Transformer-based Prediction of IoT-Events

Adrian Rumpel[1,*], Marc C. Hennig[1] and Rainer Schmidt[1]

[1]Munich University of Applied Sciences, Lothstrasse 64, 80335 Munich, Germany

Abstract
This research uses a deep learning-based software system to integrate IoT devices into configuration management for small and medium-sized companies. The system employs transformer neural networks, which can handle long time series and complex dependencies better than previous deep learning technologies. The system also uses transformer-based event prediction, outperforming traditional ARIMA methods and other machine learning approaches such as RNNs and LSTMs. The research follows the Design Science Research method and considers the challenges of aligning different structures of event descriptions and forecasting statics. The research expects to demonstrate significant improvements in learning long-time series using transformer architectures with attention mechanisms.

Keywords
Event prediction, IoT, Transformer

1. Introduction

The Internet of Things (IoT) allows companies to develop new business models, create new products and services and optimize existing business processes [1]. Gartner estimates that global spending on IoT will exceed $300 billion in 2020 [2]. To tap into this potential and appropriately use the expected large number of IoT devices, companies must manage those devices adequately.

However, IoT devices have several special features preventing them from directly integrating into management systems, such as configuration management databases [3]. IoT devices are connected not directly with an information system but using an unreliable network connection. This makes it necessary to introduce "third" states such as "not connected" compared to the normal classification as functional or non-functional. Nevertheless, these additional states increase the complexity of IoT-Management processes.

Our idea is to replace IoT device "indeterminate" states such as "not connected" or "unknown" with educated guesses based on innovative deep-learning methods. For this purpose, a deep learning-based software system is to be developed that significantly improves the integration of IoT devices into configuration management for small and medium-sized companies. By training a deep-learning model, we want to replace "indeterminate" states of IoT devices with predictions of the status. Thus, the status "not-connected" shall be replaced by "probably functional" or "non-functional".

Research Projects Track @ RCIS 2023: The 17th International Conference on Research Challenges in Information Science, May 23–26, 2023, Corfu, Greece
adrian.rumpel@hm.edu (A.R.); mhennig@hm.edu (M.H.); rainer.schmidt@hm.edu (R.S.)

The project uses the Design Science Research [4] method to develop and evaluate the proposed system. The method considers the requirements of aligning different structures of event descriptions and forecasting statistics. The use of transformer architectures with attention mechanisms is expected to lead to significant improvements in learning long-time series, and the study results are expected to demonstrate the effectiveness of this approach.

Central to the solution is the use of so-called transformer neural networks [5], which have only recently been fundamentally researched but offer considerable advantages over previous deep learning technologies. Thus, significantly longer time series from historical values can be used to predict the state than with recurrent neural networks (RNN) and their derivatives with more complex dependencies than in usual statistical methods like ARIMA. With the help of a so-called attention mechanism [5], the transformer architecture also performs the targeted weighting of data points and can thus achieve significantly higher forecast accuracy.

2. Research Background

Predictive analytics in IoT has been practiced before with different goals and approaches for domain-specific challenges [6], [7]. Streaming big data [8]–[11] and edge computing [12], [13] use cases are covered in the existing work. An additional area of interest is the derivation of analytic insights from IoT devices to create digital twins [14], [15].

Traditional methods of time series forecasting usually rely on statistical models. Examples are autoregressive models, models with exponential smoothing functions, or structured time series models. ARIMA methods, an integrated combination of auto-regressive (AR) and moving average (MA) models [16], have been widely used in this task so far [17]. ARIMA models have been extended to allow for multivariate time series analysis and integrated with vector autoregression (VAR) models to further generalize the univariate ARIMA models [16]. ARIMA can handle seasonality in the data (SARIMA) [18] but require the data to be either stationary or non-stationary [17].

