Predicting Flight Delays Using Artificial Neural Networks (Project: INROVO)

Dennis Schubert[1] and **Andreas Deutschmann**[1]

[1]Department of Engineering and Natural Sciences, University of Applied Sciences, Wildau

dennis.schubert@th-wildau.de, andreas.deutschmann@th-wildau.de

Introduction

Flight delays are a key indicator of inefficiencies in the aviation industry. They are caused by a wide range of factors, including:

- **Adverse weather conditions**
- **Technical failures**
- **Airport congestion**

These delays result in:

- **Economic loss**
- **Increased environmental impact**
- **Reduced passenger satisfaction**
- **Operational and sustainability challenges**
- **Flight cancellations**

⇒ Punctuality is therefore a critical performance metric. Improving the accuracy of delay predictions is essential for enhancing operational efficiency and service quality.

Research Motivation

Despite extensive research efforts, accurately predicting aircraft arrival and departure delays remains a persistent challenge. Conventional models often fail to account for the highly complex and dynamic nature of air traffic operations.

Research Objective: Employ data-driven approaches—particularly neural networks—to enhance the accuracy of off-block time delay predictions.

Data Processing

- Preparation of the dataset for machine learning analysis.

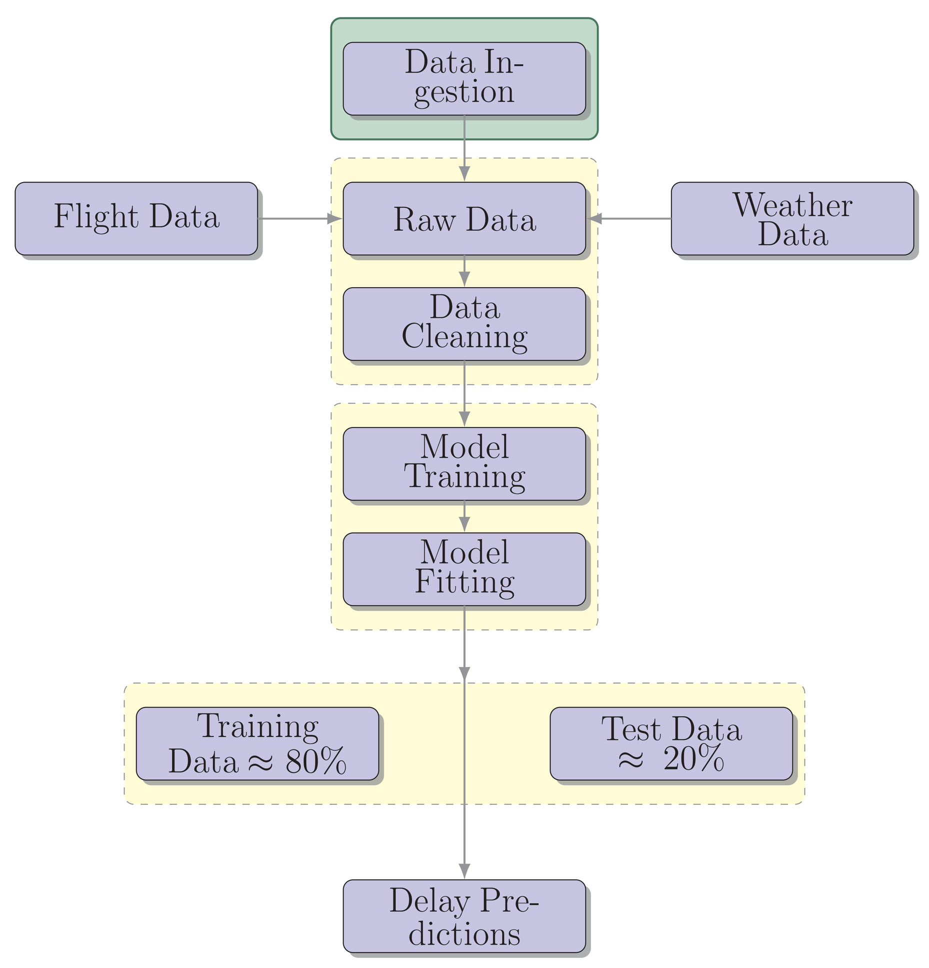

Figure 1: Data processing pipeline illustrating the workflow from data ingestion to model evaluation.

Methodology

- **Input Features:**
  - **Meteorological:** Visibility, wind speed, precipitation, temperature.
  - **Flight Operations:** Departure/arrival time, airline, airport congestion, historical delay patterns.
  - **Embeddings:** Categorical features (e.g., airline codes) encoded as trainable embeddings.
- **Neural Network Architecture:**
  - **Type:** Feedforward Neural Network with k hidden layers.
  - **Layers:** Input(n features) $\rightarrow$ Dense(ReLU) $\rightarrow$ Dropout($p = 0.2$) $\rightarrow$ Output(sigmoid/linear).

