions

This subsection introduces a preliminary theoretical cycle-count model for comparing the computational cost of the retained activation functions in recurrent virtual-sensor inference. The purpose of this model is not to reproduce the exact runtime of a specific processor, but to provide a consistent analytical basis for

comparing different activation choices from the embedded-implementation perspective.

The comparison is structured into three cases. In the first case, the standard logistic sigmoid and hyperbolic tangent functions are assumed to be evaluated directly, that is, using their original exponential-based form. This case provides the baseline reference and corresponds to the upper-cost scenario among the considered implementations. In the second case, the same standard activation functions are still used, but their evaluation is assumed to be accelerated by implementation techniques such as lookup tables, approximation polynomials, fixed-point approximations, or hardware-assisted mathematical libraries. In the third case, the standard nonlinearities are replaced by retained non-exponential bounded alternatives. This three-case structure is important because it separates two different engineering strategies: accelerating the original nonlinearities and replacing them with computationally simpler functions.

In the later experimental investigation, the activation-function comparison is performed using the best baseline recurrent architecture obtained with the standard activation functions. In the present theoretical section, the computational analysis is illustrated for a two-layer bidirectional LSTM model with 128 hidden units in the first layer, 100 hidden units in the second layer, 3 h aggregated inputs, and 11 input parameters. If the final experimental chapter reports a refined parameter combination, the same cost expressions remain valid after substituting the corresponding dimensional values.

A preliminary activation catalog was compiled for all candidate nonlinearities, and only the retained functions are considered in the present section. The catalog provides the scalar formula, the elementary operations required by one function call, and a preliminary weighted cost per call. The retained gate activations are the logistic sigmoid function, the hard sigmoid function, the softsign-based gate function, the arctangent gate function, and the algebraic gate function. The retained state activations are the hyperbolic tangent function, the softsign function, the hard hyperbolic tangent function, the arctangent function, and the algebraic state function.

Let the cost of elementary scalar operations be denoted by

$$c_{\mathrm{add}},\ c_{\mathrm{mul}},\ c_{\mathrm{div}},\ c_{\mathrm{abs}},\ c_{\mathrm{clip}},\ c_{\sqrt{\ }},\ c_{\arctan},\ c_{\exp}. \tag{2.28}$$

For the preliminary weighted comparison, the following normalized operation costs are used:

$$c_{\mathrm{add}} = c_{\mathrm{mul}} = c_{\mathrm{abs}} = 1, c_{\mathrm{clip}} = 2, c_{\mathrm{div}} = 4, c_{\sqrt{\ }} = 4, c_{\arctan} = 8, c_{\exp} = 10. \tag{2.29}$$

Table 2.6. Preliminary scalar weighted costs of the retained activation functions

Activation function	Elementary operations per call	Weighted cost
Logistic sigmoid (reference)	add:1, div:1, exp:1, mul:1	16
Hard sigmoid	add:1, clip:1, div:1	7
Softsign gate	abs:1, add:2, div:1, mul:1	8
Arctangent gate	add:1, atan:1, div:1	13
Algebraic gate	add:2, div:1, mul:2, sqrt:1	12
Hyperbolic tangent (reference)	add:2, div:1, exp:1, mul:1	17
Softsign	abs:1, add:1, div:1	6
Hard hyperbolic tangent	clip:1	2
Arctangent	atan:1, div:1, mul:1	13
Algebraic state	add:1, div:1, mul:1, sqrt:1	10

These values are not processor-specific measurements. They are used only to obtain a normalized relative estimate of scalar activation cost.

Using the operation counts provided in the activation catalog, the scalar costs of the retained activation functions are summarized in Table 2.6. These values correspond to the direct-evaluation case. They show that the standard logistic sigmoid and hyperbolic tangent functions remain the most expensive reference nonlinearities in the retained set, whereas the hard sigmoid and hard hyperbolic tangent provide the lowest scalar cost among the investigated alternatives.

The activation functions used in Table 2.6 are defined as follows.

The reference gate activation, the logistic sigmoid function, is

$$f_{\text{logistic sigmoid}}(x) = \frac{1}{1+e^{-x}}. \tag{2.30}$$

The reference state activation, the hyperbolic tangent function, is

$$f_{\text{hyperbolic tangent}}(x) = \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}. \tag{2.31}$$

The hard sigmoid function is defined as

$$f_{\text{hard sigmoid}}(x) = \begin{cases} 0, & x \le -2.5, \\ 0.2x + 0.5, & -2.5 < x < 2.5, \\ 1, & x \ge 2.5. \end{cases} \tag{2.32}$$

The hard hyperbolic tangent function is defined as

$$f_{\text{hard hyperbolic tangent}}(x) = \begin{cases} -1, & x < -1, \\ x, & -1 \le x \le 1, \\ 1, & x > 1. \end{cases} \tag{2.33}$$

The softsign function is defined as

$$f_{\text{softsign}}(x) = \frac{x}{1+|x|}. \tag{2.34}$$

The softsign gate function is obtained by shifting and scaling the softsign function to the interval $[0, 1]$:

$$f_{\text{softsign gate}}(x) = \frac{1}{2}\left(1 + \frac{x}{1+|x|}\right). \tag{2.35}$$

The arctangent state activation is defined as

$$f_{\text{arctangent}}(x) = \frac{2}{\pi}\arctan\left(\frac{\pi}{2}x\right). \tag{2.36}$$

The arctangent gate function is its affine transformation to the interval $[0, 1]$:

$$f_{\text{arctangent gate}}(x) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{\pi}{2}x\right)\right). \tag{2.37}$$

The algebraic state activation is defined as

$$f_{\text{algebraic state}}(x) = \frac{x}{\sqrt{1+x^2}}. \tag{2.38}$$

The algebraic gate function is obtained by shifting and scaling the algebraic state activation to the interval $[0, 1]$:

$$f_{\text{algebraic gate}}(x) = \frac{1}{2}\left(1 + \frac{x}{\sqrt{1+x^2}}\right). \tag{2.39}$$

In all equations above, x denotes the scalar pre-activation value. The gate activation functions map $\mathbb{R}$ to the interval $[0, 1]$, while the state activation functions map $\mathbb{R}$ to the interval $[-1, 1]$.

To make the computational analysis more transparent, it is useful to begin with the simplest recurrent case: a single-layer unidirectional LSTM with only one recurrent cell and the same number of input features as in the selected virtual-sensor model, namely $D = 11$. In this minimal configuration, the affine part of the one time-step update is

$$C_{\text{aff,cell}}(D) = 8(D+1), \tag{2.40}$$

Table 2.7. Preliminary weighted sequence costs for a single-layer LSTM with one recurrent cell and 11 input features

Gate activation	State activation	Sequence cost	Reduction vs. reference
Logistic sigmoid	Hyperbolic tangent	$182T$	reference
Hard sigmoid	Hard hyperbolic tangent	$125T$	31.32%
Softsign gate	Softsign	$136T$	25.27%
Arctangent gate	Arctangent	$165T$	9.34%
Algebraic gate	Algebraic state	$156T$	14.29%

because the four gate or candidate-state transformations are evaluated for one hidden unit only. The nonlinear and element-wise update part is

$$C_{\text{nonlin,cell}}(C_g, C_s) = 3C_g + 2C_s + 4. \tag{2.41}$$

Therefore, the total cost of a single-cell single-layer LSTM at one time step is

$$C_{1\text{cell}}(D; C_g, C_s) = 8(D+1) + 3C_g + 2C_s + 4. \tag{2.42}$$

For $D = 11$, this becomes

$$C_{1\text{cell}}(11; C_g, C_s) = 96 + 3C_g + 2C_s + 4. \tag{2.43}$$

The corresponding sequence cost for length T is

$$C_{1\text{cell}}(11, T; C_g, C_s) = T\left(100 + 3C_g + 2C_s\right). \tag{2.44}$$

The resulting costs for the retained activation pairs are summarized in Table 2.7. This table is useful because it isolates the activation-related effect before the structural matrix arithmetic becomes dominant in larger models.

The values in Table 2.7 show that, in a very small recurrent model, replacing exponential-based nonlinearities can noticeably reduce the total cost. This happens because, in the single-cell case, the nonlinear part still represents a substantial fraction of the full recurrent update.

After this simplest reference case, the same framework can be extended to a general recurrent layer. Let D_ℓ denote the input dimension of layer ℓ, H_ℓ the number of hidden units in that layer, and T the sequence length measured in aggregated input time steps. For one LSTM direction and one time step, the four affine transformations required by the input gate, forget gate, output gate, and candidate state are approximated by

$$C_{\text{aff}}^{(\ell)} = 8H_\ell(D_\ell + H_\ell), \tag{2.45}$$

where multiplication and addition are both counted with unit cost in the adopted normalized model.

If C_g denotes the scalar weighted cost of the selected gate activation and C_s the scalar weighted cost of the selected state activation, then the nonlinear and element-wise state-update cost for one LSTM direction and one time step is approximated as

$$C_{\text{nonlin}}^{(\ell)} = H_\ell \left(3C_g + 2C_s + 4\right). \tag{2.46}$$

The term $3C_g$ corresponds to the three gate nonlinearities, the term $2C_s$ corresponds to the candidate-state nonlinearity and the final state-output nonlinearity, and the constant 4 accounts for the remaining element-wise multiplications and additions in the state update.

Therefore, the preliminary cost of one single-layer unidirectional LSTM is

$$C_{\text{LSTM}}(D_\ell, H_\ell, T; C_g, C_s) = T H_\ell \left[8(D_\ell + H_\ell) + 3C_g + 2C_s + 4\right]. \tag{2.47}$$

If instead a single-layer bidirectional LSTM is used, the same computation is carried out in both directions, and the cost is doubled:

$$C_{\text{BiLSTM}}^{(\ell)}(D_\ell, H_\ell, T; C_g, C_s) = 2T \left[8H_\ell(D_\ell + H_\ell) + H_\ell \left(3C_g + 2C_s + 4\right)\right]. \tag{2.48}$$

For the two-layer reference architecture used in this dissertation, the first layer receives $D_1 = 11$ input features and has $H_1 = 128$ hidden units in each direction. Since the output of a bidirectional layer is concatenated, the second layer receives

$$D_2 = 2H_1 = 256 \tag{2.49}$$

input channels and has $H_2 = 100$ hidden units in each direction. Thus, the total preliminary cost of the full two-layer BiLSTM model becomes

$$\begin{aligned} C_{\text{2BiLSTM}}(T; C_g, C_s) = {} & C_{\text{BiLSTM}}^{(1)}(11, 128, T; C_g, C_s) \\ & + C_{\text{BiLSTM}}^{(2)}(256, 100, T; C_g, C_s). \end{aligned} \tag{2.50}$$

Substituting the selected dimensions yields

$$C_{\text{2BiLSTM}}(T; C_g, C_s) = T \left[854272 + 456 \left(3C_g + 2C_s + 4\right)\right]. \tag{2.51}$$

This expression is useful because it separates the total inference cost into two parts. The term 854272 corresponds to the matrix and structural arithmetic of the selected recurrent architecture, whereas the second term captures the contribution of the activation functions and the element-wise state updates. Therefore, replacing the activation functions affects only the second part of the total cost expression,

while the first part remains fixed for the selected model architecture. This observation also explains why a large scalar reduction in nonlinear cost may translate into a more moderate reduction at the full-model level.

The first comparison case corresponds to direct exponential-based evaluation. For the reference configuration with the standard logistic sigmoid and hyperbolic tangent functions, the scalar costs from Table 2.6 are $C_g = 16$ and $C_s = 17$. The resulting preliminary cost per sequence is

$$C_{\text{2BiLSTM}}^{\text{direct}}(T) = T\left[854272 + 456(3 \cdot 16 + 2 \cdot 17 + 4)\right] = 893488T. \tag{2.52}$$

The second comparison case corresponds to accelerated exponential-based evaluation. In this case, the model still uses the standard logistic sigmoid and hyperbolic tangent functions, but their implementation cost is reduced by techniques such as lookup tables, approximation methods, or hardware-optimized mathematical kernels. Let the corresponding accelerated scalar costs be denoted by C_g^{acc} and C_s^{acc}. Then the total preliminary cost becomes

$$C_{\text{2BiLSTM}}^{\text{acc}}(T) = T\left[854272 + 456\left(3C_g^{\text{acc}} + 2C_s^{\text{acc}} + 4\right)\right]. \tag{2.53}$$

This case is kept symbolic in the present theoretical chapter because the actual values of C_g^{acc} and C_s^{acc} depend on the selected embedded platform, numerical precision, and implementation library. In practical deployment, especially on ARM Cortex-M class microcontrollers, these values may differ substantially from the direct-evaluation case.

The third comparison case corresponds to the replacement of exponential-based nonlinearities by non-exponential bounded alternatives. For the bigger, two-layer architecture, if the standard functions are replaced by hard sigmoid and hard hyperbolic tangent, then $C_g = 7$ and $C_s = 2$, and the cost becomes

$$C_{\text{2BiLSTM}}^{\text{HSig-HTanh}}(T) = T\left[854272 + 456(3 \cdot 7 + 2 \cdot 2 + 4)\right] = 867496T. \tag{2.54}$$

The corresponding preliminary reduction with respect to the direct-evaluation baseline is

$$\Delta_{\%}^{\text{HSig-HTanh}} = 100 \cdot \frac{893488 - 867496}{893488} \approx 2.91\%. \tag{2.55}$$

If softsign-based functions are used in both gate and state roles, $C_g = 8$ and $C_s = 6$, yielding

$$C_{\text{2BiLSTM}}^{\text{Softsign}}(T) = T\left[854272 + 456(3 \cdot 8 + 2 \cdot 6 + 4)\right] = 872512T, \tag{2.56}$$

Table 2.8. Preliminary weighted sequence costs for the retained activation pairs in the reference two-layer BiLSTM with 11 input features and layer sizes 128 and 100

Gate activation	**State activation**	**Sequence cost**	**Reduction vs. direct baseline**
Logistic sigmoid	Hyperbolic tangent	$893488T$	reference
Hard sigmoid	Hard hyperbolic tangent	$867496T$	2.91%
Softsign gate	Softsign	$872512T$	2.35%
Arctangent gate	Arctangent	$885736T$	0.87%
Algebraic gate	Algebraic state	$881632T$	1.33%

with a preliminary reduction of

$$\Delta_{\%}^{\mathrm{Softsign}} \approx 2.35\%. \tag{2.57}$$

If arctangent-based functions are used in both roles, $C_g = 13$ and $C_s = 13$, yielding

$$C_{\mathrm{2BiLSTM}}^{\mathrm{Arctangent}}(T) = T\,[854272 + 456(3 \cdot 13 + 2 \cdot 13 + 4)] = 885736T, \tag{2.58}$$

with a preliminary reduction of

$$\Delta_{\%}^{\mathrm{Arctangent}} \approx 0.87\%. \tag{2.59}$$

If algebraic gate and algebraic state functions are used, $C_g = 12$ and $C_s = 10$, yielding

$$C_{\mathrm{2BiLSTM}}^{\mathrm{Algebraic}}(T) = T\,[854272 + 456(3 \cdot 12 + 2 \cdot 10 + 4)] = 881632T, \tag{2.60}$$

with a preliminary reduction of

$$\Delta_{\%}^{\mathrm{Algebraic}} \approx 1.33\%. \tag{2.61}$$

The resulting preliminary costs for the retained activation pairs are summarized in Table 2.8. These values should be interpreted as reductions relative to the direct-evaluation baseline. The table shows that the theoretical reduction in total sequence cost is smaller than the reduction observed at the scalar activation level. This is expected because for the selected two-layer BiLSTM architecture, the matrix arithmetic remains the dominant contribution to total inference cost. Nevertheless, the nonlinear part is still relevant, especially in repeated continuous inference and in low-power embedded execution.

It is also useful to examine the nonlinear part separately. For the standard logistic-sigmoid–hyperbolic-tangent configuration, the activation-related part of the selected model contributes

$$456(3 \cdot 16 + 2 \cdot 17 + 4) = 39216 \tag{2.62}$$

weighted cycles per aggregated time step. For the hard-sigmoid–hard-hyperbolic-tangent alternative, this part decreases to

$$456(3 \cdot 7 + 2 \cdot 2 + 4) = 13224, \tag{2.63}$$

which corresponds to a reduction of approximately 66.3% in the nonlinear component itself. Therefore, although the reduction in total model cost is more moderate, the effect of activation replacement remains theoretically meaningful and may become more pronounced on processors where nonlinear scalar operations are particularly expensive relative to matrix arithmetic.

From the dissertation perspective, the three-case interpretation is as follows. The direct exponential-based case defines the upper-cost baseline. The accelerated exponential-based case represents practical embedded implementations in which the same standard activation functions are still used, but their runtime is reduced by implementation methods. The non-exponential replacement case evaluates whether changing the activation function itself can provide an additional reduction beyond the direct baseline and, potentially, beyond the accelerated baseline. In this way, the activation catalog is useful because it makes it possible to propagate scalar activation costs into a model-level cycle-count estimate for the selected recurrent architecture.

The present theoretical investigation defines the expected direction and approximate magnitude of computational savings, but not their exact processor-specific value. The final evaluation of whether these theoretical savings are practically worthwhile, considering both the associated prediction-accuracy changes and the selected embedded implementation method, is presented later in the experimental investigation chapter.

2.4. Short-Term Wind Farm Power Generation Forecasting Problem

This section introduces the second application problem studied in the dissertation. The task is to forecast day-ahead energy generation for a small wind farm when detailed SCADA information about individual turbines is not available. In this case, the forecasting problem is affected by substantial uncertainty, and the prac-

tical objective is not only to reduce the prediction error itself but also to reduce the financial losses associated with market imbalance. Wind farm power forecasting is, therefore, treated as a regression problem in which future generated power is predicted from historical plant-level measurements and meteorological forecast inputs.

2.4.1. Problem Setting in Small Wind Farms with Limited Observability

The forecasting problem considered in this dissertation arises in small wind farms where only aggregate plant-level generation data are available, while detailed turbine-level SCADA measurements are not accessible. As a result, the prediction model has limited visibility into the condition, availability, maintenance status, and directional behavior of individual turbines. This makes the forecasting task more difficult than in fully instrumented wind farms, because several important operational factors remain unobserved.

Let P_t denote the generated power at time t, and let h denote the forecast horizon. A general day-ahead forecasting model can be written as

$$\hat{P}_{t+h} = g_\phi(z_{t-L+1}, \ldots, z_t,\ w_{t:t+h})\,, \tag{2.64}$$

where P_t and $\hat{P}_{t+h}$ denote measured and predicted power in the same physical unit, typically kW or MW, z denotes historical measurements, and w denotes meteorological forecast features over the forecasting interval. This formulation makes explicit that the predictor combines two information sources: recent operational history and atmospheric forecasts. Their relative importance depends on the forecast horizon. Very short horizons are more strongly influenced by recent measurements, whereas day-ahead forecasting depends much more on the quality and representativeness of the meteorological input.

When turbine-level operating states are not directly observed, several uncertainty sources remain unresolved. These include technical maintenance periods, partial unavailability of individual turbines, unobserved curtailment, turbine-specific sensitivity to wind direction, and local operating effects that are not visible in aggregate production data alone. Consequently, the model must learn plant-level behavior under incomplete observability, which increases forecast uncertainty and motivates explicit investigation of the role of meteorological inputs and loss-aware evaluation criteria.

Table 2.9. Characteristics of the numerical weather prediction sources used in the dissertation

NWP model name	Temporal resolution	Spatial resolution	Update rate / time	Key output parameters used
ECMWF-ifsa (IFS)	1 h	7.6 km	2/day (06, 18 UTC)	Wind speed, wind gust, wind direction, temperature
Ukmo-euro4b (EURO)	1 h	3.05 km	4/day (00, 06, 12, 18 UTC)	Wind speed, wind gust, wind direction, temperature
MEPS	1 h	2.5 km	4/day (00, 06, 12, 18 UTC)	Wind speed, wind gust, wind direction, temperature
ICON	Hourly (typical)	Varies (depending on provider)	Varies (depending on provider)	Wind speed (10 m, 80 m), wind gust, sea level pressure, temperature, humidity
GEM Global	Hourly (typical)	Varies (depending on provider)	Varies (depending on provider)	Wind speed (10 m, 80 m), wind gust, sea level pressure, temperature, humidity
Meteo France	Hourly (typical)	Varies (depending on provider)	Varies (depending on provider)	Wind speed (10 m, 80 m), wind gust, sea level pressure, temperature, humidity
GFS Global	Hourly (typical)	Varies (depending on provider)	Varies (depending on provider)	Wind speed (10 m, 80 m), wind gust, sea level pressure, temperature, humidity
Best Match	Hourly (typical)	Specific to the site (ensemble)	Varies (depending on provider)	Wind speed (10 m, 80 m), wind gust, sea level pressure, temperature, humidity

2.4.2. Numerical Weather Prediction Inputs and Data Preparation

Since detailed turbine-level operating data are unavailable, numerical weather prediction inputs become the main external information source for day-ahead power forecasting. Their quality has a direct influence on forecast performance because different NWP models differ in spatial resolution, temporal resolution, update frequency, and regional suitability. For this reason, the selection of the meteorological source is treated in this dissertation as a modeling decision rather than as a neutral data-access step.

