**Tables: Data Overview**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Frequency** | **Micro** | **Industry** | **Macro** | **Finance** | **Demographic** | **Other** | **Total** |
| **Yearly** | 6,538 | 3,716 | 3,903 | 6,519 | 1,088 | 1,236 | 23,000 |
| **Quarterly** | 6,020 | 4,637 | 5,315 | 5,305 | 1,858 | 865 | 24,000 |
| **Monthly** | 10,975 | 10,017 | 10,016 | 10,987 | 5,728 | 277 | 48,000 |
| **Weekly** | 112 | 6 | 41 | 164 | 24 | 12 | 359 |
| **Daily** | 1,476 | 422 | 127 | 1,559 | 10 | 633 | 4227 |
| **Hourly** |  |  |  |  |  | 414 | 414 |
| **Total** | 25,121 | 18,798 | 19,402 | 24,534 | 8,708 | 3,437 | 100,000 |

**Benchmark methods overview:**

|  |  |  |
| --- | --- | --- |
| **Methods** | **Description** | **Type** |
| Naïve 2 | Seasonally adjusted naïve forecast using a multiplicative decomposition. | M4 Benchmark used to scale OWA. |
| Comb | Arithmetic average of SES, Holt, Damped exponential smoothing. | M4 Benchmark |
| L&K | Theta method after deseasonalization and Box-Cox transformation using an optimized transformation parameter. | M4 Competition method (Rank 8) |
| ARIMA | Automatic ARIMA framework based on information criteria. | Comparison benchmark |
| ETS | Automatic exponential smoothing state space model based on information criteria. | Comparison benchmark |
| ETSARIMA | Arithmetic average of ETS and ARIMA forecasts. | Comparison benchmark |

**RNN methods generation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Generation** | **Method** | **Local/Global** | **Description / Issues** |
| Zero | Simple RNN | Local RNN with short “memory”. |  |
|  | RNN (LSTM, GRU cells) | Local RNN with |  |
| First | DeepAR | Global RNN | Pure globally trained RNN with LSTM cells that uses autoregressive features as inputs to capture seasonality. |
|  | DeepAR\* | Global RNN, local dummies | Same as DeepAR but uses series-specific dummies to recognize individual series. |
|  | DeepState | Global RNN, local ES | Globally trained RNN that determines the best parameters of the local exponential smoothing model. |
|  | DF-RNN | Hybrid: Global RNN and local RNN. |  |
|  | MQ-RNN |  |  |
|  | Smyl | Hybrid: Global RNN and local ES. |  |
| Second | ? | Local and global | Balanced combination of global and local aspects. |

**Computational Complexity:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Method** | **OWA** | **sMAPE** | **MASE** | **Rank / Type** | **Software** | **Complexity vs. Naïve2** |
| Smyl | 0.821 | 0.114 | 1.536 | 1 | C++ | 2,777.9 |
| MM | 0.838 | 0.117 | 1.551 | 2 | R | 15,899.4 |
| L & K | 0.861 | 0.119 | 1.601 | 8 | R | 8.6 |
| Naïve2 | 1.000 | 0.136 | 1.912 | Benchmark | R | 1.0 |
| Comb | 0.898 | 0.125 | 1.663 | Benchmark | R | 11.4 |
| ARIMA | 0.903 | 0.127 | 1.666 | Comparison | R | 1,045.1 |
| ETS | 0.908 | 0.127 | 1.680 | Comparison | R | 306.5 |
| ETSARIMA | 0.880 | 0.123 | 1.624 | Comparison | R | ~1,351.6 |
| DeepAR\* | 0.910 | 0.123 | 1.668 | Global RNN | Python | - |
| DeepAR\*\* | 0.837 | 0.119 | 1.500 | Global RNN | Python | - |
| DeepState | 0.999 | 0.136 | 1.839 | Global RNN / ES | Python | - |
| Complexity vs Naïve2 = Time of method x (minutes) / Time of Naïve2 (minutes).  \* Own calculations according the framework outlined in this paper.  \*\* According to Januschowski et al. (2020).  Source: Makridakis et al. (2019), Januschowski et al. (2020) and own results. | | | | | | |

Costs:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Estimated costs by AWS Instance type:** | |
| **Method** | **Rank** | **a1.2xlarge** | **c5.2xlarge** |
| Smyl | 1 | 27.39 USD | 45.65 USD |
| MM | 2 | 156.77 USD | 261.28 USD |
| ARIMA | 20 | 12.12 USD | 17.18 USD |
| ETS | 23 | 3.02 USD | 5.04 USD |
| Instance types are chosen according the description of Makridakis et al. (2019), i.e. 8 core, 16 GiB RAM, Linux Ubuntu.  Based on on-demand costs: https://aws.amazon.com/de/ec2/pricing/on-demand/  Costs = Running time \* on-demand costs of instance type. Local energy costs are not included. | | | |