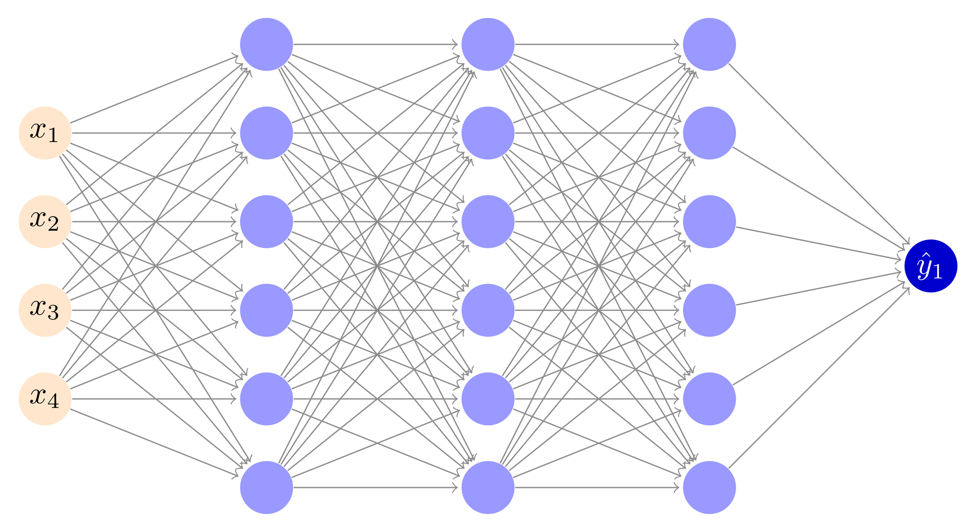

Figure 2: A simple neural network with 4 inputs, three hidden layers (6 neurons each), and one output neuron.

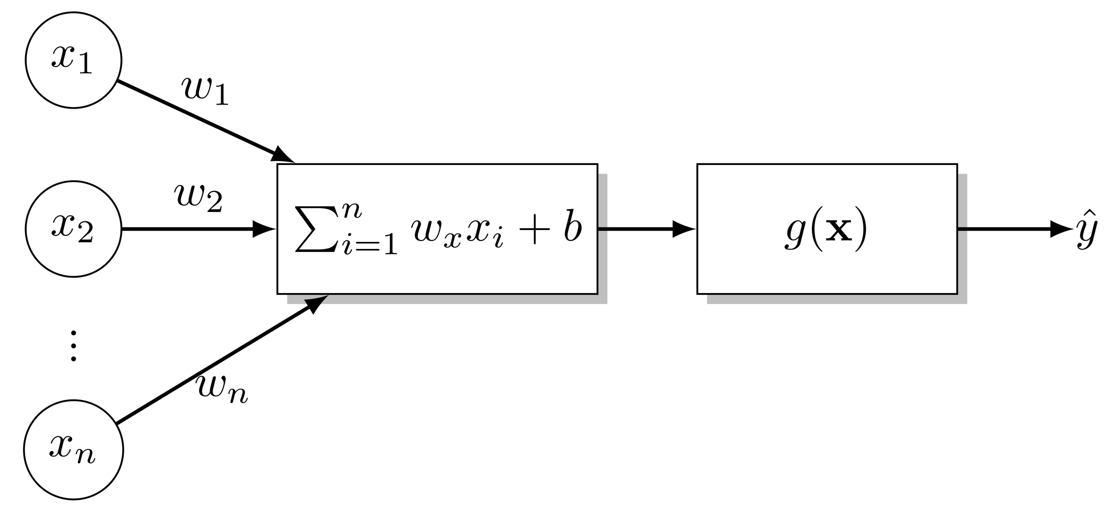

Figure 3: Perceptron with input vector $\mathbf{x}$, weights $\mathbf{w}$, a weighted sum, and activation function $g(\mathbf{x})$ and output $\hat{y}$.

- **Training Protocol:**
  - **Loss:** Mean Squared Error: most commonly used loss functions for regression tasks in neural networks
  - **Optimizer:** Stochastic Gradient Descent (SGD) with early stopping on the validation set (other optimizers are also available).

Example: Stochastic Gradient Descent (SGD)

- Stochastic Gradient Descent (SGD) is one of the most widely used optimization algorithms for training neural networks
- iteratively updates the model parameters to minimize a loss function based on a small, randomly selected subset (mini-batch) of the training data
- **Central idea:** Instead of computing gradients on the entire dataset, SGD approximates the gradient using a single or a few data points:

$$\theta_{k+1} = \theta_k - \eta \cdot \nabla_\theta \mathcal{L}(\theta_k | x_i, y_i) \,.$$

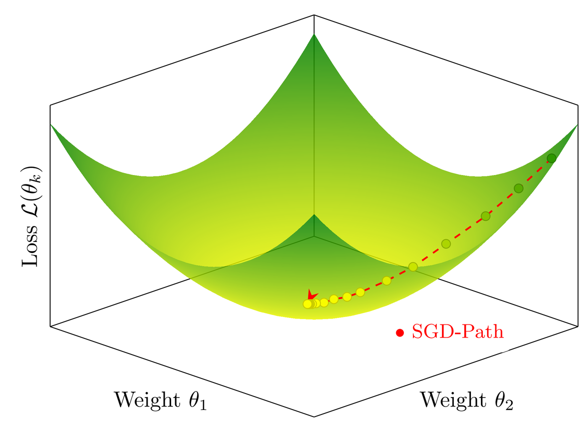

Figure 4: Illustration of the Stochastic Gradient Descent (SGD) optimization algorithm.

Outlook

- **Integration of additional data sources:** e.g., real-time air traffic data, weather conditions, or socioeconomic indicators.
- **Model comparison and ensemble techniques** Combining multiple machine learning models to improve prediction accuracy.
- **Use of time-series based approaches?**