Numerical weather prediction models provide key variables for forecasting, such as wind speed, wind gust, wind direction, temperature, humidity, and pressure. Typical preprocessing includes synchronization of weather forecasts with plant timestamps, alignment of forecast horizons, and feature scaling for stable model training. Because different forecast providers and models may exhibit different biases and spatial representativeness, the same nominal weather variable may carry different predictive value depending on the source from which it is obtained.

Table 2.9 summarizes the characteristics of the NWP sources considered in the research. The table combines two groups of sources. The first group includes

representative NWP models frequently discussed in the literature, while the second group includes forecast sources that were actually available through the data-access workflow used in the dissertation. This distinction is important because it explains why the theoretical chapter first compares the general characteristics of NWP inputs and only later experimentally evaluates the subset that was available for model training.

Table 2.9 shows that the candidate NWP inputs differ not only by provider name, but also by temporal resolution, spatial resolution, and update frequency. These differences justify treating the meteorological source as a modeling factor rather than as an interchangeable background setting. In particular, a source that is updated more frequently or more closely matched to the site may provide more relevant short-term atmospheric information than a coarser or less frequently refreshed model. Therefore, the experimental chapter later treats NWP source selection as one of the investigated factors affecting forecast quality.

2.4.3. Forecasting Workflow and Model Selection

Forecasting models for wind power range from linear baselines to nonlinear machine-learning and deep-learning methods. In this dissertation, recurrent neural-network models based on BiLSTM structures are used because they can exploit temporal dependencies in both directions during training and are well-suited for learning nonlinear relationships between recent plant behavior and weather forecasts.

The forecasting workflow considered in the dissertation consists of several connected stages: synchronization of historical and meteorological inputs, feature preparation, recurrent prediction, and forecast evaluation. This is illustrated in Figure 2.1. The figure emphasizes that forecast performance depends not only on the recurrent predictor itself, but also on the quality and consistency of the full data-processing chain.

Since model performance depends on architectural choices and training hyperparameters, Bayesian optimization is used in the dissertation as a sequential validation-driven search procedure. After each trial, the observed validation error is fed back to the optimizer, which proposes the next configuration in a more promising region of the hyperparameter space. This avoids exhaustive enumeration of all combinations while still making it possible to identify one best-performing configuration before the final experimental comparisons. In this way, the forecasting problem is treated not only as a model-fitting task, but also as a structured design problem involving the choice of meteorological source, recurrent architecture, and training configuration.

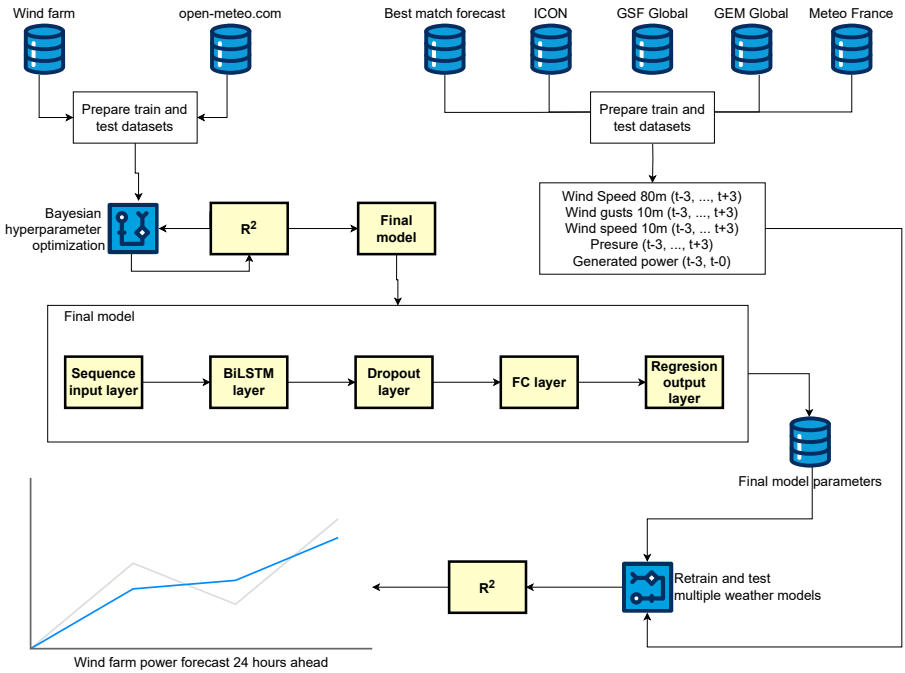


Fig. 2.1. Structure of the wind farm power forecasting model

2.4.4. Forecast Error, Economic Interpretation, and Evaluation Criteria

The conventional objective in regression-based forecasting is to minimize the numerical difference between predicted and actual energy generation. In recurrent forecasting models such as LSTM or BiLSTM, this is usually done by optimizing a standard regression loss that penalizes forecast residuals. Such losses are mathematically convenient and widely used, but they do not directly represent the financial consequences of forecast errors in electricity markets.

In practical day-ahead operation, forecast deviations lead to balancing costs because the scheduled energy differs from the actually delivered energy. The economic effect of forecast error depends not only on its magnitude, but also on its direction and on the market context at the settlement time. The same absolute prediction error may therefore lead to different operational consequences depending on whether generation is underpredicted or overpredicted and on the penalty

structure applied to the imbalance. This motivates the need to interpret forecast residuals not only statistically, but also economically.

Forecast errors may also be expressed in different forms, for example, as absolute deviations, normalized deviations, or relative deviations with respect to the forecasted or actual generation level. These representations are not equivalent, especially when large and small production levels are compared under similar uncertainty. For this reason, the dissertation treats the economic meaning of forecast error as an important part of the theoretical formulation, even when the final model is still trained using a standard regression loss.

Forecast accuracy is commonly evaluated using the root mean square error, the mean absolute error, and the coefficient of determination:

$$\begin{aligned} \mathrm{RMSE} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_i - \hat{P}_i)^2}, \\ \mathrm{MAE} &= \frac{1}{N}\sum_{i=1}^{N}|P_i - \hat{P}_i|, \end{aligned} \tag{2.65}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(P_i - \hat{P}_i)^2}{\sum_{i=1}^{N}(P_i - \bar{P})^2}. \tag{2.66}$$

In these metrics, N is the number of evaluated forecast samples, P_i and $\hat{P}_i$ are the measured and predicted power values, and $\bar{P}$ is the sample mean of the measured power. Accordingly, RMSE and MAE inherit the unit of power, whereas R^2 is dimensionless.

From the market perspective, forecast error can be translated into a generic imbalance-cost functional:

$$C = \sum_{t} \pi_t \left| P_t - \hat{P}_t \right|, \tag{2.67}$$

where C is the accumulated imbalance cost and π_t is a time-dependent market-price weighting factor. In the implementation adopted in this dissertation, π_t is derived from the Nord Pool day-ahead electricity price of the forecasted delivery hour: the hourly price series is shifted by one hour so that each training sample is paired with the price of its prediction target, and the obtained price term is then normalized before being used as a weighting coefficient in the proposed objective function. Accordingly, π_t should be interpreted as a dimensionless price-based penalty factor rather than as an unscaled raw electricity price. This formulation does not yet define one specific market rule, but it makes explicit that the practical objective of wind power forecasting is not limited to minimizing statistical error

alone. Instead, the forecasting model should ultimately be assessed with respect to both predictive accuracy and its economic implications under balancing-market conditions.

2.5. Conclusions of the Second Chapter

1. Given the rarity of fault events and the wide operating range of the monitored temperature signal, the most suitable strategy for early fault detection is to estimate the expected sensor response under normal conditions and detect deviations from it using a virtual sensor.
2. The obtained rankings confirm that a substantial part of the predictive information is concentrated in variables describing turbine load and operating regime, while the final selected input set should provide a compact and physically justified representation of the excitation conditions affecting the target generator bearing temperature sensor.
3. The preliminary scalar-cost analysis shows that the standard logistic sigmoid and hyperbolic tangent remain the most expensive retained activation functions, with weighted costs of 16 and 17, respectively, whereas the lowest costs are achieved by the hard sigmoid and hard hyperbolic tangent, with weighted costs of 7 and 2, respectively, indicating a substantial theoretical reduction in nonlinear evaluation cost at the function level.
4. The single-cell single-layer LSTM analysis demonstrates that activation replacement can significantly reduce the total recurrent update cost in small models, since the sequence cost decreases from 182T for the logistic-sigmoid–hyperbolic-tangent baseline to 125T for the hard-sigmoid–hard-hyperbolic-tangent pair, corresponding to a reduction of 31.32%, while the softsign-, algebraic-, and arctangent-based alternatives yield smaller but still measurable reductions of 25.27%, 14.29%, and 9.34%, respectively.
5. For the selected two-layer BiLSTM architecture with 11 input features and layer sizes 128 and 100, the total sequence cost decreases from 893488T for the logistic-sigmoid–hyperbolic-tangent baseline to 867496T for the hard-sigmoid–hard-hyperbolic-tangent pair, which corresponds to a reduction of 2.91%; however, the activation-related component alone decreases from 39216 to 13224 weighted cycles per aggregated time step, that is, by 66.3%, indicating that activation replacement remains theoretically relevant even when the full-model gain is limited by affine matrix arithmetic.

3

Experimental Research on Forecasting and Fault Detection Methods

This chapter reports experimental studies on three tasks: anomaly detection of wind turbine drivetrain components from SCADA data, virtual sensor modeling from SCADA data using recurrent networks with modified activation functions for efficient inference, and wind farm power forecasting over short horizons using BiLSTM models and NWP inputs. The experiments use public SCADA data from the EDP Wind Turbine Failure Detection Challenge and measurements from an operational Lithuanian wind farm synchronized with forecasts obtained through the Open-Meteo API. The results of the investigations presented in this chapter were published in four scientific publications (Jankauskas et al., 2023a; Jankauskas et al., 2024; Jankauskas et al., 2026).

3.1. Experimental Investigation of the Virtual Sensor for Condition Monitoring and Early Fault Detection

This section presents the experimental investigation of the virtual sensor proposed for data-driven condition monitoring and early fault detection. The main purpose of the experiments is to evaluate how accurately the virtual sensor can reproduce the monitored physical signal under normal operating conditions and how reliably the resulting residual can be used as an indicator of abnormal behavior. In accor-

dance with the theoretical framework introduced in Section 2.2, the virtual sensor is treated as a sequence-based regression model that estimates the target sensor value from selected SCADA inputs, while the difference between the measured and predicted values is used as the diagnostic residual.

The experimental investigation is organized as a stepwise analysis of the main factors that may influence the quality of virtual sensing and, consequently, the reliability of residual-based anomaly detection. First, the selected SCADA input representation and data preparation choices are evaluated. Next, the influence of temporal aggregation, training data composition, sequence length, number of training epochs, and recurrent model structure is analyzed. After identifying the best-performing baseline model with standard activation functions, an additional experimental comparison is carried out for alternative activation functions to assess the trade-off between computational simplicity and prediction accuracy. Finally, the residual behavior of the selected model is examined from the early fault detection perspective by analyzing how the prediction error changes as the monitored component approaches failure.

These experiments are intended to validate both parts of the proposed framework: the virtual sensor itself as an accurate estimator of the target variable during healthy operation, and the residual derived from this estimator as a practical signal for early anomaly indication. The detailed results presented in the following subsections provide the basis for selecting the most suitable model configuration for the later stages of the dissertation.

3.1.1. Effect of Temporal Aggregation on Virtual Sensor Accuracy

This experiment investigated how the additional temporal aggregation of SCADA inputs affects the accuracy of the virtual sensor when different recurrent neural-network structures are used. The target variable was the SCADA channel listed in the public EDP dataset as *gearbox bearing temperature*, and the same selected SCADA input set was used in all compared cases. The selected input variables were maximum, minimum, mean, and standard deviation of generator rotational speed; maximum, minimum, and mean of rotor rotational speed; maximum, minimum, mean, and standard deviation of ambient wind speed; relative and absolute wind direction; and mean ambient temperature. These variables were extracted from the public EDP wind turbine dataset and were then additionally aggregated over three alternative intervals: 24 h, 6 h, and 3 h.

The aggregation experiment was carried out with four recurrent model structures introduced in the theoretical chapter: a single-layer GRU model with 128 recurrent cells, a single-layer BiLSTM model with 128 recurrent cells, a two-layer LSTM–LSTM model with 128 and 100 recurrent cells, and a two-layer LSTM–

BiLSTM model with 128 and 128 recurrent cells. In the experiment worksheet, the last structure was stored under the label *LSTM*, but in the present dissertation, it is referred to explicitly as the LSTM–BiLSTM architecture. The comparison included multiple train–test scenarios and several test turbines, and the prediction accuracy was evaluated using RMSE.

Table 3.1 summarizes the average RMSE values obtained for each aggregation interval and each neural-network structure. A clear pattern emerges. For all four analyzed structures, the 3 h aggregation interval yielded the lowest average RMSE, whereas the 6 h interval produced the highest average RMSE. The 24 h interval remained intermediate in all cases. Moderate temporal aggregation, therefore, improved the prediction accuracy of the virtual sensor more effectively than either stronger smoothing over 24 h or the intermediate 6 h aggregation level.

Table 3.1. Average RMSE of the virtual sensor for different temporal aggregation intervals and recurrent model structures

Model structure	24 h	6 h	3 h	Best interval
LSTM–BiLSTM (128, 128)	3.8467	3.9664	3.5695	3 h
BiLSTM (128)	4.3004	5.2343	3.9517	3 h
GRU (128)	4.6745	4.9331	3.8933	3 h
LSTM–LSTM (128, 100)	2.9989	4.0180	2.8595	3 h

The average results also make it possible to compare the sensitivity of the analyzed model structures to the aggregation interval. In this comparison, the two-layer LSTM–LSTM model was the strongest option throughout: its average RMSE was 2.9989 at 24 h, 4.0180 at 6 h, and 2.8595 at 3 h. The next best results came from the LSTM–BiLSTM model, with corresponding values of 3.8467, 3.9664, and 3.5695. The single-layer BiLSTM and GRU models reacted more strongly to the choice of aggregation interval and produced higher average RMSE values, especially at 6 h. For example, the average RMSE of the BiLSTM model increased from 3.9517 at 3 h to 5.2343 at 6 h, while the corresponding GRU values changed from 3.8933 to 4.9331.

Averaging over all tested model structures and evaluation cases leads to the same conclusion. The overall mean RMSE was 3.5685 for the 3 h aggregation interval, 3.9551 for the 24 h interval, and 4.5380 for the 6 h interval. Across the full aggregation experiment, the 3 h interval improved the average prediction accuracy by approximately 9.8% relative to 24 h and by approximately 21.4% relative to 6 h.

To complement the average comparison, Table 3.2 presents the best-performing results obtained for each model structure. Two criteria are reported: the best average aggregation interval for each structure and the lowest individual RMSE achieved in any single evaluation case. This distinction is important because the interval with the best average behavior is not necessarily the same as the interval that yields the absolute minimum RMSE in one isolated case.

Table 3.2. Best temporal aggregation results for the analyzed recurrent model structures

Model structure	Best average interval	Average RMSE	Best individual RMSE	Best individual case
GRU (128)	3 h	3.8933	1.8058	6 h, T09, 6m/3m
BiLSTM (128)	3 h	3.9517	1.7921	3 h, T09, 6m/3m
LSTM–LSTM (128, 100)	3 h	2.8595	1.6925	3 h, T09, 6m/3m
LSTM–BiLSTM (128, 128)	3 h	3.5695	1.7735	24 h, T09, 6m/3m

The best overall average result of the whole experiment was obtained with the two-layer LSTM–LSTM architecture and the 3 h aggregation interval, which achieved an average RMSE of 2.8595. The best individual result was also obtained by the same architecture, again with 3 h aggregation, reaching RMSE = 1.6925 in the 6m/3m setup on turbine T09. These results indicate that the 3 h aggregation interval not only improved the average behavior of all tested models, but also supported the strongest overall result observed in the experiment.

From the modeling perspective, these findings suggest that the 3 h aggregation interval provides a more suitable balance between preserving relevant operational dynamics and suppressing short-term fluctuations that are less useful for temperature prediction. The 24 h aggregation interval still produced competitive results for some stacked architectures, which indicates that stronger smoothing can remain beneficial in selected cases. However, the 6 h interval was consistently the weakest option in the average comparison, suggesting that it may remove part of the useful short-term variation without providing the same stability as the longer 24 h aggregation.

The aggregation investigation shows that the temporal representation of the SCADA inputs has a measurable influence on virtual-sensor accuracy, and that this influence is observed across all analyzed recurrent architectures. Since the 3 h aggregation interval yielded the best average RMSE for each of the four tested neural-network structures and also supported the strongest overall result, it was selected as the preferred aggregation setting for the subsequent stages of the experimental investigation.

3.1.2. Effect of Training Data Composition

This experiment investigated how the composition of the training dataset influences the prediction accuracy of the virtual sensor when the selected 3 h aggregation interval is used. The input variables were the same as in the previous experiments and were derived from the public EDP wind turbine dataset. The input set consisted of maximum, minimum, mean, and standard deviation of generator rotational speed; maximum, minimum, and mean of rotor rotational speed; maximum, minimum, mean, and standard deviation of ambient wind speed; relative

and absolute wind direction; and mean ambient temperature. The target variable was the SCADA channel listed in the dataset as *gearbox bearing temperature*.

This comparison had two stages. In the first part, the virtual sensor was trained only on historical data from turbine T01 and then tested on turbines T01, T06, and T09. Two train–test configurations were compared: 3m/3m, in which 3 months of healthy historical data were used for training and the following 3 months were used for testing, and 1m/5m, in which only 1 month of data was used for training and testing was carried out over 5 months of later data. This comparison was intended to examine whether a short 1-month training interval is sufficient for constructing a stable virtual sensor and whether the model remains accurate when applied over a longer future period.

In the second part of the experiment, historical data from turbines T01, T06, and T09 were combined into one training time series, and the resulting virtual sensor was then tested separately on each of these turbines. Two training-history lengths were analyzed: 3m/3m and 6m/3m. In this part of the experimental investigation, more detailed statistics were evaluated, including RMSE, standard deviation of the prediction error, and the minimum and maximum residual values.

The results of the first part of the experiment are summarized in Table 3.3. For all analyzed neural-network structures, the 3m/3m setup outperformed the 1m/5m setup when the model was trained only on T01 data. The best overall result in this part of the experimental investigation was obtained by the two-layer LSTM–LSTM architecture with 128 and 100 recurrent cells. In the 3m/3m setup, this model achieved RMSE values of 2.7220 on T01, 2.4652 on T06, and 2.4437 on T09, yielding the lowest cross-turbine average RMSE of 2.5436. When only 1 month of training data was used, and the prediction horizon was extended to 5 months, the same architecture produced higher RMSE values of 3.7906, 4.1849, and 3.8594, respectively, with an average of 3.9450. Even for the best-performing model, the shorter training interval increased the average error by approximately 35.5%.