Alongside statistical methods like ARIMA and its derivates, mostly machine learning approaches are used in time series analysis [17]. The most prominent examples are RNNs, which can efficiently handle short-term dependencies. Several RNN-based architectures have been developed for prediction, and RNNs have traditionally been used for sequence modeling and have achievements in areas such as natural language processing. The core of RNN-based methods is using memory to store the preceding information but are prone to exploding or vanishing gradients [17]. Due to these problems, RNNs were enhanced to long short-term neural networks (LSTM) [19], which integrate a feedback loop that allows the output values of the network to affect the current output value at earlier points in time that prevent gradient dispersion with three gates and enable the capturing of long-term correlations in sequences. The feedback loop is the basis for processing time series with RNNs because the history of input values affects the output value. On the downside, however, LSTMs have some limitations [20] and are computationally expensive, and cannot be parallelized, limiting their potential applications [17]. Another limitation of LSTM is that transfer learning has never been satisfactorily developed. As a result, complete training must be provided for each application.

A newer architecture that has emerged recently is transformers [5]. Primarily emerging from natural language processing [21], transformers are efficient in time series analysis [17], [21], [22], especially due to their ability to parallelize computations and capture complex input dependencies. Using Transformer architectures with attention mechanisms has already led to significant improvements in learning long-time series [20]. Transformer-based architectures are often considered state-of-the-art [20] but have only been used sparsely in IoT-based learning [23]. Transformer-based event prediction promises significant advantages over previous approaches, such as the ARIMA method [16]. These theoretical considerations are supported by practical investigations in [16], [24]–[26].

In this project, we will investigate the use of statistical methods and neural networks, specifically transformers, and LSTMs, on regression and classification tasks with different data sets to determine advantages and challenges in the IoT domain.

3. Research Method

We follow the Design Science Research method from Johannesson & Perjons [4]. It has five steps: a) Problem explication, b) Define requirements, c) Design and develop artifact, d) Demonstrate artifact, and e) Evaluate artifact. We apply these steps as follows:

1. Problem explication: We identify the need for efficient and effective management of IoT devices in small and medium-sized companies. The growth of IoT technology has increased the number of IoT devices that need to be managed. We need a configuration management system that can handle the complexity of these devices.
2. Define requirements: We define the functionalities that our system must have. Our system should integrate IoT devices into configuration management for small and medium-sized companies. It should use transformer neural networks to handle long time series and complex dependencies. It should also align different structures of event descriptions and forecast statistics accurately.
3. Design and develop artifact: We design and develop our system as a deep learning-based software system. It uses transformer neural networks with attention mechanisms to improve the learning of long-time series. We use Python and deep learning frameworks such as TensorFlow and Keras.
4. Demonstrate artifact: We demonstrate our system by showing how it integrates IoT devices into configuration management for small and medium-sized companies. We use transformer neural networks to show how it handles long time series and complex dependencies. We show how it aligns different event description structures and accurately forecasts statistics using attention mechanisms.
5. Evaluate artifact: We evaluate our system by comparing it with other methods such as ARIMA, RNNs, and LSTMs. We measure its performance using accuracy, precision, recall, F-score, MAE, MSE, RMSE, etc. We expect our system to outperform other methods in learning long-time series.

4. Requirements

The integration of events requires aligning different structures of event descriptions and semantics. So, there are differences in the representation of event types that can be numeric, character-based, or specific data types. Despite these differences, the event types must be presented consistently. Another difference lies in the different identification of individual events. The simplest form is to use a continuous counter. However, different counting methods exist, such as starting values, counting direction, etc. Another possibility is timestamp-based methods with different resolutions, i.e., minutes, seconds, or fractions of a second [27]. The algorithms best suited to the specific task are to be selected and implemented for the project.

For the prediction of events, it is often helpful, for example, to include data series from sensors. Events, in turn, can support the forecasting of status. Therefore, not only the previous statics but also events and time series are directly or indirectly included in the forecasting of statuses. The forecast of statuses is to be carried out in two steps. First, a device's total set of statuses is represented as the sum of probability values. Each possible status's probability is represented by a number between 0 and 1, where the sum of the probabilities of all states is 1. Thus, the problem is projected into a classification problem. This vector is supplemented by a time stamp. A time series is created from a series of such vectors of probabilities, from which the most probable consequence vector is determined.