Table 3.3. RMSE results for the virtual sensor trained on turbine T01 and tested on turbines T01, T06, and T09 at the 3 h aggregation interval

Model structure	Train/Test	T01	T06	T09	Average RMSE
LSTM–BiLSTM (128, 128)	3m/3m	3.9368	3.4067	3.2016	3.5150
BiLSTM (128)	3m/3m	3.7684	3.1931	4.1896	3.7170
GRU (128)	3m/3m	3.4737	3.0610	3.5224	3.3524
LSTM–LSTM (128, 100)	3m/3m	2.7220	2.4652	2.4437	2.5436
LSTM–BiLSTM (128, 128)	1m/5m	4.0468	4.3526	4.4896	4.2963
BiLSTM (128)	1m/5m	5.4163	6.1083	6.5731	6.0326
GRU (128)	1m/5m	5.7227	6.9166	6.4021	6.3471
LSTM–LSTM (128, 100)	1m/5m	3.7906	4.1849	3.8594	3.9450

The same tendency can be observed for the other model structures. Relative to the 1m/5m setup, the average RMSE in the 3m/3m setup decreased by approximately 18.2% for LSTM–BiLSTM, 38.4% for BiLSTM, 47.2% for GRU, and 35.5% for LSTM–LSTM. These results indicate that one month of healthy historical data is not sufficient for stable virtual-sensor training, even if the risk of component degradation during that interval is relatively low. The experiment suggests that a longer healthy-state history is needed in order to expose the model to a broader range of normal operating regimes and thereby improve generalization to later unseen data.

The second part of the experiment used the combined historical data of turbines T01, T06, and T09 as one training series and then evaluated the trained model separately on each turbine. The 3m/3m results are presented in Table 3.4. In this setting, the two-layer LSTM–LSTM model again achieved the best overall average performance, with RMSE values of 3.0232 for T01, 2.6230 for T06, and 2.2984 for T09. The corresponding average RMSE was 2.6482, with an average residual standard deviation of 1.9974. Among the single-layer models, GRU performed better than BiLSTM on average, while LSTM–BiLSTM achieved the highest average RMSE in this configuration.

Table 3.4. Detailed results for combined training on turbines T01, T06, and T09 with the 3m/3m setup at the 3 h aggregation interval

Model structure	Test turbine	RMSE	STD	Min	Max
LSTM–BiLSTM (128, 128)	T01	4.2170	3.5122	0.0034	39.4883
LSTM–BiLSTM (128, 128)	T06	3.4112	2.7800	0.0006	23.6484
LSTM–BiLSTM (128, 128)	T09	3.2219	2.5386	0.0002	34.4981
BiLSTM (128)	T01	3.7157	3.1804	0.0041	57.3204
BiLSTM (128)	T06	2.6322	1.9502	0.0018	13.6757
BiLSTM (128)	T09	4.2663	3.9127	0.0015	64.0744
GRU (128)	T01	3.4275	2.8497	0.0003	51.5149
GRU (128)	T06	2.6379	1.9418	0.0022	15.2266
GRU (128)	T09	3.8232	3.2588	0.0125	57.5066
LSTM–LSTM (128, 100)	T01	3.0232	2.3045	0.0018	29.6387
LSTM–LSTM (128, 100)	T06	2.6230	1.8481	0.0042	14.9412
LSTM–LSTM (128, 100)	T09	2.2984	1.8397	0.0013	27.0365

When the training period was extended from 3 months to 6 months, and testing was still performed over 3 months, the prediction accuracy improved further for all analyzed models. The results are presented in Table 3.5. The best overall result of the whole training-composition experiment was obtained by the two-layer LSTM–LSTM model, which achieved RMSE values of 2.6518 on T01, 2.5597 on T06, and 1.6925 on T09, with the lowest average RMSE of 2.3013. In this configuration, the same model also achieved the lowest residual standard deviation on

average, namely 1.5941, and the lowest maximum residual among the compared architectures.

Table 3.5. Detailed results for combined training on turbines T01, T06, and T09 with the 6m/3m setup at the 3 h aggregation interval

Model structure	Test turbine	RMSE	STD	Min	Max
LSTM–BiLSTM (128, 128)	T01	3.3694	2.6264	0.0026	20.7748
LSTM–BiLSTM (128, 128)	T06	3.2706	2.5443	0.0001	19.2259
LSTM–BiLSTM (128, 128)	T09	1.9095	1.4947	0.0012	11.5805
BiLSTM (128)	T01	3.0609	2.3946	0.0039	16.9008
BiLSTM (128)	T06	2.7046	1.9439	0.0012	10.7327
BiLSTM (128)	T09	1.7921	1.3380	0.0033	10.2677
GRU (128)	T01	3.1423	2.4403	0.0023	24.0636
GRU (128)	T06	2.7400	1.9389	0.0015	9.9756
GRU (128)	T09	1.8501	1.3250	0.0058	10.4159
LSTM–LSTM (128, 100)	T01	2.6518	1.9071	0.0208	14.9322
LSTM–LSTM (128, 100)	T06	2.5597	1.7827	0.0011	13.4841
LSTM–LSTM (128, 100)	T09	1.6925	1.0925	0.0027	7.0366

Averaging the results across the three test turbines confirms the advantage of longer and more diverse training data. For the combined-training setup, the average RMSE decreased from 3.6167 to 2.8498 for the LSTM–BiLSTM model, from 3.5381 to 2.5192 for BiLSTM, from 3.2962 to 2.5775 for GRU, and from 2.6482 to 2.3013 for LSTM–LSTM when the training history was extended from 3 to 6 months. These changes correspond to relative improvements of approximately 21.2%, 28.8%, 21.8%, and 13.1%, respectively.

The training-data-composition experiment leads to two main observations. First, training on only 1 month of historical data is not sufficient for robust virtual-sensor prediction when the model is expected to generalize over a longer future interval, even if the short training period is less likely to include component degradation. Second, combining healthy historical data from several turbines and extending the training history to 6 months improves cross-turbine robustness and leads to the best overall performance, especially when the two-layer LSTM–LSTM architecture is used. The best result obtained in this experiment was achieved with the LSTM–LSTM structure trained on the combined T01, T06, and T09 dataset for 6 months and tested over 3 months, yielding RMSE = 2.3013 on average across the three turbines and a minimum individual RMSE of 1.6925 on turbine T09.

3.1.3. Effect of Input Sequence Length and Number of Training Epochs

This experiment investigated how the length of the input sequence and the number of training epochs affect the prediction accuracy of the virtual sensor. In this part of

the experimental investigation, one recurrent model structure was fixed in advance to isolate the influence of the temporal input horizon and training duration. The selected model was a three-layer BiLSTM architecture with 80 recurrent cells in each layer and dropout equal to 0.2. The input data were formed from the selected SCADA features aggregated over 3 h intervals, and the objective was to determine how many consecutive aggregated input vectors provide the most suitable temporal context for predicting the real sensor value with minimum error.

Five sequence lengths were analyzed: 1, 3, 6, 12, and 24 aggregated input vectors. Since each vector represents a 3 h summary of the SCADA signals, these sequence lengths correspond to 3 h, 9 h, 18 h, 36 h, and 72 h of historical context, respectively. In addition, the number of training epochs was varied from 100 to 2000 in steps of 100. The prediction quality was evaluated using RMSE, the standard deviation of the residuals, and the minimum and maximum residual values.

Table 3.6 summarizes the RMSE values obtained for all combinations of sequence length and training epochs. Both factors strongly influenced the final prediction accuracy, and their interaction was highly nonlinear. At 100 epochs, the RMSE already ranged from 2.1671 for the 3-value sequence to 8.3222 for the 24-value sequence. The strongest instability was observed for the longer temporal contexts, especially 12 and 24 values: the 24-value sequence produced repeated large error spikes at 200, 400, and 600 epochs (RMSE = 8.6078, 8.3947, and 6.1918), while the 12-value sequence started from RMSE = 8.2186 before converging later. By comparison, the 1- and 3-value sequences behaved more steadily in the early stage of training, while the 6-value sequence settled quickly after the initial transient.

The lowest single RMSE of the whole sweep was obtained with a sequence length of 12 at 1800 epochs, where the virtual sensor achieved RMSE = 1.0440. The same configuration also produced residual standard deviation = 0.8264, minimum residual = 0.000061, and maximum residual = 5.1744. However, the mature part of the sweep shows that the practically relevant differences between the strongest configurations are small. The best RMSE values for sequence lengths 3, 6, 12, and 24 are 1.0943, 1.0673, 1.0440, and 1.1021, respectively, so the spread between these four best mature configurations is only 0.0581 °C. This is below one tenth of a degree, and smaller than the 0.22–0.89 °C pre-failure median shifts discussed earlier. RMSE alone is therefore not sufficient to claim a practically large superiority of one temporal context; the stability across training epochs and the residual dispersion must also be considered.

This does not mean that RMSE is an invalid metric in the present experiment. On the contrary, RMSE remains useful for identifying clearly weak or unstable settings. For example, the longer 24-value sequence can be recognized immediately as problematic at several intermediate training stages, whereas the 3-, 6-, and 12-value sequences all converge into the strong-performance region after sufficient

Table 3.6. RMSE values for different sequence lengths and training epochs in the three-layer BiLSTM model with 80 recurrent cells per layer and dropout 0.2

Epochs	1 value	3 values	6 values	12 values	24 values
100	2.2095	2.1671	4.7405	8.2186	8.3222
200	2.1800	1.8417	1.7670	2.5796	8.6078
300	2.1497	1.7788	1.6134	1.6079	2.1531
400	2.1126	1.6886	1.5818	1.4590	8.3947
500	2.1154	1.6244	1.4663	1.4297	1.6448
600	2.0460	1.5119	1.3317	1.3288	6.1918
700	4.0679	1.4425	1.1975	1.2229	1.5692
800	1.9491	1.4264	1.2646	1.1948	1.3598
900	1.9724	1.2680	1.1929	1.1472	1.1652
1000	1.9188	1.3098	1.2256	1.1738	1.2748
1100	1.8826	1.2307	1.2021	1.1219	1.3072
1200	1.8072	1.2294	1.0956	1.1819	1.1050
1300	1.8037	1.1477	1.1004	1.1593	1.1021
1400	1.8324	1.1506	1.0716	1.1441	1.1184
1500	1.7584	1.0943	1.0847	1.0908	1.1194
1600	1.6943	1.1106	1.0673	1.1534	1.1565
1700	1.6680	1.1204	1.1502	1.0623	1.1472
1800	1.7590	1.1063	1.1070	1.0440	1.1361
1900	2.9929	1.1448	1.1112	1.1033	1.1132
2000	1.6928	1.1509	1.0869	1.0908	1.1864

training. The limitation appears only when the comparison is restricted to the best mature configurations, because in that regime, the remaining RMSE differences become too small to support a strong practical preference on their own.

To analyze the influence of sequence length more systematically, Table 3.7 presents the average RMSE and average residual standard deviation across all tested epoch counts, together with the best result obtained for each sequence length. Across the full experiment, the 3-value sequence had the best average behavior, with average RMSE = 1.3772 and average residual standard deviation = 0.9226. This indicates that a 9 h historical context provides the most stable performance when the entire range of training epochs is considered. The 6-value sequence was a close second, with average RMSE = 1.4229 and average residual standard deviation = 0.9583, while also reaching a stronger best-case RMSE of 1.0673 than the 3-value sequence. In practical terms, 6 values can be interpreted as the most balanced compromise between robustness and peak accuracy. The 12-value sequence yielded the global minimum RMSE, but its average behavior was worse than that of the 3- and 6-value sequences because it required longer training and exhibited stronger fluctuations. The 24-value sequence improved substantially after sufficient training, yet remained the least stable option on average, with RMSE = 2.6087.

Table 3.7. Results summarized by sequence length for the three-layer BiLSTM model with 3 h aggregated inputs

Sequence length	Avg. RMSE	Avg. STD	Best epoch	Best RMSE	Min	Max
1 value	2.0806	1.3663	1700	1.6680	0.001747	6.7067
3 values	1.3772	0.9226	1500	1.0943	0.001122	5.0452
6 values	1.4229	0.9583	1600	1.0673	0.001553	5.8371
12 values	1.6257	1.0830	1800	1.0440	0.000061	5.1744
24 values	2.6087	1.5317	1300	1.1021	0.001064	6.2592

The effect of the number of training epochs can be analyzed from another perspective by averaging the results across all tested sequence lengths. Table 3.8 shows that the average RMSE decreased from 5.1316 at 100 epochs to 1.2838 at 1200 epochs and then remained within 1.2295–1.2634 from 1200 to 1800 epochs, apart from a temporary deterioration at 1900 epochs. The average residual standard deviation followed the same tendency, decreasing from 2.6440 to 0.9144 at 1200 epochs and then remaining close to 0.90–0.92 in the same mature-training region. After 1200 epochs, the best-performing sequence alternated between 6 and 12 values rather than being dominated by a single temporal context. This shows that longer training was necessary for this model structure, but also that the experiment reached a practical plateau after about 1200 epochs and that additional training beyond this point provided only marginal RMSE gains.

Table 3.8. Results summarized by training epoch count across all tested sequence lengths

Epochs	Avg. RMSE	Avg. STD	Best sequence	Best RMSE	Min	Max
100	5.1316	2.6440	3 values	2.1671	0.000294	8.6458
200	3.3952	1.8555	6 values	1.7670	0.001701	6.5176
300	1.8606	1.1373	12 values	1.6079	0.000526	5.4800
400	3.0473	1.6374	12 values	1.4590	0.000652	5.7906
500	1.6561	1.0042	12 values	1.4297	0.002270	5.8471
600	2.4820	1.4390	12 values	1.3288	0.000404	5.5024
700	1.9000	1.4373	6 values	1.1975	0.002228	5.6961
800	1.4389	0.9425	12 values	1.1948	0.000328	6.2877
900	1.3491	0.9319	12 values	1.1472	0.000340	6.5566
1000	1.3806	0.9492	12 values	1.1738	0.000031	5.5394
1100	1.3489	0.9537	12 values	1.1219	0.002251	5.8369
1200	1.2838	0.9144	6 values	1.0956	0.000652	5.5986
1300	1.2626	0.9116	6 values	1.1004	0.001472	5.8616
1400	1.2634	0.9035	6 values	1.0716	0.001736	5.7865
1500	1.2295	0.8958	6 values	1.0847	0.001060	5.7955
1600	1.2364	0.9202	6 values	1.0673	0.001553	5.8371
1700	1.2296	0.9067	12 values	1.0623	0.000027	7.0745
1800	1.2305	0.9113	12 values	1.0440	0.000061	5.1744
1900	1.4931	1.2226	12 values	1.1033	0.002674	5.3809
2000	1.2416	0.9292	6 values	1.0869	0.000378	6.3559

These results indicate that virtual sensor accuracy depends strongly on both sequence length and training duration. Shorter sequences, especially 3 values, provide more stable behavior over a wide range of epoch counts, which suggests that moderate temporal context is easier to learn and less sensitive to incomplete convergence. Longer sequences, especially 12 values, offer the lowest single RMSE once sufficient training epochs are used. However, after about 1200 epochs, the comparison becomes less about large absolute RMSE differences and more about robustness, because the mature configurations already lie in a very narrow error band. The 24-value sequence did not provide a sufficiently strong practical advantage to justify its instability, which suggests that extending the input horizon to 72 h introduces redundant or more difficult-to-optimize information.

The experimental investigation shows that the optimal temporal context for the analyzed three-layer BiLSTM virtual sensor cannot be identified by RMSE alone. If the objective is purely the lowest single observed RMSE, the preferred setting is 12 values and 1800 epochs. If the objective is robust performance over a realistic training schedule, 3 values remain the best average choice, while 6 values is the most balanced practical option because its best-case RMSE is only 0.0233 °C above the global minimum and its average RMSE is only 0.0457 °C above the most stable sequence. This interpretation is more meaningful for the later fault-detection application, where residual stability matters at least as much as the smallest isolated temperature error.

In the broader context of the dissertation, the main result of this experiment is, therefore, not that one sequence length is overwhelmingly better in an absolute sense. The more important conclusion is that medium-length temporal contexts are sufficient for accurate virtual sensing, while very short contexts underuse historical information and very long contexts reduce robustness. Since the final diagnostic objective is based on residual behavior rather than on direct temperature thresholding, a stable configuration is more valuable than a marginally smaller isolated RMSE value.

3.1.4. Effect of Recurrent Model Structure

This experiment investigated how the recurrent model structure influences the prediction accuracy of the virtual sensor. The objective was to determine whether the virtual sensor can be improved by increasing the number of BiLSTM layers, increasing the number of recurrent cells per layer, and adjusting the dropout ratio. To isolate the effect of structural complexity, the experiment was carried out using only BiLSTM-based architectures, while the input representation and the remaining experimental settings were kept unchanged.

Three structural factors were varied. The first factor was the number of recurrent layers, with one-, two-, three-, and four-layer BiLSTM architectures consid-

ered. The second factor was the number of recurrent cells per layer, with 16, 32, 48, 64, 80, 96, 112, and 128 cells tested. The third factor was the dropout ratio, with values from 0.2 to 0.8 in steps of 0.1. Altogether, this produced a full factorial set of 224 model configurations. The comparison was performed using RMSE as the main evaluation criterion.

The best overall result of the entire experiment was obtained with a three-layer BiLSTM architecture, 80 recurrent cells per layer, and dropout equal to 0.2, for which RMSE reached 2.2127. Very similar results were also obtained with a four-layer BiLSTM using 96 cells and dropout 0.2 (RMSE = 2.2171), and with a three-layer BiLSTM using 128 cells and dropout 0.4 (RMSE = 2.2206). The best-performing configurations were concentrated around the deeper architectures with medium-to-large hidden-state size and relatively low dropout.

To analyze the effect of model depth, Table 3.9 summarizes the average RMSE obtained for each number of layers across all tested dropout values and hidden-unit counts, together with the best individual configuration found for that depth. The single-layer BiLSTM architecture was clearly the weakest option, with an average RMSE equal to 9.7825. The average error decreased substantially when more layers were introduced, dropping to 5.4421 for two layers and further to 5.0552 for three layers. The four-layer architecture did not improve the average result further and reached 5.5351, which was slightly worse than the two-layer case. Taken together, these values indicate that increasing depth from one to three layers was beneficial, but a further increase to four layers did not provide consistent improvement.

Table 3.9. Effect of the number of BiLSTM layers on RMSE

Layers	Average RMSE	Best RMSE	Best cells	Best dropout
1	9.7825	5.1109	128	0.5
2	5.4421	2.5300	128	0.4
3	5.0552	2.2127	80	0.2
4	5.5351	2.2171	96	0.2

The effect of the number of recurrent cells per layer is summarized in Table 3.10. The smallest hidden sizes were not suitable for the considered task. The average RMSE was 17.9246 when only 16 cells were used and 8.8130 with 32 cells. A substantial improvement appeared when the hidden size increased to 48 and 64 cells, with average RMSE values dropping to 6.1190 and 4.3537, respectively. The best average performance was achieved with 112 cells, where the mean RMSE over all tested layer counts and dropout values reached 3.3734. The neighboring hidden sizes of 96 and 128 cells also performed well, with average RMSE values of 3.5652 and 3.5183, respectively. This indicates that increasing the number of cells

was beneficial up to a medium-to-large hidden size, after which the gain became marginal and the performance entered a plateau region.

Table 3.10. Effect of the number of recurrent cells per layer on RMSE

Cells	Average RMSE	Best RMSE	Best layers	Best dropout
16	17.9246	14.7925	2	0.6
32	8.8130	6.5603	2	0.3
48	6.1190	3.1665	3	0.2
64	4.3537	2.3105	3	0.2
80	3.9627	2.2127	3	0.2
96	3.5652	2.2171	4	0.2
112	3.3734	2.2362	3	0.2
128	3.5183	2.2206	3	0.4

The effect of dropout is presented in Table 3.11. The lowest average RMSE across all tested architectures and hidden sizes was obtained with dropout equal to 0.2, for which the mean RMSE was 6.2313. A very similar average result was achieved with dropout 0.4, where the mean RMSE was 6.2615. As the dropout ratio increased further, the average RMSE gradually worsened, reaching 6.6630 at dropout 0.8. This suggests that the analyzed BiLSTM structures required only mild regularization, whereas stronger dropout tended to suppress useful temporal information together with overfitting effects. It is also notable that, for every dropout level from 0.2 to 0.8, the best-performing configuration always belonged to the three-layer BiLSTM family.

Table 3.11. Effect of dropout on RMSE

Dropout	Average RMSE	Best RMSE	Best layers	Best cells
0.2	6.2313	2.2127	3	80
0.3	6.6604	2.3645	3	80
0.4	6.2615	2.2206	3	128
0.5	6.3701	2.2338	3	128
0.6	6.5136	2.2450	3	128
0.7	6.4763	2.3727	3	96
0.8	6.6630	2.4465	3	128

For completeness, Table 3.12 lists the five best model configurations found in the whole experiment. The table confirms that the strongest results were concentrated in the deeper architectures, especially the three-layer BiLSTM models, and that low dropout values were preferred. The difference between the best and second-best configuration was very small, only about 0.0043 RMSE units, which suggests that several structurally similar configurations were nearly equivalent in performance.