5. Development

For the integration platform, we have introduced the basic architecture shown in Figure 1. In the “integration” step, the data, events, and status from the IoT device are collected and integrated at normal operation (Figure 1, top). For example, data formats and types, etc., must be adapted. The “preparation” step aims to prepare the data for applying following AI methods. Typical steps are scaling the data, normalization, and mapping the data in vector space. In the “prediction model” step, prediction models for data are developed and trained based on the Transformer architecture. The data of the prediction models are transformed into digital twins of the IoT devices. In the “update” step, the prediction model is continuously updated by the data provided by the IoT device. The quality of the predictions is also constantly checked, and if necessary, an adjustment of the prediction model is required.

Finally, the “operational prediction” step is the application of the trained networks in the data prediction domain. Suppose there is an outage in the data flow in the event of an incident (Figure 1, bottom). In that case, the Transformer-based prediction models take over data, events, and status delivery.

The different prediction solutions are developed using Python and a selection of libraries, specifically Darts2 and statsmodels3 for implementing SARIMA-based models and PyTorch4 for the self-implemented LSTM and transformer neural networks. All parts are executed in Google Colab.

[2] https://unit8co.github.io/darts/
[3] https://www.statsmodels.org/
[4] https://pytorch.org/

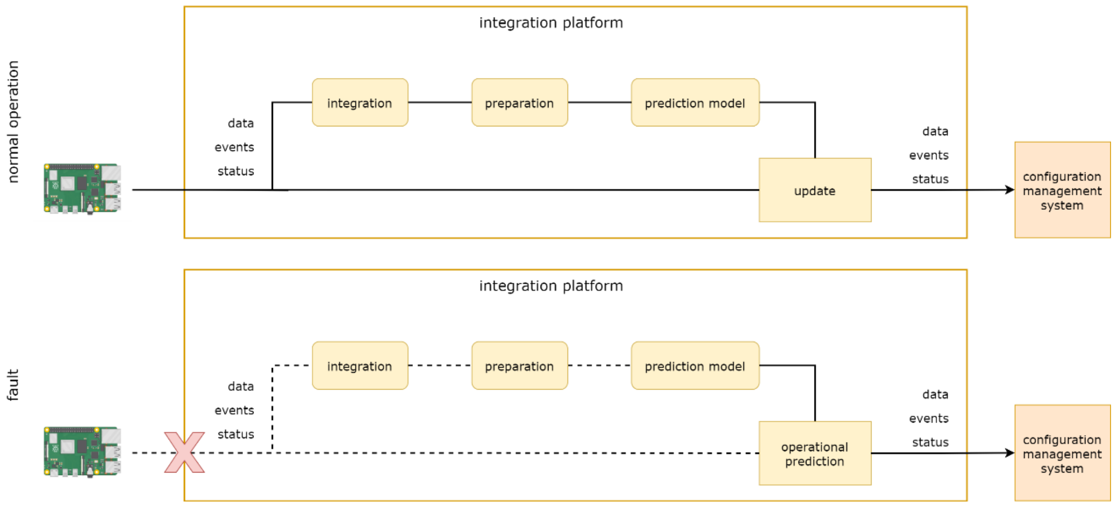

Figure 1: Architecture of integration platform in normal operation and in the event of a fault

Since this is a cooperative project with industry partners, proprietary and multi-variate real-life IoT data sets are used. Specifically, one data set with categorical data for status prediction and two numerical data sets for regression are employed during the training and evaluation (see Table 1). Since multifold cross-validation [28] can be problematic with time series [29], forward chaining is used to assess the model stability during the evaluation.