Table 3.12. Top-performing BiLSTM model configurations according to RMSE

Layers	Cells	Dropout	RMSE
3	80	0.2	2.2127
4	96	0.2	2.2171
3	128	0.4	2.2206
4	80	0.2	2.2282
3	128	0.5	2.2338

Taken together, i.e., model depth, hidden-state size, and dropout, influence virtual-sensor prediction accuracy, however, unequally. The transition from one layer to three layers produced the largest systematic improvement, which indicates that the gearbox-temperature prediction task benefits from a deeper temporal representation. Increasing the number of recurrent cells per layer was also beneficial, but only up to a certain point; after approximately 96–112 cells, the average gain became small. Increasing dropout, by contrast, did not improve performance and, on average, slightly worsened it beyond the lowest tested value. The experiment, therefore, suggests that the most suitable BiLSTM-based virtual sensor in this investigation should combine moderate-to-high hidden-state capacity with relatively low regularization and a depth of about three recurrent layers.

The best RMSE of the structural comparison was achieved by the three-layer BiLSTM architecture with 80 recurrent cells in each layer and dropout equal to 0.2. This configuration was therefore selected as the preferred model structure for the subsequent experimental investigation stages.

3.1.5. Experimental Investigation of Alternative Activation Functions

This experiment investigated whether replacing the standard exponential-based activation functions in the BiLSTM-based virtual sensor with bounded alternatives without exponentials causes a noticeable degradation in the accuracy of gearbox bearing temperature prediction. In this part of the experimental investigation, the recurrent model structure, dataset split, and all remaining training settings were kept unchanged, while only the gate and state activation functions were varied. The standard *logistic sigmoid + hyperbolic tangent* configuration was used as the baseline reference. The comparison included four alternative gate activations, namely *hard sigmoid*, *softsign gate*, *arctangent gate*, and *algebraic gate*, and four alternative state activations, namely *softsign*, *hard hyperbolic tangent*, *arctangent state*, and *algebraic state*. In total, sixteen alternative gate–state pairs were evaluated, together with the baseline configuration.

The main objective of the experiment was to compare the resulting sensor-value prediction accuracy using standard regression metrics, namely test RMSE,

test MAE, coefficient of determination R^2, residual standard deviation, and residual bias. Since this part of the dissertation focuses on prediction quality, the activation-cost calculations are not considered here.

Table 3.13 presents the test RMSE values for all evaluated alternative gate–state combinations. The best-performing region of the matrix is concentrated around the *softsign* state activation. In particular, the three best alternative configurations were *hard sigmoid + softsign*, *arctangent gate + softsign*, and *softsign gate + softsign*, with test RMSE values of 3.6022, 3.6969, and 3.7649, respectively. All three combinations outperformed the standard baseline, whose test RMSE was 3.8465. The weakest results were obtained when the state activation was a *hard hyperbolic tangent* or when the gate activation was an *algebraic gate*. The worst configuration in the whole experiment was *algebraic gate + hard hyperbolic tangent*, for which test RMSE increased to 4.6497.

Table 3.13. Test RMSE values for the investigated gate–state activation pairs in the BiLSTM-based virtual sensor model

Gate activation	Algebraic state	Arctangent state	Hard hyperbolic tangent	Softsign
Algebraic gate	4.2089	4.4129	4.6497	4.0524
Arctangent gate	4.1014	4.1375	4.4724	3.6969
Hard sigmoid	4.1510	3.9971	4.2615	3.6022
Softsign gate	4.1441	4.2545	4.3015	3.7649

To complement the RMSE matrix, Table 3.14 summarizes the prediction-quality metrics of the baseline and the best-performing alternative configurations. The best overall result was achieved by the *hard sigmoid + softsign* pair, which reduced test RMSE from 3.8465 to 3.6022 and test MAE from 2.9749 to 2.8797, while increasing R^2 from 0.5676 to 0.6208. The residual standard deviation fell from 3.1900 to 3.0338, and the absolute residual bias decreased from -2.1493 to -1.9422. In other words, the best non-exponential alternative did not degrade prediction accuracy; it improved all of the main evaluation metrics relative to the standard baseline.

Table 3.14. Prediction-quality metrics of the baseline and best alternative activation-function configurations

Configuration	Test RMSE	Test MAE	R^2	Residual STD	Bias
Sigmoid + tanh (baseline)	3.8465	2.9749	0.5676	3.1900	-2.1493
Hard sigmoid + softsign	3.6022	2.8797	0.6208	3.0338	-1.9422
Arctangent gate + softsign	3.6969	2.9966	0.6006	3.0366	-2.1086
Softsign gate + softsign	3.7649	2.9563	0.5858	3.2095	-1.9681

A more general pattern can be observed by averaging the results across all configurations that share the same gate or state activation. Among the alternative

gate activations, the best average performance was obtained with the *hard sigmoid*, for which the mean test RMSE over all tested state activations was 4.0030. This was followed by an *arctangent gate* with average RMSE = 4.1021, a *softsign gate* with 4.1163, and an *algebraic gate* with 4.3310. From the gate-function perspective, a *hard sigmoid* was the most suitable replacement of the standard sigmoid in this experiment.

The effect of the state activation was even more pronounced. The average test RMSE of all configurations using *softsign* as the state activation was 3.7791, which was not only the best among the alternative states, but was also slightly better than the baseline *tanh* result of 3.8465. For the remaining state activations, the average RMSE increased to 4.1513 for *algebraic state*, 4.2005 for *arctangent state*, and 4.4213 for *hard hyperbolic tangent*. This indicates that the choice of state activation had a stronger influence on prediction quality than the choice of gate activation, and that *softsign* was the most suitable non-exponential state activation among the tested alternatives.

From the accuracy perspective, the most important result of this experiment is that the use of alternative activation functions without exponentials did not necessarily cause a noticeable degradation in the prediction of gearbox bearing temperature. On the contrary, the best alternative configuration, *hard sigmoid* + *softsign*, improved test RMSE by approximately 6.35% relative to the standard *sigmoid* + *tanh* baseline, while also improving R^2 and reducing both MAE and residual dispersion. At the same time, the experiment also showed that not all non-exponential alternatives are equally suitable: some combinations, especially those involving *hard hyperbolic tangent* or *algebraic gate*, degraded prediction quality noticeably.

The results indicate that the replacement of exponential-based activation functions can be performed without sacrificing virtual-sensor accuracy, provided that the gate and state activation pair is selected appropriately. Among the investigated alternatives, the combination of *hard sigmoid* for recurrent gates and *softsign* for the state update provided the best balance from the prediction-accuracy perspective and was therefore identified as the most promising activation-function alternative for the subsequent discussion.

3.1.6. Residual Analysis and Early Fault Detection Performance

After the virtual sensor had been trained to reproduce the gearbox bearing temperature under healthy operating conditions, its residual signal was analyzed from the early fault detection perspective. The objective of this experiment was not to classify fault categories directly, but to determine whether persistent growth of the residual could provide an early warning before the registered component failure. Under this logic, the virtual sensor was treated as a reference model of expected

healthy behavior, while the residual between the measured and predicted sensor values was treated as the diagnostic signal.

The fault-related evaluation was performed on failure-labeled periods of the public EDP wind turbine dataset. The modeling assumption was that the gearbox bearing temperature can be predicted from ambient and internal SCADA signals during healthy operation, whereas increasing disagreement between the real and virtual sensor values indicates that the monitored component is no longer behaving consistently with the learned healthy-state relationship. In representative cases, the predicted temperature trajectory followed the normal operating trend closely, while the largest deviations appeared near abnormal operating periods. This confirms that the residual is suitable not only for pointwise error evaluation, but also for tracking the temporal development of abnormal behavior.

To convert the residual into an early warning signal, two threshold estimation techniques were compared: a fixed Gaussian-type threshold based on $\mu + 3\sigma$ of the residuals and an adaptive threshold based on the moving median of the residual signal. The early warning performance was evaluated in terms of lead time, defined as the number of days between the first persistent threshold exceedance and the registered failure event. Table 3.15 summarizes the obtained lead times for the analyzed failure groups.

Table 3.15. Estimated lead time before failure based on residual anomalies for two stacked recurrent virtual-sensor models and two threshold estimation methods

Component failure	**LSTM–LSTM**		**LSTM–BiLSTM**	
	Threshold estimation		**Threshold estimation**	
	$\mu + 3\sigma$	**Moving median**	$\mu + 3\sigma$	**Moving median**
Gearbox	14	22	22	32
Generator	15	24	26	37
Generator Bearing	13	24	22	36
Transformer	23	22	24	35
Hydraulic Group	12	22	20	33

The adaptive threshold had a consistent advantage. For the two-layer LSTM model, the moving-median threshold increased the warning horizon from 14–23 days to 22–24 days, except for the transformer case, where the lead time remained nearly unchanged. The average lead time of this model increased from 15.4 days with the fixed threshold to 22.8 days with the adaptive threshold. Even for the stacked LSTM virtual sensor, the moving-median approach produced earlier and more stable warning signals than the fixed $\mu + 3\sigma$ rule.

An even stronger effect was observed for the two-layer BiLSTM model. With the fixed threshold, the lead time ranged from 20 to 26 days, depending on the failure group, with an average of 22.8 days. When the moving-median threshold was applied, the lead time increased to 32–37 days, with an average of 34.6 days.

For the BiLSTM model, the adaptive threshold extended the warning horizon by 10–14 days for all analyzed component groups and improved the average lead time by 11.8 days relative to the fixed-threshold case.

The longest warning horizon of the whole experiment was obtained for generator failures using the two-layer BiLSTM model and the moving-median threshold, where the first persistent residual anomaly appeared 37 days before the registered fault event. Very similar results were obtained for generator bearing failures (36 days), transformer failures (35 days), hydraulic group failures (33 days), and gearbox failures (32 days). These values indicate that the proposed residual-based monitoring approach can provide a practically meaningful early warning interval across several component groups, even though the virtual sensor itself was trained only to predict the expected sensor value and not to classify faults directly.

Overall, the residual analysis confirms that accurate virtual sensing can be transformed into a useful early fault detection mechanism when an appropriate thresholding strategy is used. The residual of the virtual sensor carries diagnostically relevant information before the registered failure events, and adaptive thresholding, especially when combined with a stacked BiLSTM virtual sensor, provides the longest and most consistent early warning intervals. This supports the main idea of the dissertation that early fault detection in wind turbine components can be effectively formulated as the detection of sustained deviations from a learned model of healthy sensor behavior.

3.2. Experimental Investigation of Short-Term Wind Farm Power Forecasting

This section presents the experimental investigation of short-term wind farm power forecasting. The objective was to evaluate how the choice of numerical weather prediction input affects day-ahead forecasting accuracy when a BiLSTM-based predictor is used. Besides comparing weather forecast sources, this part of the experimental investigation also included feature selection and hyperparameter tuning, so that the final NWP comparison was carried out with one fixed forecasting configuration.

3.2.1. Forecasting Dataset, Numerical Weather Prediction Inputs, and Experimental Setup

The forecasting dataset was built from two aligned data sources. The first contained hourly numerical weather prediction forecasts retrieved through the Open-Meteo API. The second contained hourly power measurements from a Lithuanian

Table 3.16. Wind power forecasting dataset used in the BiLSTM experiments

Attribute	Description
Wind farm	6 turbines, 2.75 MW each (Lithuania)
Power data resolution	Hourly power measurements
Meteorological source	Hourly Open-Meteo API forecasts (multiple NWP models)
Training period	May 2023 – October 2023
Test period	November 2023
Candidate inputs	Wind speed (10 m, 80 m), wind gust (80 m), wind direction, temperature, humidity, sea level pressure

wind farm with six turbines rated at 2.75 MW each. The records were matched by timestamp and location, and only the observations present in both sources were kept for supervised learning.

Meteorological forecasts were taken from several NWP sources, namely ICON, GEM Global, Meteo France, GFS Global, and Best Match. The candidate inputs included wind speed at 10 m and 80 m, wind gust at 80 m, wind direction, temperature, humidity, and sea level pressure. Recent power production history was added alongside the weather variables. Before training, all inputs were normalized so that variables with larger numerical values would not dominate the optimization. The data were then split chronologically to avoid information leakage and to keep the forecasting task close to a real operating scenario.

The resulting experimental dataset is summarized in Table 3.16. The forecasting setup combines recent production history and meteorological forecasts within one supervised learning problem. This matters because the differences observed later between the compared NWP sources can then be interpreted as effects of input quality under an otherwise fixed forecasting pipeline.

3.2.2. Selection and Comparison of Numerical Weather Prediction Sources

After the forecasting pipeline had been fixed, the main comparison focused on the influence of the selected numerical weather prediction source. The same optimized BiLSTM forecasting model and the same selected input representation were reused for all compared NWP inputs. In this way, the observed differences in forecasting performance could be attributed primarily to the quality and suitability of the weather forecast source rather than to changes in the neural-network architecture or training procedure.

The comparison included five NWP sources: ICON, GEM Global, Meteo France, GFS Global, and Best Match. The results of this comparison are summarized in Table 3.17. The table shows that the choice of NWP source had a clear effect on day-ahead forecasting accuracy. The two best-performing sources were

ICON and Best Match, which achieved RMSE values of 1.7565 and 1.7604, respectively. Their MAE and NMAE values were also the lowest among all tested models, and both yielded the highest coefficients of determination, with $R^2 = 0.85543$ for ICON and $R^2 = 0.85478$ for Best Match.

The weakest results were obtained with GFS Global and GEM Global. GFS Global produced RMSE = 2.0242, MAE = 1.4621, and $R^2 = 0.80801$, while GEM Global produced RMSE = 2.0086, MAE = 1.4447, and $R^2 = 0.81094$. Relative to GFS Global, ICON reduced RMSE by approximately 13.2%, and relative to GEM Global by approximately 12.6%. These differences are sufficiently large to justify treating the NWP source as a core design variable in the forecasting pipeline rather than as an interchangeable external input.

3.2.3. Bidirectional Long Short-Term Memory Forecasting Model and Hyperparameter Optimization

The forecasting model used in this investigation was based on a BiLSTM architecture. The purpose of using BiLSTM was not to introduce a new recurrent network structure, but to identify an effective forecasting configuration and then use it consistently in the comparison of meteorological inputs. The model was selected because it can exploit temporal dependencies in both directions within the input window and is therefore suitable for the considered day-ahead forecasting problem.

Bayesian optimization was used as a validation-driven search procedure for tuning the forecasting configuration. After each trial, the observed validation error was returned to the optimizer, which proposed the next candidate configuration in a more promising region of the hyperparameter space. This made it possible to avoid exhaustive search while still identifying one best-performing configuration before the final NWP comparison was carried out.

The optimization stage covered both architectural and training-related parameters. These included the number of BiLSTM layers, dropout regularization, learning rate, mini-batch size, gradient threshold, L2 regularization, sequence padding, and the final input-window configuration. After optimization, the best-performing feature set included wind speed at 80 m and 10 m, wind gust at 80 m, sea level pressure, and recent power production. These optimized settings were then fixed and reused in the subsequent comparison of alternative NWP sources.

3.2.4. Day-Ahead Forecasting Accuracy Results

The day-ahead forecasting performance was evaluated using RMSE, MAE, R^2, and normalized mean absolute error (NMAE). The comparative results are pre-

Table 3.17. Forecasting accuracy obtained with different numerical weather prediction sources

Model	**RMSE**	**MAE**	R^2	**NMAE**
Best Match	1.7604	1.2580	0.85478	0.21643
ICON	1.7565	1.2549	0.85543	0.21591
GEM Global	2.0086	1.4447	0.81094	0.24857
Meteo France	1.9520	1.3909	0.82146	0.23930
GFS Global	2.0242	1.4621	0.80801	0.25155

sented in Table 3.17. As already indicated, ICON and Best Match formed the strongest pair, while Meteo France remained intermediate, and GEM Global and GFS Global produced the largest errors.

From the accuracy perspective, ICON achieved the lowest RMSE, MAE, and NMAE, and also the highest R^2. Best Match produced nearly identical results, differing only marginally from ICON across all four metrics. Meteo France performed noticeably worse than the two leading models, but still remained clearly superior to GEM Global and GFS Global. This comparison shows that not all weather forecast sources are equally suitable for the studied wind farm, even when the same BiLSTM forecasting architecture and optimized hyperparameters are used.

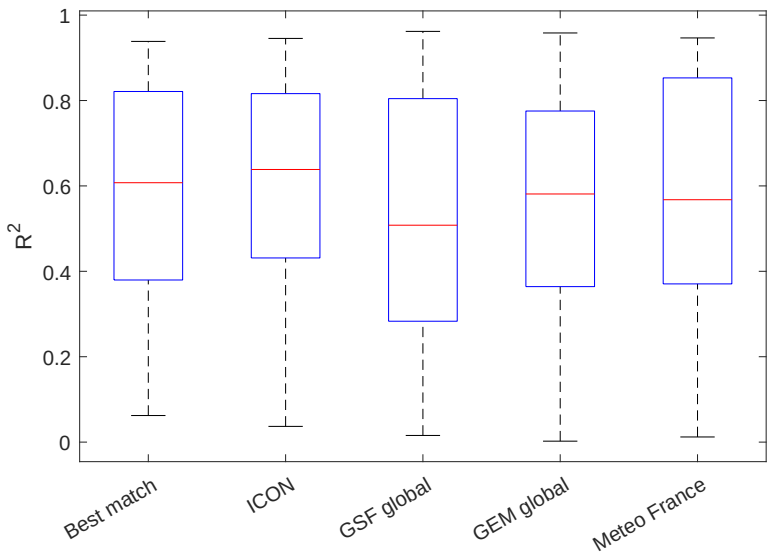


Fig. 3.1. Comparison of daily R^2 for different weather prediction models

The results also show that the forecasting error is sensitive to the regional suitability and effective representativeness of the meteorological source. Since the

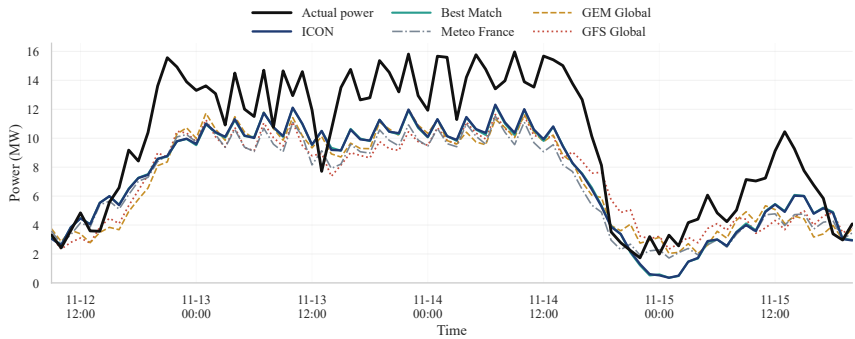


Fig. 3.2. Comparison of actual vs. predicted power for different models using NWP inputs

model architecture and feature selection were kept fixed during the comparison, the observed performance differences can be interpreted primarily as consequences of the NWP input itself. This supports the conclusion that the selection of the meteorological source is one of the main determinants of day-ahead wind power forecasting accuracy in the considered limited-observability setting.

The daily and time-series comparisons shown in Figures 3.1 and 3.2 further confirm the numerical results from Table 3.17. ICON and Best Match maintained more consistent daily R^2 values and followed the measured power trajectory more closely during changing operating periods. In contrast, the weaker NWP inputs produced larger deviations, especially during ramps and near local extrema of the production curve.

The forecasting experiment shows that the BiLSTM-based day-ahead prediction framework can achieve high accuracy when appropriate meteorological inputs are selected, and that the choice of NWP source is one of the main factors determining the final quality of the forecast.

3.2.5. Comparison of Standard and Price-Weighted Training Objectives

This experiment investigated whether a BiLSTM-based wind power forecasting model can be trained not only to minimize the conventional forecast error, but also to improve the economic outcome of day-ahead market participation. In the standard formulation, the model is trained using a usual regression objective that directly penalizes the difference between predicted and actual generated power. In the alternative formulation, the model is trained using an objective function with a normalized Nord Pool price multiplier, in which the cost of grid balancing is incorporated indirectly through the optimization target. Concretely, the Nord

Table 3.18. Statistical forecasting performance of the BiLSTM model trained with the proposed objective function for the *Best Match* NWP source

Dataset	RMSE	MAE	R^2	Residual STD	Min error	Max error
Train	1.9364	1.5089	0.7810	1.2136	-7.4029	9.5176
Validation	1.9572	1.5584	0.7461	1.1840	-3.8464	6.1886
Test	1.9921	1.5715	0.7437	1.2243	-8.5603	10.7233

Pool hourly electricity-price series was merged with the forecasting dataset, shifted by one hour so that each sample was associated with the price of the forecasted delivery hour, and then converted into normalized price-based weights used in the proposed objective function. The purpose of the experiment was, therefore, to determine whether this economically motivated objective function can improve the accumulated market result, and to quantify the corresponding trade-off in statistical forecast accuracy.

The comparison was carried out for the *Best Match* meteorological input, which had already shown competitive forecasting performance in the preceding experiments. The same BiLSTM forecasting framework was used, and the objective function was the main factor varied in this stage of the investigation. Two complementary result groups were analyzed. First, the forecasting quality of the model trained with the proposed objective function was evaluated on the training, validation, and test subsets using standard statistical metrics. Second, the standard and price-weighted models were compared directly on a representative validation week in terms of both forecast fit and accumulated economic result.

Table 3.18 summarizes the statistical forecasting performance of the BiLSTM model trained with the proposed objective function. The model remained reasonably accurate across all data subsets. On the training set, it achieved RMSE = 1.9364, MAE = 1.5089, and $R^2 = 0.7810$. On the validation set, RMSE increased slightly to 1.9572, MAE to 1.5584, and R^2 decreased to 0.7461. On the test set, the final accuracy was RMSE = 1.9921, MAE = 1.5715, and $R^2 = 0.7437$. The residual standard deviation remained relatively stable, varying from 1.1840 on the validation set to 1.2243 on the test set. These values indicate that the price-weighted model preserved a stable level of predictive performance and did not collapse statistically, even though its optimization target was no longer the conventional forecast-error loss.

A direct comparison between the standard and price-weighted training objectives on the validation week is presented in Table 3.19. The standard objective achieved a higher coefficient of determination, $R^2 = 0.846$, while the proposed objective function reached $R^2 = 0.746$. The price-weighted model was therefore statistically less accurate on this validation interval. However, this loss in fit quality was accompanied by a clear improvement in the accumulated economic result. The total validation-week profit increased from EUR 45,405 for the standard ob-

Table 3.19. Validation-week comparison of standard and price-weighted training objectives for the *Best Match* NWP source

Training objective	R^2	**Accumulated profit (EUR)**	**Profit change**
Standard objective	0.846	45405	reference
Proposed objective	0.746	53955	+8550 (+18.8%)

jective to EUR 53,955 for the proposed objective function, that is, by EUR 8,550 or approximately 18.8%.

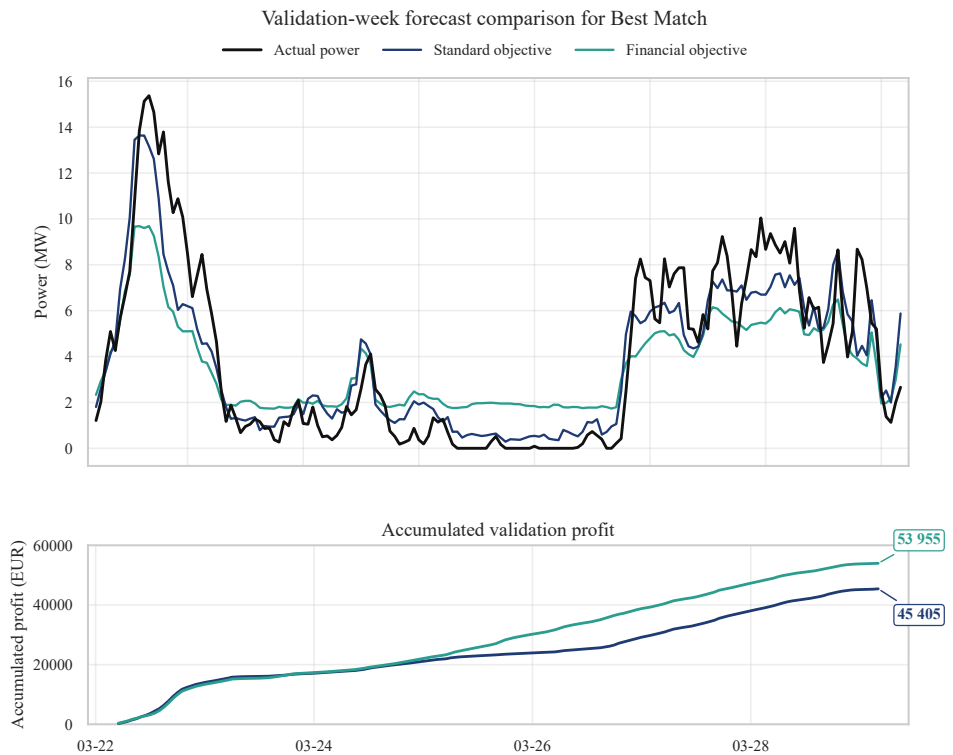


Fig. 3.3. Validation-week comparison of the standard and proposed training objectives for the *Best Match* NWP source (*upper panel*: actual and predicted power trajectories; *lower panel*: accumulated validation profit)

This trade-off is also illustrated in Figure 3.3. In the upper panel, the standard objective follows the measured power curve more closely, which is consistent with its higher R^2. The forecast produced by the proposed objective function is smoother and deviates more strongly from the actual production in some intervals,

but this smoothing appears to be beneficial from the market perspective. The lower panel shows that the accumulated profit of the price-weighted model grows faster throughout the validation week and remains above the standard-objective curve over the whole interval. The figure makes this trade-off explicit: the statistically more accurate model is not necessarily the economically preferable one.

From the forecasting perspective, the most important result of this experiment is that optimization with respect to the proposed price-weighted objective function changes the behavior of the predictor in a meaningful way. The model trained with this objective sacrifices part of the conventional pointwise fit quality, as reflected by the lower R^2 on the validation week, but at the same time, it improves the accumulated economic outcome. In other words, minimizing the usual forecast error is not equivalent to maximizing the market-oriented utility of the forecast. The results, therefore, support the idea that, in practical wind power trading, model assessment should not rely on statistical accuracy measures alone.

Overall, the experiment shows that a BiLSTM-based forecasting model trained with the proposed objective function can improve the economic result of day-ahead forecasting, even when this is accompanied by a moderate loss in statistical fit. This confirms that the economic interpretation of forecast errors is not only theoretical, but also experimentally observable in the optimization outcome.

3.3. Conclusions of the Third Chapter

1. The experimental results demonstrate that temporal aggregation has a substantial influence on virtual sensor accuracy, and among the tested alternatives, the 3 h aggregation interval was consistently the most effective, yielding the lowest average RMSE for all four analyzed recurrent neural-network structures and improving the overall mean RMSE to 3.5685, compared with 3.9551 for 24 h and 4.5380 for 6 h.
2. The experiment shows that a short training history is not sufficient for robust virtual-sensor construction, since for all analyzed recurrent architectures the 3 months for training/3 months for testing setup outperformed the 1 month for training/5 months for testing setup when training was performed on turbine T01 only; for the best-performing LSTM–LSTM architecture, the average RMSE increased from 2.5436 to 3.9450 when the training history was reduced from 3 months to 1 month and the test interval was extended to 5 months, which corresponds to a deterioration of approximately 35.5%.
3. The obtained results further showed that shorter sequences, especially 3 aggregated input vectors, provide the most stable performance across a

broad range of training epochs, with the lowest average RMSE of 1.3772 and average residual standard deviation of 0.9226, while 6 vectors provided the most balanced compromise (average RMSE = 1.4229, best RMSE = 1.0673); after 1200 epochs, the strongest sequence-length configurations differed by less than 0.06 °C in RMSE, so configuration choice must also consider stability rather than minimum RMSE alone.

4. The structural experiment showed that increasing the depth of the BiLSTM model substantially improved virtual-sensor accuracy up to three recurrent layers, since the average RMSE decreased from 9.7825 for the single-layer architecture to 5.4421 for two layers and further to 5.0552 for three layers, while the best overall result of the whole experiment was achieved by the three-layer BiLSTM with 80 recurrent cells and dropout 0.2, yielding RMSE = 2.2127.
5. The experimental results show that replacing the standard exponential-based *sigmoid* + *tanh* activation pair in the BiLSTM-based virtual sensor with appropriately selected non-exponential bounded alternatives does not degrade gearbox bearing temperature prediction accuracy and may even improve it, as the best alternative configuration, *hard sigmoid* + *softsign*, reduced test RMSE from 3.8465 to 3.6022 and increased R^2 from 0.5676 to 0.6208.
6. The residual analysis showed that the virtual-sensor-based approach can provide practically meaningful early warnings before registered component failures, and the comparison of thresholding methods demonstrated that adaptive residual thresholding is more effective than the fixed $\mu + 3\sigma$ rule: for the two-layer BiLSTM model with a moving-median threshold, the average lead time increased from 22.8 to 34.6 days, provided a consistent improvement across all analyzed failure groups, and reached the longest warning horizon of 37 days before generator failure.
7. The experimental investigation showed that the selection of the numerical weather prediction source is a key determinant of day-ahead wind farm power forecasting accuracy, since the BiLSTM model achieved the best overall performance with the ICON forecast input (RMSE=1.7565, MAE=1.2549, R^2 = 0.85543, NMAE=0.21591), while the weakest results were obtained with GFS Global and GEM Global.
8. Training the BiLSTM forecasting model with the proposed objective function that uses a normalized Nord Pool price multiplier improved the practical economic outcome of day-ahead wind power forecasting: the accumulated validation-week profit increased from EUR 45,405 to EUR 53,955, that is, by EUR 8,550 (18.8%), even though the statistical fit decreased

from R^2 = 0.846 to R^2 = 0.746, which confirms that the economically optimal forecast is not necessarily the one with the smallest conventional prediction error.

General Conclusions

1. The developed virtual-sensor-based method is suitable for wind-turbine condition monitoring and early fault detection from SCADA time-series data, because the healthy-operation variability of the monitored , listed as gearbox bearing temperature (20.00–71.50 °C), is substantially larger than the early pre-failure median shift (0.22–0.89 °C), making direct thresholding of the measured signal insufficiently reliable. Using selected informative SCADA inputs together with residual analysis and an adaptive moving-median threshold increased the average warning lead time from 22.8 to 34.6 days and extended the maximum warning horizon to 37 days before generator failure.
2. Optimization of recurrent virtual-sensor models showed that prediction accuracy and practical deployability depend on temporal-input representation, training scheme, architecture, and activation functions. The most suitable setup used 3 h aggregation and longer healthy-state training histories, while the best architecture was a three-layer BiLSTM with 80 cells per layer and dropout 0.2 (RMSE = 2.2127); replacing the standard *sigmoid* + *tanh* pair with *hard sigmoid* + *softsign* reduced RMSE from 3.8465 to 3.6022, increased R^2 from 0.5676 to 0.6208, and reduced nonlinear computation cost.
3. The developed BiLSTM-based short-term wind-farm power-forecasting method showed that both the meteorological input source and the training

objective materially affect practical forecast quality. Among the investigated NWP sources, ICON yielded the best statistical accuracy (RMSE = 1.7565, MAE = 1.2549, $R^2 = 0.85543$, NMAE = 0.21591), whereas the proposed objective function with a normalized Nord Pool price multiplier increased the validation-week profit from EUR 45,405 to EUR 53,955, i.e., by EUR 8,550 (18.8%), confirming that the economically preferable forecast does not necessarily coincide with the minimum conventional prediction error.

Taken together, the dissertation contributes to the electronics direction not only through AI model selection but also through the adaptation of recurrent models for resource-constrained embedded monitoring scenarios.

References

Aggarwal, R., & Kumar, R. (2013). A comprehensive review of numerical weather prediction models, *International Journal of Computer Applications*, 74, 44–48. https://doi.org/10.5120/12989-0246

Ali, M. H. E., Abdel-Raman, A. B., & Badry, E. A. (2022). Developing novel activation functions based deep learning lstm for classification, *IEEE Access*, 10, 97 259–97 275. https://doi.org/10.1109/ACCESS.2022.3205774

Azzam, B., Schelenz, R., Roscher, B., Baseer, A., & Jacobs, G. (2021). Development of a wind turbine gearbox virtual load sensor using multibody simulation and artificial neural networks, *Forschung im Ingenieurwesen*, 85(2), 241–250. https://doi.org/10.1007/s10010-021-00460-3

Banbury, C. R., Reddi, V. J., Lam, M., et al.(2021). Micronets: Neural network architectures for deploying tinyml models on microcontrollers, in *Proceedings of Machine Learning and Systems*, vol. 3, 517–532.

Becker, E., & Poste, P. (2006). Keeping the blades turning: Condition monitoring of wind turbine gears, *Refocus*, 7(2), 26–32. https://doi.org/10.1016/S1471-0846(06)70544-3

Berscheid, D. (2021). *Data-centric Machine Learning: Making customized ML solutions production-ready*. https://dida.do/blog/data-centric-machine-learning. Blog post.

Bharatheedasan, K., Maity, T., Kumaraswamidhas, L., & Durairaj, M. (2023). An intelligent of fault diagnosis and predicting remaining useful life of rolling bearings based on convolutional neural network with bidirectional lstm, *Sādhanā*, 48. https://doi.org/10.1007/s12046-023-02169-1

Bouallègue, Z. B., Clare, M., Magnusson, L., Gascon, E., Maier-Gerber, M., Janousek, M., Rodwell, M., Pinault, F., Dramsch, J., Lang, S., Raoult, B., Rabier, F., Chevallier, M., Sandu, I., Düben, P., Chantry, M., & Pappenberger, F. (2023). The rise of data-driven weather forecasting, .

Campoverde-Vilela, L., Feijóo, M., Vidal, Y., Sampietro, J., & Tutivén, C. (2023). Anomaly-based fault detection in wind turbine main bearings, *Wind Energy Science*, 8, 557–574. https://doi.org/10.5194/wes-8-557-2023

Cao, L., Qian, Z., Zareipour, H., Huang, Z., & Zhang, F. (2019). Fault diagnosis of wind turbine gearbox based on deep bi-directional long short-term memory under time-varying non-stationary operating conditions, *IEEE Access*, 7, 155 219–155 228. https://doi.org/10.1109/ACCESS.2019.2947501

Chai, Z. (2023). Bilstm short-term wind power prediction based on attention mechanism, in *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*, IEEE, 1341–1346.

Chandra, M. (2020). Comparative analysis of polynomial and rational approximations of hyperbolic tangent function for vlsi implementation, *arXiv preprint arXiv:2007.11976*, https://doi.org/10.48550/arXiv.2007.11976

Chen, X., Zhang, X., Dong, M., Huang, L., Guo, Y., & He, S. (2021). Deep learning-based prediction of wind power for multi-turbines in a wind farm, *Frontiers in Energy Research*, 9, 723 775.

Chong, Y. S., Goh, W. L., Ong, Y. S., Nambiar, V. P., & Do, A. T. (2021). Efficient implementation of activation functions for lstm accelerators, in *2021 IFIP/IEEE 29th International Conference on Very Large Scale Integration (VLSI-SoC)*, IEEE, 1–5. https://doi.org/10.1109/VLSI-SoC53125.2021.9606971

Choudhary, A., Jain, P., & Prajesh, A. (2023). Wind power forecasting using deep learning method: A review, in *2023 1st International Conference on Intelligent Computing and Research Trends (ICRT)*, IEEE, 1–6.

de Azevedo, H. D. M., Araújo, A. M., & Bouchonneau, N. (2016). A review of wind turbine bearing condition monitoring: State of the art and challenges, *Renewable and Sustainable Energy Reviews*, 56, 368–379. https://doi.org/10.1016/j.rser.2015.11.032

Demolli, H., Dokuz, A. S., Ecemis, A., & Gokcek, M. (2019). Wind power forecasting based on daily wind speed data using machine learning algorithms, *Energy Conversion and Management*, 198. https://doi.org/10.1016/j.enconman.2019.111823

Dimd, B. D., Voller, S., Cali, U., & Midtgard, O. M. (2022). A review of machine learning-based photovoltaic output power forecasting: Nordic context, *IEEE Access*, 10, 26 404–26 425. https://doi.org/10.1109/ACCESS.2022.3156942

Ding, Y., Yi, Z., Li, M., Long, J., Lei, S., Guo, Y., Fan, P., Zuo, C., & Wang, Y. (2023). Hi-mvit: A lightweight model for explainable skin disease classification based on modified mobilevit, *Digital Health*, 9, 20552076231207 197. https://doi.org/10.1177/20552076231207197

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in deep learning: A comprehensive survey and benchmark, *Neurocomputing*, 503, 92–108. https://doi.org/10.1016/j.neucom.2022.06.111

El-Fouly, T., El-Saadany, E., & Salama, M. (2008). One day ahead prediction of wind speed and direction, *IEEE Transactions on Energy Conversion*, 23, 191–201. https://doi.org/10.1109/TEC.2007.905069

Elsayed, N., Maida, A. S., & Bayoumi, M. (2019). Effects of different activation functions for unsupervised convolutional lstm spatiotemporal learning, *Advances in Science, Technology and Engineering Systems Journal*, 4(2), 260–269. https://doi.org/10.25046/aj040234

Eriksson, J. (2020). *Machine learning for predictive maintenance on wind turbines: Using SCADA data and the Apache Hadoop ecosystem*. https://www.liu.se

Garan, M., Tidriri, K., & Kovalenko, I. (2022). A data-centric machine learning methodology: Application on predictive maintenance of wind turbines, *Energies*, 15, 826. https://doi.org/10.3390/en15030826

Gecevičius, G., Marčiukaitis, M., & Tamašauskienė, M. (2019). The investigation of factors determining wind power prediction accuracy: case study of western lithuania, *Energetika*, 65(1), 95–102. https://lmaleidykla.lt/ojs/index.php/energetika/article/download/3978/2777

Georgilakis, P. S. (2008). Technical challenges associated with the integration of wind power into power systems, *Renewable and Sustainable Energy Reviews*, 12(3), 852–863. https://doi.org/10.1016/j.rser.2006.10.007

Global Wind Energy Council (GWEC) (2023). *Global Wind Report 2023*. https://gwec.net/globalwindreport2023. Accessed: 2023-12-01.

Global Wind Energy Council (GWEC) Market Intelligence (2025). *Global Wind Market Outlook: Update Q3 2025*. https://marketintelligence.gwec.net/wp-content/uploads/2025/12/GWEC_-Global-Wind-Market-Outlook-_Q3-2025.pdf. Editorial close: 05 Dec 2025.

Gomes, G. S. d. S., Ludermir, T. B., & Lima, L. M. M. R. (2011). Comparison of new activation functions in neural network for forecasting financial time series, *Neural Computing and Applications*, 20(3), 417–439. https://doi.org/10.1007/s00521-010-0407-3

Gonzalez, E., Reder, M., & Melero, J. J. (2016). Scada alarms processing for wind turbine component failure detection, *Journal of Physics: Conference Series*, 753. https://doi.org/10.1088/1742-6596/753/7/072019

González-Sopeña, J., Pakrashi, V., & Ghosh, B. (2021a). An overview of performance evaluation metrics for short-term statistical wind power forecasting, *Renewable and Sustainable Energy Reviews*, 138, 110 515. https://doi.org/10.1016/j.rser.2020.110515

González-Sopeña, J. M., Pakrashi, V., & Ghosh, B. (2021b). An overview of performance evaluation metrics for short-term statistical wind power forecasting, *Renewable and Sustainable Energy Reviews*, 138. https://doi.org/10.1016/j.rser.2020.110515

Gruhl, C., Hannan, A., Huang, Z., Nivarthi, C., & Vogt, S. (2021). The problem with real-world novelty detection - issues in multivariate probabilistic models, in *2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*, Institute of Electrical and Electronics Engineers Inc., 204–209. https://doi.org/10.1109/ACSOS-C52956.2021.00055

Hameed, Z., Hong, Y. S., Cho, Y. M., Ahn, S. H., & Song, C. K. (2009). Condition monitoring and fault detection of wind turbines and related algorithms: A review, *Renewable and Sustainable Energy Reviews*, 13, 1–39. https://doi.org/10.1016/j.rser.2007.05.008

Heinermann, J., & Kramer, O. (2016). Machine learning ensembles for wind power prediction, *Renewable Energy*, https://doi.org/10.1016/j.renene.2015.11.073

Higashiyama, K., Fujimoto, Y., & Hayashi, Y. (2018). Feature extraction of nwp data for wind power forecasting using 3d-convolutional neural networks, *Energy Procedia*, https://doi.org/10.1016/j.egypro.2018.11.043

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for mobilenetv3, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, 1314–1324. https://doi.org/10.1109/ICCV.2019.00140

Iorgulescu, M., & Beloiu, R. (2008). *Vibration and Current Monitoring for Fault's Diagnosis of Induction Motors One Teacher and One Student Working with ProjectX View project Experimental validation of a propulsion system with hydrogen fuel cell for a light vehicle-Demonstrator of Mobility with Hydrogen View project Vibration and Current Monitoring for Fault's Diagnosis of Induction Motors*. https://www.researchgate.net/publication/238660542. ResearchGate record.