5.1. SARIMA-Model

All the data sets are used with a SARIMA model as the default statistical model, an LSTM as the baseline neural network in time series analysis, and a transformer as a relatively novel approach to derive comparisons of the most effective method. For the SARIMA models, stationarity is ensured by using the Augmented Dickey-Fuller Test before training the model. Ranges for the required model parameters were then determined based on the (partial) autocorrelation plots of the data. This was complemented by a grid search in the identified ranges, optimizing the model's Akaike Information Criterion (AIC) to find the best model. For the binary classification in the second data set, the logistic regression was used instead of the SARIMA model as the baseline statistical model, providing the timestamp in separate variables. The SARIMA model performed worse than the transformer model, reaching a minimum RMSE of 0.32 on the first data set and 4.11 on the third data set.

5.2. Neural Networks

A stacked LSTM architecture [2, 19] was developed for the neural network models to work with the given time series. The network features five LSTM layers with dropout after each layer and can be seen as a commonly used LSTM architecture in time series analysis. The best results could be reached using the Adam [30], [31] optimizer, which was determined along the learning rate and dropout during a grid search. While the results for the LSTM are not yet available for the categorical second data set, a minimum RMSE of 0.55, which is slightly worse than the SARIMA model and of 2.32 on the third data set, could be reached with the optimized models.

Dataset	Prediction	Description
Dataset 1	Regression	Temperature sensor measurements over two years in 10-minute-steps in Germany.
Dataset 2	Classification	Measurement of Sensor defects and missing values with binary classes (sensor available/unavailable) every 60 seconds.
Dataset 3	Regression	Availability of free stations as measured by multiple sensors in an EV charging system with data points every 10 minutes.

Table 1
Overview of the employed data sets

The transformer is implemented as a "vanilla" transformer [5], [22] with timestamp encoding, including an encoder and decoder with four layers each. This constitutes a very basic transformer architecture that is comparable to Wu et al. [32] but extended by the timestamp encoding mechanism used by Zhou et al. and Wu et al. [33], [34] to leverage additional information that might be present in the data [22]. The model parameters were optimized with a grid search as with the LSTM. For both neural networks, numerical variables are standardized before the training. The transformer outperformed the other models on all regression data sets with an RMSE of 0.22 on the first and 0.31 on the third data set. On the second data set, an accuracy of 0.97 on par with the logistic regression was achieved.

6. Demonstration and Evaluation

The preliminary results, as displayed in Table 2, already show an advantage of the transformer architecture compared to the other approaches. For the evaluation, the Root Mean Squared Error (RMSE) was used for the regression, which is a commonly used metric for regression model performance [35], [36]. The categorical results are evaluated using accuracy.

	Transformer	LSTM	SARIMA	Logistic Regression
Dataset 1 (RMSE)	**0,22**	0,55	0,32	-
Dataset 2 (Accuracy)	**0,97**	-	-	**0,97**
Dataset 3 (RMSE)	**0,31**	2,32	4,11	-

Table 2
Preliminary results of the prediction models

Given the results of the regression data sets, an advantage of the transformer architecture can be seen with the used metrics. However, the classification results show an almost equal accuracy for both results, which could be due to class imbalances in predicting anomalies, i.e., the prediction of sensor failures. For the final results, this should be extended to use F-Scores, which as the harmonic mean of precision and recall, are insensitive to such imbalances [37].

7. Discussion

In the following sections, the preliminary results reached during this research project will be summarized and discussed in the context of IoT devices. For this, we will first focus on the results reached during the development and evaluation and then progress to known and possible limitations as well as implications of the work.

7.1. Contribution

Since this research is ongoing, the results for all algorithm and data set pairings, and all fully optimized models are not yet available. However, the results until now, as displayed in Table 2, already show significant improvements in the transformer model. Apart from the classification, where possible due to the data set used, the statistical method delivers similar results, and the transformer consistently outperforms the alternative methods. This roughly replicates the results from, e.g., Zhou et al. [33], [38], who also found significant benefits in transformers compared to LSTMs. In comparison with the transformers, the LSTMs are also significantly slower. For the same number of training epochs, an average increase of 124% was measured across all data sets. The benefits of neural networks compared to ARIMA models were demonstrated in several studies [16, 24, 25] before and confirmed.