Jalayer, M., Orsenigo, C., & Vercellis, C. (2021). Fault detection and diagnosis for rotating machinery: A model based on convolutional lstm, fast fourier and continuous wavelet transforms, *Computers in Industry*, 125. https://doi.org/10.1016/j.compind.2020.103378

Jankevičienė, J. (2024). *Assessment of Wind Energy Resources Potential in Extensive Urban Environments under the Changing Climate*: PhD thesis. Kaunas University of Technology and Lithuanian Energy Institute. https://epubl.ktu.edu/object/elaba%3A205108662/205108662.pdf

Jastrzebska, A., Morales-Hernández, A., Nápoles, G., Salgueiro, Y., & Vanhoof, K. (2021). Measuring wind turbine health using drifting concepts, http://arxiv.org/abs/2112.04933

Joseph, T., & Bindiya, T. (2023). Realization and hardware implementation of gating units for long short-term memory network using hyperbolic sine functions, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(12), 5141–5145. https://doi.org/10.1109/TCAD.2023.3293045

Katinas, V., Marčiukaitis, M., Gecevičius, G., & Markevičius, A. (2017). Statistical analysis of wind characteristics based on weibull methods for estimation of power generation in lithuania, *Renewable Energy*, 113, 190–201. https://doi.org/10.1016/j.renene.2017.05.071

Khan, P., Yeun, C., & Byun, Y. (2023). Fault detection of wind turbines using scada data and genetic algorithm-based ensemble learning, *Engineering Failure Analysis*, 148, 107 209. https://doi.org/10.1016/j.engfailanal.2023.107209

Kisvari, A., Lin, Z., & Liu, X. (2021). Wind power forecasting – a data-driven method along with gated recurrent neural network, *Renewable Energy*, 163, 1895–1909. https://doi.org/10.1016/j.renene.2020.10.119

Kusiak, A., Zheng, H., & Song, Z. (2009). On-line monitoring of power curves, *Renewable Energy*, 34(6), 1487–1493. https://doi.org/10.1016/j.renene.2008.10.022

Lagos, A., Caicedo, J. E., Coria, G., Quete, A. R., Martínez, M., Suvire, G., & Riquelme, J. (2022a). State-of-the-art using bibliometric analysis of wind-speed and -power forecasting methods applied in power systems, *Energies*, 15(18). https://doi.org/10.3390/en15186545

Lagos, A., Caicedo, J. E., Coria, G., Quete, A. R., Martínez, M., Suvire, G., & Riquelme, J. (2022b). State-of-the-art using bibliometric analysis of wind-speed and -power forecasting methods applied in power systems, *Energies*, 15(18). https://doi.org/10.3390/en15186545

Lane, N. D., & Bhattacharya, S. (2016). Deepx: A software accelerator for low-power deep learning inference on mobile devices, in *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*, 1–12.

Li, Y., Wang, R., Li, Y., Zhang, M., & Long, C. (2023). Wind power forecasting considering data privacy protection: A federated deep reinforcement learning approach, *Applied Energy*, 329. https://doi.org/10.1016/j.apenergy.2022.120291

Lin, Z., & Liu, X. (2020a). Wind power forecasting of an offshore wind turbine based on high-frequency scada data and deep learning neural network, *Energy*, 201. https://doi.org/10.1016/j.energy.2020.117693

Lin, Z., & Liu, X. (2020b). Wind power forecasting of an offshore wind turbine based on high-frequency scada data and deep learning neural network, *Energy*, 201, 117 693. https://doi.org/10.1016/j.energy.2020.117693

Liu, J. H., Corbita, N. T., Lee, R. M., & Wang, C. C. (2022). Wind turbine anomaly detection using mahalanobis distance and scada alarm data, *Applied Sciences (Switzerland)*, 12. https://doi.org/10.3390/app12178661

Liu, L., Liu, J., Ye, Y., Liu, H., Chen, K., Li, D., Dong, X., & Sun, M. (2023a). Ultra-short-term wind power forecasting based on deep bayesian model with uncertainty, *Renewable Energy*, 205, 598–607. https://doi.org/10.1016/j.renene.2023.01.038

Liu, X., Du, J., & Ye, Z. S. (2021). A condition monitoring and fault isolation system for wind turbine based on scada data, *IEEE Transactions on Industrial Informatics*, https://doi.org/10.1109/TII.2021.3075239

Liu, Y., He, J., Wang, Y., Liu, Z., He, L., & Wang, Y. (2023b). Short-term wind power prediction based on ceemdan-se and bidirectional lstm neural network with markov chain, *Energies*, 16(14), 5476.

Liu, Y., Wu, Z., & Wang, X. (2020). Research on fault diagnosis of wind turbine based on scada data, *IEEE Access*, 8, 185 557–185 569. https://doi.org/10.1109/ACCESS.2020.3029435

Liu, Y., Zhang, H., Wu, C., Shao, M., Zhou, L., & Fu, W. (2024). A short-term wind speed forecasting framework coupling a maximum information coefficient, complete ensemble empirical mode decomposition with adaptive noise, shared weight gated memory network with improved northern goshawk optimization for numerical weather prediction correction, *Sustainability*, 16(16). https://doi.org/10.3390/su16166782

Liu, Z., Hajiali, M., Torabi, A., et al.(2018). Novel forecasting model based on improved wavelet transform, informative feature selection, and hybrid support vector machine on wind power forecasting, *Journal of Ambient Intelligence and Humanized Computing*, 9, 1919–1931. https://doi.org/10.1007/s12652-018-0886-0

Marti-Puig, P., Blanco, A. M., Cárdenas, J. J., Cusidó, J., & Solé-Casals, J. (2019). Feature selection algorithms for wind turbine failure prediction, *Energies*, 12. https://doi.org/10.3390/en12030453

Miele, E. S., Bonacina, F., & Corsini, A. (2022). Deep anomaly detection in horizontal axis wind turbines using graph convolutional autoencoders for multivariate time series, *Energy and AI*, 8. https://doi.org/10.1016/j.egyai.2022.100145

Mundotiya, P., Mathuria, P., & Tiwari, H. (2022a). Comprehensive review on uncertainty in wind power forecasting-models and challenges, in *2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT)*, IEEE, 1–6.

Mundotiya, P., Mathuria, P., & Tiwari, H. P. (2022b). Comprehensive review on uncertainty in wind power forecasting-models and challenges, in *Proceedings of the 2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT)*. https://doi.org/10.1109/CISCT55310.2022.10046605

Ng, A. (2021). *A Chat with Andrew on MLOps: From Model-centric to Data-centric AI*. https://www.youtube.com/watch?v=06-AZXmwHjo&ab_channel=DeepLearningAI. Video interview.

OpenAI (2026). *Codex (April 2026 version) [Large language model]*. https://openai.com/codex

Pan, J.-S., Hu, P., & Chu, S.-C. (2019). Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power, *Processes*, 7, 845. https://doi.org/10.3390/pr7110845

Parisi, L., Ma, R., RaviChandran, N., & Lanzillotta, M. (2021). hyper-sinh: An accurate and reliable function from shallow to deep learning in tensorflow and keras, *Machine Learning with Applications*, 6, 100 112. https://doi.org/10.1016/j.mlwa.2021.100112

Piotrowski, P., Rutyna, I., Baczyński, D., & Kopyt, M. (2022). Evaluation metrics for wind power forecasts: A comprehensive review and statistical analysis of errors, *Energies*, 15(24). https://doi.org/10.3390/en15249657

Portugal, E. D. (2019). *Wind Turbine Failure Detection*. https://opendata.edp.com/pages/challenges/#description. Accessed: 2022-06-01.

Qian, P., Tian, X., Kanfoud, J., Lee, J. L. Y., & Gan, T.-H. (2019). A novel condition monitoring method of wind turbines based on long short-term memory neural network, *Energies*, 12(18), 3411. https://doi.org/10.3390/en12183411

Rahaman, H., Rubaith Bashar, T. M., Munem, M., Hasib, M. H. H., Mahmud, H., & Alif, A. N. (2020). Bayesian optimization based ann model for short term wind speed forecasting in newfoundland, canada, in *2020 IEEE Electric Power and Energy Conference (EPEC)*, Edmonton, AB, Canada, 1–5. https://doi.org/10.1109/EPEC48502.2020.9320075

Roelofs, C. M., Lutz, M. A., Faulstich, S., & Vogt, S. (2021). Autoencoder-based anomaly root cause analysis for wind turbines, *Energy and AI*, 4. https://doi.org/10.1016/j.egyai.2021.100065

Rybalkin, V., Sudarshan, C., Weis, C., Lappas, J., Wehn, N., & Cheng, L. (2020). Efficient hardware architectures for 1d-and md-lstm networks, *Journal of Signal Processing Systems*, 92(11), 1219–1245. https://doi.org/10.1007/s11265-020-01554-x

Seidel, E., Franzen, J., Strake, M., & Fingscheidt, T. (2021). Y^2-net fcrn for acoustic echo and noise suppression, *arXiv preprint arXiv:2103.17189*, .

Serrato, R., Maru, M. M., & Padovese, L. R. (2007). Effect of lubricant viscosity grade on mechanical vibration of roller bearings, *Tribology International*, 40, 1270–1275. https://doi.org/10.1016/j.triboint.2007.01.025

Sideratos, G., & Hatziargyriou, N. D. (2007). An advanced statistical method for wind power forecasting, *IEEE Transactions on Power Systems*, 22(1), 258–265. https://doi.org/10.1109/TPWRS.2006.889078

Silfa, F., Arnau, J. M., & González, A. (2020). Boosting lstm performance through dynamic precision selection, in *2020 IEEE 27th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, IEEE, 323–333.

Stetco, A., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., Keane, J., & Nenadic, G. (2019). Machine learning methods for wind turbine condition monitoring: A review, *Renewable Energy*, 133, 620–635. https://doi.org/10.1016/j.renene.2018.10.047

Tautz-Weinert, J., & Watson, S. J. (2017a). Using scada data for wind turbine condition monitoring - a review, *IET Renewable Power Generation*, 11, 382–394. https://doi.org/10.1049/iet-rpg.2016.0248

Tautz-Weinert, J., & Watson, S. J. (2017b). Using scada data for wind turbine condition monitoring - a review, *IET Renewable Power Generation*, 11, 382–394. https://doi.org/10.1049/iet-rpg.2016.0248

Tidriri, K., Braydi, A., & Kazmi, H. (2021). Data-driven decision-making methodology for prognostic and health management of wind turbines*, in *2021 Australian and New Zealand Control Conference (ANZCC)*, Institute of Electrical and Electronics Engineers Inc., 104–109. https://doi.org/10.1109/ANZCC53563.2021.9628240

Timmons, N. G., & Rice, A. (2020). Approximating activation functions, *arXiv preprint arXiv:2001.06370*, .

Tsai, W.-C., Hong, C.-M., Tu, C.-S., Lin, W.-M., & Chen, C.-H. (2023). A review of modern wind power generation forecasting technologies, *Sustainability*, 15(14), 10 757.

Turnbull, A., Carroll, J., & McDonald, A. (2021). Combining scada and vibration data into a single anomaly detection model to predict wind turbine component failure, *Wind Energy*, 24, 197–211. 01, https://doi.org/10.1002/we.2567

U.S. Department of Energy (2022). *The Inside of a Wind Turbine*. https://www.energy.gov/eere/inside-wind-turbine. Accessed: 2026-03-18.

Vallés-Pérez, I., Soria-Olivas, E., Martínez-Sober, M., Serrano-López, A. J., Vila-Francés, J., & Gómez-Sanchís, J. (2023). Empirical study of the modulus as activation function in computer vision applications, *Engineering Applications of Artificial Intelligence*, 120, 105 863. https://doi.org/10.1016/j.engappai.2023.105863

Wang, J., Li, X., Li, J., Sun, Q., & Wang, H. (2022a). Ngcu: A new rnn model for time-series data prediction, *Big Data Research*, 27. https://doi.org/10.1016/j.bdr.2021.100296

Wang, L., Dernoncourt, F., & Bui, T. (2020). Bayesian optimization for selecting efficient machine learning models, .

Wang, X., Zheng, Z., Jiang, G., He, Q., & Xie, P. (2022b). Detecting wind turbine blade icing with a multiscale long short-term memory network, *Energies*, 15. https://doi.org/10.3390/en15082864

Wang, Y., Zou, R., Liu, F., Zhang, L., & Liu, Q. (2021). A review of wind speed and wind power forecasting with deep neural networks, *Applied Energy*, 304. https://doi.org/10.1016/j.apenergy.2021.117766

Wang, Z., Wang, X., & Liu, W. (2023). Genetic least square estimation approach to wind power curve modelling and wind power prediction, *Scientific Reports*, 13(1), 9188.

Wen, W., Liu, Y., Sun, R., & Liu, Y. (2022). Research on anomaly detection of wind farm scada wind speed data, *Energies*, 15. https://doi.org/10.3390/en15165869

Wuraola, A., & Patel, N. D. (2022). Resource efficient activation functions for neural network accelerators, *Neurocomputing*, 482, 163–185. https://doi.org/10.1016/j.neucom.2021.11.032

Yan, G., Yu, C., & Bai, Y. (2021). Wind turbine bearing temperature forecasting using a new data-driven ensemble approach, *Machines*, 9(11), 248. https://doi.org/10.3390/machines9110248

Yang, Y., Gohari, P., & Topcu, U. (2023). On the privacy risks of deploying recurrent neural networks in machine learning models, *Proceedings on Privacy Enhancing Technologies*, 2023(1), 68–84. https://doi.org/10.56553/popets-2023-0005

Ye, H., Yang, B., Han, Y., Li, Q., Deng, J., & Tian, S. (2022). Wind speed and power prediction approaches: Classifications, methodologies, and comments, *Frontiers in Energy Research*, 10, 901 767.

Yin, H., Ou, Z., Huang, S., & Meng, A. (2019). A cascaded deep learning wind power prediction approach based on a two-layer of mode decomposition, *Energy*, https://doi.org/10.1016/j.energy.2019.116316

Ying, H., Deng, C., Xu, Z., Huang, H., Deng, W., & Yang, Q. (2023). Short-term prediction of wind power based on phase space reconstruction and bilstm, *Energy Reports*, 9, 474–482. https://doi.org/10.1016/j.egyr.2023.04.288

Zhang, C., Hu, D., & Yang, T. (2022). Anomaly detection and diagnosis for wind turbines using long short-term memory-based stacked denoising autoencoders and xgboost, *Reliability Engineering & System Safety*, 222, 108 445. https://doi.org/10.1016/J.RESS.2022.108445

Zhang, L., Li, J., & Walter, N. G. (2025). Pretrained deep neural network kin-sim for single-molecule fret trace idealization, *The Journal of Physical Chemistry B*, 129(4), 1167–1175. https://doi.org/10.1021/acs.jpcb.4c05276

Zhang, P., Li, C., Peng, C., & Tian, J. (2020). Ultra-short-term prediction of wind power based on error following forget gate-based long short-term memory, *Energies*, 13(20). https://doi.org/10.3390/en13205400

Zheng, M., Man, J., Wang, D., Chen, Y., Li, Q., & Liu, Y. (2023). Semi-supervised multivariate time series anomaly detection for wind turbines using generator scada data, *Reliability Engineering & System Safety*, 109235. https://doi.org/10.1016/j.ress.2023.109235

List of Scientific Publications by the Author on the Topic of the Dissertation

Papers in the Reviewed Scientific Journals

Jankauskas, M., Serackis, A., Šapurov, M., Pomarnacki, R., Baskys, A., Hyunh, V. K., Vaimann, T., & Zakis, J. (2023). Exploring the Limits of Early Predictive Maintenance in Wind Turbines Applying an Anomaly Detection Technique. *Sensors*, 23(12), 5695. https://doi.org/10.3390/s23125695

Jankauskas, M., Serackis, A., Paulauskas, N., Pomarnacki, R., & Hyunh, V. K. (2024). The Impact of the Weather Forecast Model on Improving AI-Based Power Generation Predictions through BiLSTM Networks. *Electronics*, 13(17), 3472. https://doi.org/10.3390/electronics13173472

Jankauskas, M., Katkevičius, A., & Serackis, A. (2026). Investigation of Exponent–Free LSTM Cells for Virtual Sensing Applications. *Electronics*, 15(3), 576. https://doi.org/10.3390/electronics15030576

Papers in Other Editions

Jankauskas, M., Serackis, A., Pomornacki, R., Hyunh, V. K., Šapurov, M., & Baškys, A. (2023). Short-term wind energy forecasting with advanced recurrent neural network models: A comparative study. In *2023 IEEE 10th Jubilee Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)* (pp. 1–5). IEEE. https://doi.org/10.1109/AIEEE58915.2023.10134882

Summary in Lithuanian

Įvadas

Problemos formulavimas

Disertacijoje nagrinėjamos galimybės didinti duomenimis grįstų metodų tikslumą, patikimumą, skaičiavimo efektyvumą ir praktinį naudingumą sprendžiant du vėjo energetikos uždavinius: ankstyvą vėjo turbinų gedimų aptikimą ir trumpalaikį vėjo elektrinių parko galios prognozavimą. Šiuolaikinėse vėjo energetikos sistemose sukaupiami dideli valdymo, priežiūros ir duomenų surinkimo (SCADA) bei meteorologinių laiko sekų duomenų kiekiai, tačiau ribotas registruotų gedimų skaičius, platus darbo režimų kintamumas, nevisiškas atskirų turbinų stebimumas ir nevienoda skaitinių orų prognozių įvesčių kokybė mažina įprastų slenkstinių metodų, rankinės įvesčių atrankos ir tik statistinės paklaidos minimizavimu grindžiamų tikslų efektyvumą. Todėl pagrindinė disertacijoje nagrinėjama problema yra sukurti ir optimizuoti sekų modeliavimu grįstus rekurentinius modelius, galinčius iš SCADA ir meteorologinių duomenų išmokti informatyvius laikinius požymius, sudaryti virtualųjį jutiklį ir atlikti nuokrypio analize grįstą anomalijų aptikimą, parinkti tinkamas meteorologines įvestis paros į priekį prognozėms bei kartu sumažinti skaičiavimo sąnaudas ir pagerinti praktinį ekonominį naudingumą.

Darbo aktualumas

Sparčiai plėtojantis vėjo energetikai, didėja patikimos turbinų eksploatacijos ir tikslaus trumpalaikio galios prognozavimo poreikis, nes abu šie aspektai tiesiogiai veikia priežiūros sąnaudas, elektros sistemos stabilumą ir dalyvavimą elektros rinkose. Vėjo elektrinių

parkai generuoja didelius SCADA ir meteorologinių laiko sekų duomenų kiekius, tad duomenimis grįsti metodai tapo natūraliu pasirinkimu ankstyviesiems gedimams aptikti ir galiai prognozuoti.
Lietuvos kontekste gimininginiai tyrimai daugiausia buvo skirti vietinių vėjo išteklių vertinimui ir prognozavimui (Gecevičius et al. 2019; Jankevičienė 2024; Katinas et al. 2017). Ši disertacija išplečia kontekstą, nes joje sujungiami SCADA pagrįstas ankstyvas gedimų aptikimas ir trumpalaikis vėjo elektrinių parko galios prognozavimas, kartu įvertinant rekurentinių modelių efektyvumą, praktinio diegimo apribojimus, aktualius elektros ir elektronikos inžinerijai, ir prognozės paklaidos ekonominį poveikį. Todėl pažangesni požymių sudarymo, modelių optimizavimo ir ekonomiškai prasmingo prognozavimo metodai išlieka aktualūs šiuolaikinėms vėjo energetikos sistemoms.

Tyrimų objektas

Rekurentiniais neuroniniais tinklais grįsti ankstyvojo gedimų aptikimo ir trumpalaikio galios prognozavimo metodai.

Darbo tikslas

Sukurti ir ištirti duomenimis grįstus metodus, skirtus pagerinti trumpalaikio vėjo elektrinių galios prognozavimo ir ankstyvo vėjo turbinų gedimų aptikimo tikslumą, efektyvumą ir praktinį pritaikomumą, remiantis SCADA ir meteorologinių prognozių duomenimis.

Darbo uždaviniai

Darbo tikslui pasiekti sprendžiami šie uždaviniai:

1. Sukurti ir ištirti virtualiu jutikliu pagrįstą metodą, skirtą techninei būklei stebėti ir ankstyviems vėjo turbinų gedimams aptikti, taikant SCADA laiko eilučių duomenis ir įvertinant informatyviausių požymių atranką bei prognozavimo tikslumą lemiančius veiksnius.
2. Išanalizuoti ir optimizuoti virtualiajam jutikliui taikomas rekurentinių neuroninių tinklų struktūras, įvertinant požymių sekų sudarymą, mokymo schemą ir alternatyvias aktyvavimo funkcijas, siekaint padidinti tikslumą ir sumažinti praktiniam diegimui svarbias skaičiavimo sąnaudas.
3. Sukurti ir ištirti BiLSTM pagrindu sudarytą trumpalaikio vėjo elektrinių parko galios prognozavimo metodą, naudojant meteorologinių prognozių duomenis, ir įvertinti meteorologinių prognozių šaltinių poveikį bei tikslo funkcijos su normalizuotu „Nord Pool“ kainos daugikliu tinkamumą išankstinėms energijos gamybos prognozėms.

Tyrimų metodika

Sprendžiant disertacijos uždavinius taikyti duomenimis grįstos laiko sekų analizės metodai, rekurentiniai neuroniniai tinklai ir anomalijų aptikimo metodai, paremti nuokrypio

tarp matuotos ir prognozuotos reikšmės stebėsena. Metodika apėmė SCADA ir meteorologinių duomenų paruošimą, požymių atranką, laikinį agregavimą, LSTM, GRU ir BiLSTM architektūrų taikymą bei virtualiuoju jutikliu grįsto ankstyvo gedimų aptikimo ir trumpalaikio vėjo elektrinių galios prognozavimo modelių lyginamąjį eksperimentinį vertinimą. Modeliai vertinti pagal šakninę vidutinę kvadratinę paklaidą, vidutinę absoliučiąją paklaidą, R^2 rodiklį, liekanų standartinį nuokrypį ir kitus statistinius kriterijus, o ten, kur tai svarbu praktiniam taikymui, papildomai taikyti su ekonominiu rezultatu susieti rodikliai. Hiperparametrų parinkimas ir modelių palyginimas atlikti eksperimentiškai, taip pat analizuoti skirtingi skaitinių orų prognozių šaltiniai ir alternatyvios aktyvavimo funkcijos.

Darbo mokslinis naujumas

Disertacijos mokslinį naujumą sudaro šie pagrindiniai indėliai:

- Sukurtas ankstyvų vėjo turbinų gedimų aptikimo metodas, pagrįstas virtualiuoju jutikliu ir nuokrypio tarp išmatuotų bei prognozuotų reikšmių analize, leidžiantis stebėti dinamiškai kintančius signalus retų gedimų atvejais.
- Parodyta, kad adaptyviai nustatomas nuokrypio slenkstis leidžia 20–30 dienų iš anksto prognozuoti jutikliu stebimo komponento gedimą.
- Pasiūlytos ribotos neeksponentinės aktyvavimo funkcijų alternatyvos rekurentiniams neuroniniams tinklams, leidžiančios sumažinti skaičiavimo sąnaudas nemažinant prognozavimo tikslumo.
- Parodyta, kad Lietuvos vėjo elektrinių parko atveju agreguotas meteorologinių prognozių šaltinis, naudojamas su BiLSTM pagrįstu rekurentiniu modeliu, lemia didesnę trumpalaikio prognozavimo paklaidą.
- Pasiūlyta alternatyvi modelio mokymo tikslo funkcija, įtraukianti „Nord Pool“ elektros kainą ir mažinanti finansinius nuostolius, atsirandančius dėl prognozės ir faktinės gamybos neatitikimo.

Darbo rezultatų praktinė reikšmė

Virtualiuoju jutikliu grįstas metodas leidžia anksčiau aptikti neįprastus SCADA signalų pokyčius ir padeda laiku planuoti techninę priežiūrą, o sukurtos požymių atrankos, laikino agregavimo ir rekurentinių modelių parinkimo gairės leidžia kurti tikslesnius ir skaičiavimo požiūriu efektyvesnius modelius.

Trumpalaikio galios prognozavimo rezultatai padeda veiksmingiau naudoti meteorologinius ir eksploatacinius duomenis 24 valandų prognozėms ir parodo, kad prognozavimo modelius tikslinga vertinti ne tik pagal statistinį tikslumą, bet ir pagal prognozės paklaidos lemiamus finansinius nuostolius.

Disertacijos rezultatai buvo panaudoti įgyvendinant projektą „Pramoniniai interneto metodai elektros energijos keitimo sistemų stebėjimui ir diagnostikai“, Nr. S-BMT-21-5 (LT08-2-LMT-K-01-040).

Ginamieji teiginiai

1. Nuokrypį tarp matuotos ir virtualiojo jutiklio prognozuotos reikšmės vertinant adaptyviu judančios medianos slenksčiu vietoje fiksuoto slenksčio, perspėjimo laikotarpį iki gedimo galima pailginti apie 50 %.
2. Rekurentinių laiko eilučių modelių tikslumas priklauso nuo įvesties pateikimo ir mokymo parametrų parinkimo, o parinkus laikinį agregavimą, įvesties sekos ilgį ir mokymo duomenų laikotarpį, prognozavimo paklaidą galima sumažinti daugiau kaip 20 %, palyginti su kitomis tyrime nagrinėtomis konfigūracijomis.
3. Rekurentinių modelių skaičiavimo efektyvumą galima didinti standartines eksponentines aktyvavimo funkcijas pakeičiant tinkamai parinktomis ribotomis neeksponentinėmis alternatyvomis, taip netiesinių skaičiavimų sąnaudas sumažinant daugiau kaip 30 % mažuose modeliuose ir daugiau kaip 60 % aktyvavimo funkcijų dedamosios didesniuose modeliuose, nemažinant prognozavimo tikslumo.
4. Trumpalaikiame vėjo elektrinių parko galios prognozavime parinkus ICON meteorologinių prognozių šaltinį vietoje GFS Global ar GEM Global, prognozavimo paklaidą galima sumažinti daugiau kaip 12 %, o taikant tikslo funkciją su normalizuotu „Nord Pool“ kainos daugikliu ekonominį rezultatą galima pagerinti daugiau kaip 18 %.

Darbo rezultatų aprobavimas

Disertacijos rezultatai paskelbti trijuose *Clarivate Analytics Web of Science* duomenų bazės žurnaluose, turinčiuose citavimo rodiklį, ir viename konferencijos leidinyje. Disertacijos rezultatai pristatyti septyniose konferencijose ir seminaruose Lietuvoje bei užsienyje:

- Faculty of Electronics Scientific Seminar, 2025 m., Vilniuje, Lietuvoje.
- Modelling, Data Analytics and AI in Engineering. 2024. Porto, Portugal.
- UiA to the Renewable Energy Group Seminar. 2024. Grimstad, Norway.
- Data Analysis Methods for Software Systems. 2024. Druskininkai, Lithuania.
- The 10th Jubilee IEEE Workshop on Advances in Information, Electronic and Electrical Engineering. 2023. Vilnius, Lithuania.
- Data Analysis Methods for Software Systems. 2022. Druskininkai, Lithuania.
- Electronics and Electrical Engineering. 25th Lithuanian Conference of Young Scientists. 2022. Vilnius, Lithuania.

Disertacijos struktūra

Disertaciją sudaro įvadas, trys pagrindiniai skyriai, bendrosios išvados ir literatūros sąrašas.
Disertacijos apimtis yra 114 puslapiai, pateikta 12 paveikslų, 67 formulės ir 37 lentelės. Tekste cituojami 102 bibliografinis šaltinis.

Padėka

Dėkoju Prof. Dr Artūrui Serackiui už vadovavimą, Agderio universitetui ir Prof. Dr Joao Leal už bendradarbiavimą, taip pat savo šeimai už palaikymą rengiant disertaciją.

1. Duomenimis grįsta techninė priežiūra ir prognozavimas vėjo energetikos sistemose

Pirmajame disertacijos skyriuje aptariami du duomenimis grįsti vėjo energetikos uždaviniai: ankstyvas gedimų aptikimas pagal SCADA signalus ir trumpalaikis vėjo elektrinių parko galios prognozavimas pagal istorinius matavimus bei meteorologines prognozes. Abu uždaviniai nagrinėjami kartu, nes abiem atvejais rezultatą lemia ne tik pasirinktas modelis, bet ir duomenų paruošimas, požymių sudarymas bei vertinimo kriterijai.

Techninės būklės stebėsenos uždaviniuose nuolat tenka susidurti su tais pačiais ribojimais: retais ir nesubalansuotais gedimų pavyzdžiais, vėluojančiu gedimų registravimu, darbo režimų kaita ir sezoniškumu. Dėl to vien tiesioginis temperatūrų ar kitų signalų slenkstinis stebėjimas dažnai neleidžia patikimai atskirti ankstyvos degradacijos nuo natūralaus kintamumo. Tokiomis sąlygomis nuosekliausia kryptis yra virtualiojo jutiklio taikymas kartu su nuokrypio tarp matuotos ir prognozuotos reikšmės analize, kai pirmiausia modeliuojamas įprastas jutiklio matuojamo signalo kitimas, o po to vertinama, ar šis skirtumas ima sistemingai didėti. Toks pasirinkimas natūraliai sutelkia dėmesį į kitą klausimą: kaip paruošti duomenis ir požymius, kad silpni degradacijos signalai nebūtų prarasti.

Požymiai šiuose tyrimuose formuojami keturiais pagrindiniais būdais: rankiniu būdu apskaičiuojant statistinius požymius, taikant virtualųjį jutiklį ir nuokrypio analizę, sudarant bendros būklės rodiklius iš kelių SCADA kanalų ir taikant automatinį požymių sudarymą giliojo mokymosi metodais. Svarbiausia ne pati metodų klasifikacija, o tai, kad duomenų paruošimas tampa savarankišku sprendimu: mastelio keitimas, išskirčių apdorojimas, vėluojančių gedimų žymėjimas ir tikslinių reikšmių formavimas nulemia, kokią degradacijos informaciją modelis gali išmokti. Todėl agregavimas, požymių atranka ir tikslinių reikšmių formavimas nagrinėjami kaip teorinių ir eksperimentinių tyrimų veiksniai.

Sprendimo logikos požiūriu techninės priežiūros modelių negalima vertinti vien pagal klasifikavimo tikslumą. Svarbūs tampa ir klaidingų pavojaus signalų kaštai, perspėjimo iki gedimo laikotarpio praktinė reikšmė ir sprendimo sluoksnio sąsaja su realiais priežiūros veiksmais. Dėl to aktualūs ne tik klasikiniai mašininio mokymosi modeliai ir sekoms modeluoti skirti LSTM ar BiLSTM tinklai, bet ir anomalijų aptikimo metodai, adaptyvūs slenksčiai bei modelių vykdymo skaičiavimo sąnaudos, kai sprendimai turi būti taikomi ribotų resursų aplinkoje. Todėl disertacijoje toliau nagrinėjama perspėjimo logika, adaptyvūs slenksčiai ir vykdymo sąnaudų vertinimas.

Trumpalaikio galios prognozavimo kokybę lemia ne vien pasirinkta architektūra. Šiame darbe toliau nagrinėjami prognozės horizontas, meteorologinių įvesčių kokybė, meteorologinių prognozių šaltinio parinkimas, mokymo strategija ir prognozės klaidos vertinimas rinkos požiūriu. Vien šakninė vidutinė kvadratinė paklaida, vidutinė absoliučioji paklaida ar R^2 ne visada atspindi tikrąją prognozės vertę, nes elektros rinkoje svarbios ir su balansavimu susijusios sąnaudos. Todėl toliau telkiamasi į tuos sprendimus, kurie antra-

jame ir trečiajame skyriuose lemia meteorologinių prognozių šaltinių palyginimą, modelio mokymo tikslo funkcijos parinkimą ir rinkos rezultato vertinimą.

Patikimiems vėjo energetikos sprendimams reikia ne vien tikslaus modelio, bet ir nuoseklaus laiko eilučių duomenų paruošimo, požymių sudarymo, meteorologinių įvesčių atrankos, perspėjimo logikos ir praktinio vertinimo metodikos. Šie veiksniai teoriškai pagrindžiami antrajame skyriuje, o trečiajame tikrinami eksperimentais.

2. Virtualiojo jutiklio ir vėjo elektrinių galios prognozavimo teoriniai pagrindai

Antrajame disertacijos skyriuje teoriškai pagrindžiami du pagrindiniai darbo sprendiniai: virtualusis jutiklis ankstyvam gedimų aptikimui ir trumpalaikio vėjo elektrinių parko galios prognozavimo schema. Ankstyvus gedimų požymius čia tikslingiausia aptikti ne tiesiogiai stebint fizinio jutiklio reikšmę, o modeliuojant tikėtiną stebimo signalo kitimą įprastomis sąlygomis ir vertinant nuokrypį tarp matuotos ir prognozuotos reikšmės.
Šioje disertacijoje virtualusis jutiklis yra regresinis modelis, kuris iš kitų SCADA kanalų apskaičiuoja tikėtiną stebimo signalo reikšmę įrenginio mazgui veikiant įprastomis sąlygomis. Dėl to ankstyvas gedimų aptikimas grindžiamas ne pačia temperatūros ar kito signalo verte, o nuoseklia nuokrypio tarp matuotos ir prognozuotos reikšmės analize.

Tokį sprendimą pagrindžia ir tai, kaip kinta pats stebimas signalas. Viešame EDP duomenų rinkinyje SCADA kanalo, įvardyto kaip guolio temperatūra, reikšmės įprasto veikimo metu kinta nuo 20,00 iki 71,50 °C, o mediana siekia 52,61 °C. Tuo tarpu artėjant gedimui medianos pokytis sudaro tik 0,22–0,89 °C. Vadinasi, įprasto veikimo metu stebimas temperatūros kitimo diapazonas yra maždaug 58–234 kartus didesnis už ankstyvą priešgediminį medianos poslinkį, todėl vien tiesioginis slenkstinis stebėjimas neleidžia atskirti ankstyvos degradacijos nuo natūralaus signalo kitimo.

Požymių atrankai buvo įvertinti 82 kandidatiniai požymiai, sudaryti iš SCADA duomenų, pagal jų monotoniškumą, Laplaso įvertį ir dispersiją. Aukščiausi svarbos įverčiai gauti su galia ir apkrovos režimu susijusiems kintamiesiems. Tačiau galutinis 14 požymių rinkinys sudarytas remiantis ne vien statistiniais įverčiais, bet ir fizine vėjo jėgainės veikimo logika. Į jį įtraukti generatoriaus ir rotoriaus sukimosi greičio, aplinkos vėjo greičio, vėjo krypties ir aplinkos temperatūros požymiai, nes jie kartu aprašo mechaninę apkrovą, aerodinaminį poveikį ir išorines šilumines sąlygas.
Praktiniam taikymui svarbus ne tik modelio tikslumas, bet ir vykdymo sąnaudos, todėl teorinėje dalyje išanalizuotos skirtingų aktyvavimo funkcijų skaičiavimo sąnaudos, išreikštos pasvertais ciklais. Supaprastintas vienos LSTM ląstelės pavyzdys aiškiausiai parodo, kokią įtaką bendrai sekos kainai turi aktyvavimo funkcijų pasirinkimas ir kodėl alternatyvias funkcijas verta tikrinti eksperimentiniu būdu.

Supaprastinto vieno sluoksnio LSTM modelio su viena rekurentine ląstele ir 11 įvesties požymių analizė parodė, kad standartinių eksponentinių aktyvavimo funkcijų keitimas gali reikšmingai sumažinti sekos apdorojimo sąnaudas. Analizę išplėtus iki dviejų sluoksnių BiLSTM architektūros su 11 įvesčių ir dviem sluoksniais po 128 ir 100 ląstelių, bendros sekos skaičiavimo sąnaudos sumažėjo nuo $893488T$ iki $867496T$, t. y. 2,91 %, o vien aktyvacijų dedamoji per vieną agreguotą laiko žingsnį sumažėjo nuo 39 216 iki 13 224 svertinių

ciklų, t. y. 66,3 %. Būtent dėl to eksperimentinėje dalyje tikrinamos ribotos neeksponentinės aktyvavimo funkcijos.
Trumpalaikio vėjo galios prognozavimo atveju meteorologinių duomenų šaltinis yra vienas iš modeliavimo sprendimų, o ne tik duomenų gavimo etapas. Skirtingi skaitinių orų prognozių modeliai skiriasi erdvine ir laiko skiriamąja geba, atnaujinimo dažniu ir tuo, kaip atvaizduoja vietovei būdingus orų procesus, todėl tas pats nominalus kintamasis skirtinguose šaltiniuose gali turėti nevienodą prognozinę vertę.

Tyrime nagrinėti ECMWF-IFSA, UKMO-EURO4B, MEPS, ICON, GEM Global, „Meteo France“, GFS Global ir „Best Match“ meteorologinių prognozių šaltiniai, kurie skiriasi laiko ir erdvine skiriamąja geba, atnaujinimo dažniu bei prieinamais meteorologiniais kintamaisiais.

Galutinę prognozės kokybę lemia ne vien pats BiLSTM modelis. Ne mažiau svarbi visa duomenų paruošimo grandinė: meteorologinių šaltinių pasirinkimas, jų sinchronizavimas su istoriniais galios matavimais, požymių sudarymas ir mokymo eiga. Todėl meteorologinių prognozių šaltinį tenka rinktis kaip vieną iš prognozavimo sistemos projektavimo sprendimų.
Trumpalaikio galios prognozavimo kokybės nepakanka vertinti vien šakninės vidutinės kvadratinės paklaidos, vidutinės absoliučiosios paklaidos ar R^2 rodikliais. Elektros rinkoje prognozės klaidos sukelia balansavimo sąnaudas, kurios priklauso ne tik nuo klaidos dydžio, bet ir nuo jos krypties bei konkrečios rinkos situacijos. Todėl modelio vertinimo kriterijai turi apimti ir ekonominę prognozės įtaką, o mokymo tikslas negali būti laikomas vien techniniu optimizavimo žingsniu.
Teorinį tyrimą apibendrina šios išvados:

- Ankstyvam gedimų aptikimui tikslinga modeliuoti įprastą jutiklio matuojamo signalo kitimą ir virtualiuoju jutikliu vertinti nuokrypį tarp matuotos ir prognozuotos reikšmės.
- Prognozuojamai informacijai svarbiausi yra turbinos apkrovą ir veikimo režimą apibūdinantys kintamieji, todėl virtualiojo jutiklio įvestį tikslinga sudaryti iš 14 požymių, aprašančių generatoriaus ir rotoriaus sukimosi greitį, vėjo greitį, vėjo kryptį ir aplinkos temperatūrą.
- Preliminarus aktyvavimo funkcijų skaičiavimo sąnaudų vertinimas parodė, kad standartinė logistinė sigmoidė ir hiperbolinis tangentas yra skaičiavimo požiūriu brangiausios nagrinėtos funkcijos, o mažiausias sąnaudas duoda *Hard sigmoid* ir *Hard hyperbolic tangent*.
- Vienos LSTM ląstelės sekos skaičiavimo sąnaudos pakeitus logistinę sigmoidę ir hiperbolinį tangentą funkcijomis *Hard sigmoid* ir *Hard hyperbolic tangent* sumažėja nuo $182T$ iki $125T$, t. y. 31,32 %, o pasirinktoje dviejų sluoksnių BiLSTM architektūroje aktyvacijų dedamoji sumažėja nuo 39 216 iki 13 224 svertinių ciklų, t. y. 66,3 %.
- Trumpalaikiame galios prognozavimo uždavinyje meteorologinių prognozių šaltinio parinkimas, įvesčių paruošimas ir modelio mokymo tikslo funkcija yra atkiri modeliavimo sprendiniai, lemiantys tiek prognozės paklaidos dydį, tiek rinkos rezultatą.