This leads to the conclusion that transformers and their defining attention mechanism offer a few decisive advantages for the project task, i.e., the prediction of IoT device-related data. By weighted consideration of all inputs for each output, specific data patterns can be addressed very well. The parallelization avoids the long gradient paths of deep LSTMs [26], making the attention mechanism particularly suitable for the desired predictions. Other central advantages of the transformers besides the more efficient training that has not yet been explored in this project include the possible use of transfer learning [39]. By using transfer learning, the effort needed to train specific models might again be drastically reduced.

7.2. Limitations

This work is subject to a few limitations, mainly the current focus on proprietary data sets. While necessary for the project partner, this only allows for limited comparison of the results to baseline solutions in the field. Despite the findings of Wen et al. [22], no seasonal trend decomposition is used on the data, which might allow for even better results with the transformer. Additionally, further extensive optimization of all model hyperparameters might be applied to the neural networks, given additional time and resources. The transformer architecture used in this project is currently quite standard and, like most neural networks, a black box model. Other transformer architectures for time series, like the Temporal Fusion Transformer [40] might give additional insight into the data.

Specifically for the domain of IoT device data, where new data points are generated frequently and concept drifts [41], [42] as location changes of sensors might occur regularly, a model update strategy might be necessary. Retraining and continuous online learning is an ongoing problem in machine learning and might apply specifically to IoT data. However, this has not been in the scope of this project.

7.3. **Implications**

The proposed deep learning-based software system for the integration of IoT devices into configuration management has several important implications for small and medium-sized companies. First, the system offers an innovative approach to managing IoT devices, which are increasingly important for business operations. Transformer neural networks represent an improvement over traditional machine learning methods, as they can handle long series and complex dependencies more effectively.

Second, the proposed system can potentially improve the accuracy and efficiency of configuration management. The use of transformer-based event prediction is expected to significantly improve forecasting accuracy and management efficiency, which can result in significant cost savings and increased productivity for small and medium-sized companies.

Third, the project's use of the Design Science Research method demonstrates a rigorous and structured approach to developing and evaluating the proposed system. This method ensures that the system meets the requirements and addresses the problem of efficient and effective management of IoT devices in small and medium-sized companies.

8. Conclusion

Overall, the proposed system has significant implications for the management of IoT devices in small and medium-sized companies. The system can potentially improve business operations and increase profitability by providing a more efficient and effective way of managing IoT devices. As IoT technology continues to grow, the development of such systems will become increasingly important for companies of all sizes.

References

[1] R. Schmidt, M. Möhring, R.-C. Härting, C. Reichstein, P. Neumaier, and P. Jozinović, “Industry 4.0 - Potentials for Creating Smart Products: Empirical Research Results,” in International Conference on Business Information Systems, Cham, Jun. 2015, pp. 16–27.

[2] “Gartner Says 5.8 Billion Enterprise and Automotive IoT Endpoints Will Be in Use in 2020,” Gartner. https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io (accessed Jun. 23, 2020).

[3] M. Brenner, M. Garschhammer, M. Sailer, and T. Schaaf, “CMDB-Yet Another MIB? On Reusing Management Model Concepts in ITIL Configuration Management,” Large Scale Management of Distributed Systems, pp. 269–280, 2006.
[4] P. Johannesson and E. Perjons, An Introduction to Design Science, 2nd ed. Cham: Springer, 2021. doi: 10.1007/978-3-030-78132-3.
[5] A. Vaswani et al., “Attention is All you Need,” in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, Dec. 2017, pp. 5998–6008. Accessed: Feb. 11, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
[6] T. Taneja, A. Jatain, and S. B. Bajaj, “Predictive analytics on IoT,” in 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, May 2017, pp. 1312–1317. doi: 10.1109/CCAA.2017.8230000.
[7] M. Marjani et al., “Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges,” IEEE Access, vol. 5, pp. 5247–5261, 2017.
[8] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, “Deep learning for IoT big data and streaming analytics: A survey,” IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2923–2960, 2018.
[9] A. Akbar, A. Khan, F. Carrez, and K. Moessner, “Predictive Analytics for Complex IoT Data Streams,” IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1571–1582, Oct. 2017, doi: 10.1109/JIOT.2017.2712672.
[10] A. Akbar, F. Carrez, K. Moessner, and A. Zoha, “Predicting complex events for pro-active IoT applications,” in 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Dec. 2015, pp. 327–332. doi: 10.1109/WF-IoT.2015.7389075.
[11] S. Verma, Y. Kawamoto, Z. Md. Fadlullah, H. Nishiyama, and N. Kato, “A Survey on Network Methodologies for Real-Time Analytics of Massive IoT Data and Open Research Issues,” IEEE Commun. Surv. Tutorials, vol. 19, no. 3, pp. 1457–1477, 2017, doi: 10.1109/COMST.2017.2694469.
[12] B. Chen, J. Wan, A. Celesti, D. Li, H. Abbas, and Q. Zhang, “Edge computing in IoT-based manufacturing,” IEEE Communications Magazine, vol. 56, no. 9, pp. 103–109, 2018.
[13] H. Li, K. Ota, and M. Dong, “Learning IoT in edge: Deep learning for the Internet of Things with edge computing,” IEEE network, vol. 32, no. 1, pp. 96–101, 2018.
[14] Y. He, J. Guo, and X. Zheng, “From surveillance to digital twin: Challenges and recent advances of signal processing for industrial internet of things,” IEEE Signal Processing Magazine, vol. 35, no. 5, pp. 120–129, 2018.
[15] S. O. Erikstad, “Merging physics, big data analytics and simulation for the next-generation digital twins,” Hiper, no. September, pp. 139–149, 2017.
[16] V. R. Prybutok, J. Yi, and D. Mitchell, “Comparison of neural network models with ARIMA and regression models for prediction of Houston’s daily maximum ozone concentrations,” European Journal of Operational Research, vol. 122, no. 1, pp. 31–40, 2000.
[17] Z. Liu, Z. Zhu, J. Gao, and C. Xu, “Forecast Methods for Time Series Data: A Survey,” IEEE Access, vol. 9, pp. 91896–91912, 2021, doi: 10.1109/ACCESS.2021.3091162.
[18] J. Hirschle, Machine Learning für Zeitreihen: Einstieg in Regressions-, ARIMA- und Deep-Learning-Verfahren mit Python. München: Hanser, 2021.