S3.1 lentelė. Vidutinė virtualiojo jutiklio paklaida skirtingiems laikinio agregavimo intervalams ir rekurentinėms struktūroms

Modelio struktūra	24 val.	6 val.	3 val.	Geriausias intervalas
LSTM–BiLSTM (128, 128)	3,8467	3,9664	3,5695	3 val.
BiLSTM (128)	4,3004	5,2343	3,9517	3 val.
GRU (128)	4,6745	4,9331	3,8933	3 val.
LSTM–LSTM (128, 100)	2,9989	4,0180	2,8595	3 val.

3. Prognozavimo ir gedimų aptikimo metodų eksperimentiniai tyrimai

Trečiajame disertacijos skyriuje eksperimentais tikrinami trys praktiniai klausimai: nuo ko labiausiai priklauso virtualiojo jutiklio tikslumas, ar aktyvavimo funkcijų pakeitimas leidžia supaprastinti modelio vykdymą neprarandant tikslumo, ir kaip trumpalaikio prognozavimo rezultatai kinta, kai jie vertinami ne vien statistiškai, bet ir rinkos požiūriu. Tyrimuose naudoti viešai prieinami EDP vėjo turbinų SCADA duomenys ir Lietuvos vėjo elektrinių parko galios matavimai, sinchronizuoti su meteorologinėmis prognozėmis, gautomis per „Open-Meteo“ API.

Pirmojoje eksperimentų dalyje tirta, kaip duomenų paruošimas ir modelio parametrai keičia virtualiojo jutiklio paklaidą. Atskirai vertintas laikinio agregavimo ir mokymo duomenų laikotarpio poveikis, taip pat įvesties sekos ilgis, mokymo epochų skaičius ir rekurentinio modelio struktūra. Tikslinis signalas buvo EDP duomenų rinkinio SCADA kanalas, įvardytas kaip guolio temperatūra, o prognozavimo kokybė vertinta pagal šakninę vidutinę kvadratinę paklaidą ir prognozės paklaidos charakteristikas.

Lentelėje S3.1 matyti aiški tendencija: visoms keturioms tirtoms struktūroms geriausias buvo 3 val. agregavimo intervalas. Šakninės vidutinės kvadratinės paklaidos vidurkis šiame intervale buvo 3,5685, palyginti su 3,9551, taikant 24 val. agregavimą, ir 4,5380, taikant 6 val. agregavimą. Geriausią vidutinį rezultatą pasiekė LSTM–LSTM struktūra su dviem sluoksniais, turinčiais 128 ir 100 ląstelių; jos šio rodiklio vidurkis taikant 3 val. intervalą buvo 2,8595. Todėl tolesniuose virtualiojo jutiklio palyginimuose 3 val. agregavimas laikytas pagrindiniu atskaitos variantu.

1 mėn. mokymo duomenų laikotarpio be registruotų gedimų nepakanka stabiliam virtualiojo jutiklio modeliui sudaryti. Geriausios LSTM-LSTM struktūros atveju, kai modelis mokytas tik iš T01 duomenų, sutrumpinus mokymo laikotarpį nuo 3 mėn. iki 1 mėn. ir pailginus testavimo intervalą iki 5 mėn., vidutinė šakninė vidutinė kvadratinė paklaida padidėjo nuo 2,5436 iki 3,9450. Sujungus kelių turbinų duomenis iš laikotarpio be registruotų gedimų ir pailginus mokymo laikotarpį iki 6 mėn., geriausio kelių turbinų LSTM-LSTM modelio šios paklaidos vidurkis sumažėjo iki 2,3013. Tai rodo, kad virtualiajam jutikliui reikia bent 3 mėn. mokymo duomenų laikotarpio, o kelių turbinų atveju jį tikslinga ilginti iki 6 mėn.

Virtualiojo jutiklio tikslumą papildomai lemia įvesčių sekos ilgis ir mokymo epochų skaičius. Mažiausia šakninės vidutinės kvadratinės paklaidos reikšmė gauta naudojant 12 agreguotų įvesčių seką ir 1800 mokymo epochų; ji siekė 1,0440. Atliekant struktūrinį palyginimą nustatyta, kad geriausia konfigūracija buvo trijų sluoksnių BiLSTM su 80 ląste-

lėmis kiekviename sluoksnyje ir *dropout* = 0,2; konfigūracija pasiekė 2,2127 nagrinėjamo rodiklio reikšmę ir buvo pasirinkta tolesnės aktyvavimo funkcijų analizės metu.

S3.2 lentelė. Bazinio ir geriausių alternatyvių aktyvavimo funkcijų konfigūracijų prognozavimo kokybės rodikliai

Konfigūracija	**Test RMSE**	**Test MAE**	R^2	**Liekanų STD**	**Poslinkis**
Logistinė sigmoidė + hiperbolinis tangentas (bazė)	3,8465	2,9749	0,5676	3,1900	$-2,1493$
Hard sigmoid + Softsign	3,6022	2,8797	0,6208	3,0338	$-1,9422$
Arctangent gate + Softsign	3,6969	2,9966	0,6006	3,0366	$-2,1086$
Softsign gate + Softsign	3,7649	2,9563	0,5858	3,2095	$-1,9681$

Lentelėje S3.2 palikti originalūs aktyvavimo funkcijų pavadinimai: *Hard sigmoid*, *Softsign*, *Arctangent gate* ir *Softsign gate*.

Iš lentelėje S3.2 pateiktų duomenų matyti, kad ribotos neeksponentinės aktyvavimo funkcijos nebūtinai mažina prognozavimo kokybę. Geriausia alternatyvi aktyvavimo funkcijų pora *Hard sigmoid + Softsign* sumažino testavimo šakninę vidutinę kvadratinę paklaidą nuo 3,8465 iki 3,6022 ir padidino R^2 nuo 0,5676 iki 0,6208. Šiame eksperimente skaičiavimo požiūriu paprastesnės aktyvavimo funkcijos išliko tinkamos ir prognozavimo kokybės nepablogino.

S3.3 lentelė. Perspėjimo laikas iki gedimo, gautas taikant skirtingas virtualiojo jutiklio architektūras ir slenksčio nustatymo metodus

Komponento gedimas	**LSTM–LSTM**		**LSTM–BiLSTM**	
	Slenkstis		**Slenkstis**	
	$\mu + 3\sigma$	**Judanti mediana**	$\mu + 3\sigma$	**Judanti mediana**
Pavarų dėžė	14	22	22	32
Generatorius	15	24	26	37
Generatoriaus guolis	13	24	22	36
Transformatorius	23	22	24	35
Hidraulinė sistema	12	22	20	33

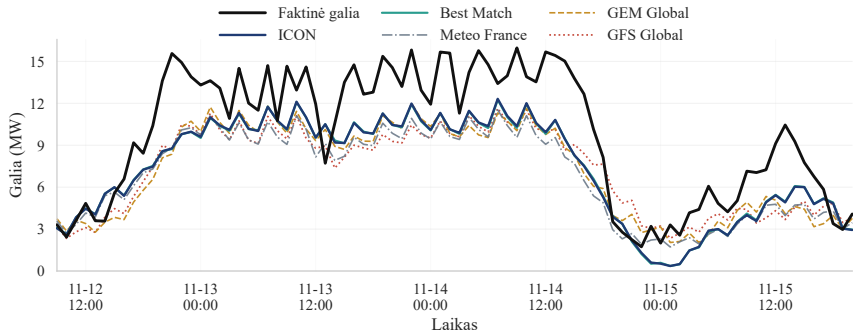
Dvisluoksnio BiLSTM modelio atveju adaptyvus judančios medianos slenkstis buvo pranašesnis už fiksuotą $\mu + 3\sigma$ taisyklę. Vidutinis laikas nuo pirmojo perspėjimo iki gedimo padidėjo nuo 22,8 iki 34,6 dienos, o ilgiausias perspėjimas prieš generatoriaus gedimą pasiekė 37 dienas. Šie rezultatai patvirtino, kad nuokrypis tarp matuotos ir prognozuotos reikšmės gali būti praktiškai naudingas ankstyvo perspėjimo signalas dar iki registruoto komponento gedimo.

Antrojoje eksperimentų dalyje tirtas trumpalaikis vėjo elektrinių parko galios prognozavimas, naudojant BiLSTM modelį ir skirtingų meteorologinių prognozių šaltinių duomenis. Tyrimui naudoti Lietuvos vėjo elektrinių parko, sudaryto iš šešių po 2,75 MW turbinų, galios matavimai, sinchronizuoti su meteorologinėmis prognozėmis. Optimizuotas požymių rinkinys apėmė vėjo greitį 10 ir 80 m aukštyje, vėjo gūsius 80 m aukštyje, slėgį jūros lygyje ir ankstesnes galios reikšmes.

S3.4 lentelė. Skirtingų meteorologinių prognozių šaltinių palyginimas BiLSTM pagrindu sudarytame trumpalaikio galios prognozavimo modelyje

Modelis	**RMSE**	**MAE**	R^2	**NMAE**
Best Match	1,7604	1,2580	0,85478	0,21643
ICON	1,7565	1,2549	0,85543	0,21591
GEM Global	2,0086	1,4447	0,81094	0,24857
Meteo France	1,9520	1,3909	0,82146	0,23930
GFS Global	2,0242	1,4621	0,80801	0,25155

Geriausi statistiniai rezultatai gauti naudojant ICON ir „Best Match“ šaltinius. ICON pasiekė mažiausią šakninę vidutinę kvadratinę paklaidą, vidutinę absoliučiąją paklaidą ir normalizuotą vidutinę absoliučiąją paklaidą bei didžiausią R^2, o „Best Match“ nuo jo atsiliko tik nežymiai. Lyginant su GFS Global, ICON sumažino šakninę vidutinę kvadratinę paklaidą apie 13,2 %, o lyginant su GEM Global – apie 12,6 %.



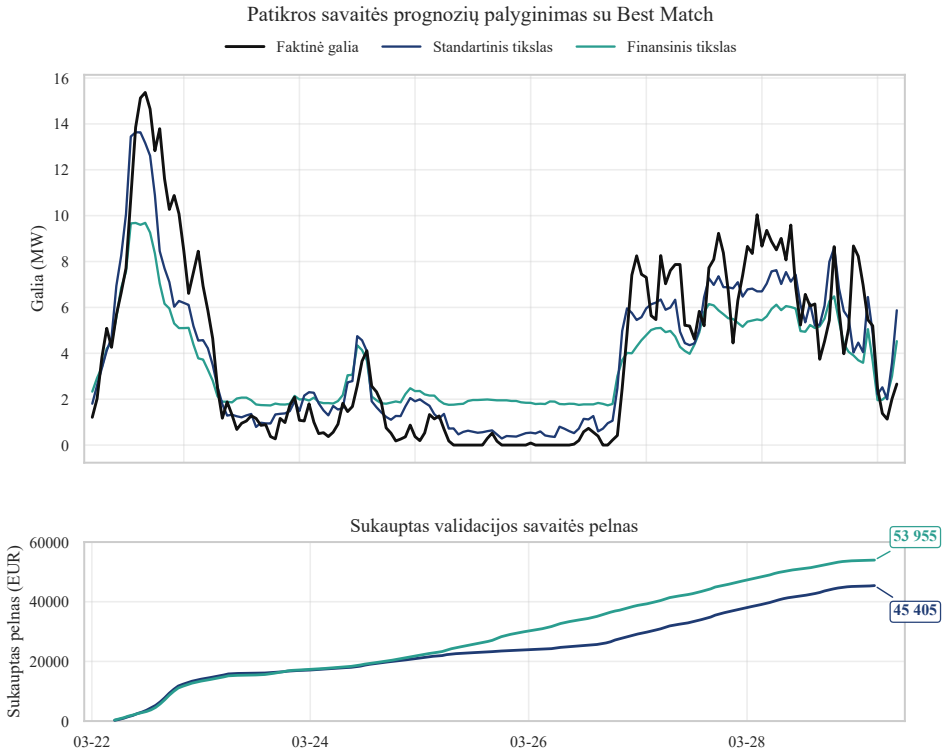
S3.1 pav. Faktinės ir prognozuotos galios palyginimas, naudojant skirtingus meteorologinių prognozių šaltinius

Paveiksle S3.1 matyti ta pati tendencija: ICON ir „Best Match“ geriau seka realią galios kreivę, o mažiau tikslūs meteorologinių prognozių šaltiniai sukelia didesnius nukrypimus, ypač staigių galios pokyčių metu ir artėjant prie vietinių ekstremumų. Todėl meteorologinių prognozių šaltinio pasirinkimą būtina laikyti vienu pagrindinių trumpalaikio prognozavimo projektavimo sprendimų.

S3.5 lentelė. Standartinės ir pasiūlytos mokymo tikslo funkcijų palyginimas patikros savaitėje su „Best Match“ meteorologiniu šaltiniu

Mokymo tikslas	R^2	**Sukauptas pelnas (EUR)**	**Pokytis**
Standartinė funkcija	0,846	45405	bazė
Pasiūlyta funkcija	0,746	53955	+8550 (+18,8%)

Rezultatai rodo vieną svarbiausių prognozavimo dalies išvadų. Taikant pasiūlytą mokymo tikslo funkciją su normalizuotu „Nord Pool“ kainos daugikliu, R^2 sumažėjo



S3.2 pav. Įprastos ir pasiūlytos tikslo funkcijos palyginimas „Best Match“ šaltiniu (viršuje parodytos faktinės ir prognozuotos galios kreivės, apačioje – sukauptas validacijos savaitės pelnas)

nuo 0,846 iki 0,746, tačiau validacijos savaitės sukauptas pelnas padidėjo nuo 45 405 iki 53 955 EUR, t. y. 8 550 EUR arba 18,8 %. Tai rodo, kad praktinis trumpalaikio prognozavimo vertinimas turi apimti ne vien statistinį tikslumą, bet ir rinkos rezultatą.

Eksperimentinius tyrimus apibendrina šios pagrindinės išvados:

- Eksperimentiškai ištyrus SCADA signalų agregavimą 24, 6 ir 3 val. intervalais nustatyta, kad virtualiojo jutiklio įvestyje tikslinga naudoti 3 val. agregavimą, nes jis davė mažiausią vidutinę šakninės vidutinės kvadratinės paklaidos reikšmę 3,5685.
- Eksperimentiškai palyginus 1, 3 ir 6 mėn. mokymo duomenų laikotarpius be registruotų gedimų nustatyta, kad 1 mėn. laikotarpio nepakanka, o sujungus kelių turbinų 6 mėn. duomenis geriausio LSTM–LSTM modelio vidutinė šakninė vidutinė kvadratinė paklaida sumažėja iki 2,3013.
- Eksperimentiškai įvertinus įvesčių sekos ilgį ir mokymo epochų skaičių nustatyta, kad mažiausia šakninės vidutinės kvadratinės paklaidos reikšmė 1,0440 gaunama naudojant 12 agreguotų įvesčių seką ir 1800 mokymo epochų.

- Parenkant virtualiojo jutiklio architektūrą, tikslinga naudoti trijų sluoksnių BiLSTM su 80 ląstelių kiekviename sluoksnyje ir *dropout* = 0,2, o aktyvavimo funkcijų eksperimente *Hard sigmoid* + *Softsign* pora sumažino testavimo šakninę vidutinę kvadratinę paklaidą nuo 3,8465 iki 3,6022 ir padidino R^2 nuo 0,5676 iki 0,6208.
- Palyginus fiksuotą $\mu + 3\sigma$ taisyklę ir adaptyvų judančios medianos slenkstį nustatyta, kad dvisluoksnio BiLSTM atveju vidutinis laikas nuo pirmojo perspėjimo iki gedimo pailgėja nuo 22,8 iki 34,6 dienos, o pavieniais atvejais leidžia perspėti iki 37 dienų prieš generatoriaus gedimą.
- Lietuvoje esančių vėjo elektrinių parkų trumpalaikiam galios prognozavimui rekomenduojama naudoti ICON meteorologinių prognozių šaltinį, nes jis šiame tyrime davė mažiausią RMSE 1,7565, o „Best Match“ pateikė artimus rezultatus.
- Paros į priekį energijos gamybą prognozuojančio modelio mokymui tikslinga taikyti pasiūlytą tikslo funkciją su normalizuotu „Nord Pool“ kainos daugikliu, nes patikros savaitės sukauptas pelnas padidėjo 8 550 EUR (18,8 %), nors R^2 sumažėjo nuo 0,846 iki 0,746.

Bendrosios išvados

1. Sukurtas virtualiuoju jutikliu grįstas metodas yra tinkamas vėjo turbinų būklės stebėsenai ir ankstyvam gedimų aptikimui pagal SCADA laiko sekas, nes normalaus veikimo metu kaip guolio temperatūros signalo kitimo intervalas (20,00–71,50 °C) gerokai viršija ankstyvą priešgediminį medianos poslinkį (0,22–0,89 °C), todėl vien tiesioginis signalo stebėjimas nėra pakankamai patikimas. Naudojant informatyvių SCADA požymių rinkinį, nuokrypio tarp matuotos ir virtualiojo jutiklio prognozuotos reikšmės analizę ir adaptyvų judančios medianos slenkstį, vidutinis perspėjimo laikas iki gedimo pailgėjo nuo 22,8 iki 34,6 dienos, o ilgiausias perspėjimo horizontas pasiekė 37 dienas iki generatoriaus gedimo.
2. Virtualiojo jutiklio rekurentinių modelių optimizavimas parodė, kad tikslumas ir praktinis pritaikomumas priklauso nuo laiko sekos reprezentacijos, mokymo schemos, architektūros ir aktyvavimo funkcijų. Tikslingiausia naudoti 3 val. agregavimo intervalą ir ilgesnę sveikos būsenos mokymo istoriją, o geriausią rezultatą pateikė trijų sluoksnių BiLSTM su 80 ląstelių kiekviename sluoksnyje ir *dropout* = 0,2 (RMSE = 2,2127); pakeitus įprastą *sigmoid* + *tanh* porą į *Hard sigmoid* + *Softsign*, testavimo RMSE sumažėjo nuo 3,8465 iki 3,6022, o R^2 padidėjo nuo 0,5676 iki 0,6208, kartu sumažinant netiesinių skaičiavimų sąnaudas.
3. Sukurtas BiLSTM grįstas trumpalaikio vėjo elektrinių parko galios prognozavimo metodas parodė, kad praktiniam rezultatui esminę įtaką turi tiek meteorologinių prognozių šaltinis, tiek mokymo tikslo funkcija. Tarp tirtų NWP šaltinių geriausią statistinį tikslumą užtikrino ICON (RMSE = 1,7565, MAE = 1,2549, NMAE = 0,21591, R^2 = 0,85543), o pasiūlyta tikslo funkcija su normalizuotu

„Nord Pool“ kainos daugikliu padidino patikros savaitės pelną nuo 45 405 iki 53 955 EUR, t. y. 8 550 EUR (18,8 %), taip patvirtindama, kad ekonomiškai naudingiausia prognozė nebūtinai sutampa su mažiausia įprastine paklaida.

Mindaugas JANKAUSKAS

INVESTIGATION OF RECURRENT NEURAL NETWORKS-BASED METHODS FOR EARLY FAULT DETECTION AND SHORT-TERM POWER FORECASTING IN WIND ENERGY APPLICATIONS

Doctoral Dissertation

Technological Sciences,
Electrical and Electronic Engineering (T 001)

REKURENTINIAIS NEURONINIAIS TINKLAIS GRĮSTŲ METODŲ TYRIMAS SIEKIANT ANKSTI APTIKTI GEDIMUS IR ATLIKTI TRUMPALAIKES GALIOS PROGNOZES VĖJO ENERGETIKOJE

Daktaro disertacija

Technologijos mokslai,
Elektros ir elektronikos inžinerija (T 001)

Anglų kalbos redaktorė Jūratė Griškėnaitė
Lietuvių kalbos redaktorė: Deimantė Grigaitė

2026 05 04. 10,8 sp. l. Tiražas 20 egz.
Leidinio el. versija *https://doi.org/10.20334/2026-027-M*
Vilniaus Gedimino technikos universitetas
Saulėtekio al. 11, 10223 Vilnius,
Spausdino UAB „Ciklonas“,
Žirmūnų g. 68, 06127 Vilnius