[19] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[20] B. Lim and S. Zohren, “Time Series Forecasting With Deep Learning: A Survey,” arXiv preprint arXiv:2004.13408, 2020.
[21] T. Lin, Y. Wang, X. Liu, and X. Qiu, “A Survey of Transformers,” AI Open, vol. 3, pp. 111–132, 2022, doi: 10.1016/j.aiopen.2022.10.001.
[22] Q. Wen et al., “Transformers in Time Series: A Survey.” arXiv, Mar. 07, 2022. Accessed: May 26, 2022. [Online]. Available: http://arxiv.org/abs/2202.07125
[23] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, “Learning Graph Structures With Transformer for Multivariate Time-Series Anomaly Detection in IoT,” IEEE Internet Things J., vol. 9, no. 12, pp. 9179–9189, Jun. 2022, doi: 10.1109/JIOT.2021.3100509.
[24] D. Janardhanan and E. Barrett, “CPU workload forecasting of machines in data centers using LSTM recurrent neural networks and ARIMA models,” in 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), 2017, pp. 55–60.
[25] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “A comparison of ARIMA and LSTM in forecasting time series,” in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 1394–1401.
[26] G. E. Box, G. M. Jenkins, and G. C. Reinsel, Time series analysis: forecasting and control. Wiley. com, 2013. Accessed: Sep. 23, 2013. [Online]. Available: http://books.google.de/books?hl=de&lr=&id=jyrCqMBW_owC&oi=fnd&pg=PP1&dq=Time+series+analysis:+Forecasting+and+control&ots=LPJX9hDKM-&sig=_fjPpfKeBK7KpwraCnlOZui42tA
[27] W. Wang and D. Guo, “Towards unified heterogeneous event processing for the Internet of Things,” in 2012 3rd IEEE International Conference on the Internet of Things, Wuxi, Jiangsu Province, China, Oct. 2012, pp. 84–91. doi: 10.1109/IOT.2012.6402308.
[28] S. Raschka, “Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning.” arXiv, Nov. 10, 2020. Accessed: Aug. 29, 2022. [Online]. Available: http://arxiv.org/abs/1811.12808
[29] C. Bergmeir and J. M. Benítez, “On the use of cross-validation for time series predictor evaluation,” Information Sciences, vol. 191, pp. 192–213, May 2012, doi: 10.1016/j.ins.2011.12.028.
[30] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” arXiv:1412.6980 [cs], Jan. 2017, Accessed: Jan. 14, 2022. [Online]. Available: http://arxiv.org/abs/1412.6980
[31] S. Ruder, “An overview of gradient descent optimization algorithms,” arXiv:1609.04747 [cs], Jun. 2017, Accessed: Jan. 14, 2022. [Online]. Available: http://arxiv.org/abs/1609.04747
[32] N. Wu, B. Green, X. Ben, and S. O’Banion, “Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case.” arXiv, Jan. 22, 2020. Accessed: Feb. 15, 2023. [Online]. Available: http://arxiv.org/abs/2001.08317
[33] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, “FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting,” in Proceedings of the 39th International Conference on Machine Learning, Honolulu, HI, USA, Jun. 2022, pp.

27268–27286. Accessed: Dec. 25, 2022. [Online]. Available: https://proceedings.mlr.press/v162/zhou22g.html
[34] H. Wu, J. Xu, J. Wang, and M. Long, “Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting,” in Advances in Neural Information Processing Systems, Virtual, 2021, vol. 34, pp. 22419–22430. doi: 9781713845393.
[35] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature,” Geosci. Model Dev., vol. 7, no. 3, pp. 1247–1250, Jun. 2014, doi: 10.5194/gmd-7-1247-2014.
[36] A. Botchkarev, “A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms,” IJIKM, vol. 14, pp. 045–076, 2019, doi: 10.28945/4184.
[37] M. Sokolova, N. Japkowicz, and S. Szpakowicz, “Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation,” in AI 2006: Advances in Artificial Intelligence, Berlin, Germany, 2006, vol. 4304, pp. 1015–1021. doi: 10.1007/11941439_114.
[38] H. Zhou et al., “Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting,” in Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, Feb. 2021, vol. 35, pp. 11106–11115. doi: 10.1609/aaai.v35i12.17325.
[39] T. Huang, P. Chen, J. Zhang, R. Li, and R. Wang, “A Transferable Time Series Forecasting Service Using Deep Transformer Model for Online Systems,” in 37th IEEE/ACM International Conference on Automated Software Engineering, Rochester, MI, USA, Oct. 2022, pp. 1–12. doi: 10.1145/3551349.3560414.
[40] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, “Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting.” arXiv, Sep. 27, 2020. Accessed: Feb. 14, 2023. [Online]. Available: http://arxiv.org/abs/1912.09363
[41] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, “Learning under Concept Drift: A Review,” IEEE Trans. Knowl. Data Eng., pp. 1–1, 2018, doi: 10.1109/TKDE.2018.2876857.
[42] M. Lima, M. Neto, T. S. Filho, and R. A. de A. Fagundes, “Learning Under Concept Drift for Regression—A Systematic Literature Review,” IEEE Access, vol. 10, pp. 45410–45429, 2022, doi: 10.1109/ACCESS.2022.3169785.