Energy Usage prediction in Industry 4.0 with Transformer Networks

1st Hummel Martin
Computer Science
Hochschule Furtwangen
Furtwangen, Deutschland

***Abstract*—Dieser Beitrag untersucht den Einsatz von Transformer-basierten Deep-Learning-Modellen zur Energieverbrauchsprognose in Industrie-4.0-Umgebungen. Aufbauend auf einem Datensatz stündlicher Messungen eines US-amerikanischen Energieversorgers demonstriert das vorgestellte Modell mithilfe von Mechanismen wie Multi-Head Attention und Encoder-Decoder-Architekturen eine hohe Vorhersagegenauigkeit (MAPE von 0,06%). Trotz dieser vielversprechenden Ergebnisse weist die Studie auf Herausforderungen bei der Übertragung in reale Produktionsszenarien hin, insbesondere hinsichtlich heterogener Datenerfassung, Echtzeitanforderungen und fehlender Standards für Datenqualität. Die Ergebnisse unterstreichen das Potenzial von Transformer-Ansätzen, liefern jedoch gleichzeitig Anhaltspunkte für weitere Forschung, etwa zur Skalierbarkeit und kontinuierlichen Anpassung an sich dynamisch ändernde Produktionsumgebungen.**

***Index Terms*—Energy Forecasting, Transformer Networks, Industry 4.0, IoT, Time Series Analysis**

I. Introduction

Ein zentraler Faktor, der die steigende Bedeutung vorausschauender Modellierung und Optimierung des Energieverbrauchs in der Industrie verdeutlicht, ist der langjährige Anstieg des Gesamtenergiebedarfs. So zeigen Daten der *U.S. Energy Information Administration (EIA)*, dass der Energieeinsatz in der US-amerikanischen Industrie zwischen 1950 und 1970 von insgesamt rund 14 Quadrillion Btu (etwa 15 Exajoule) auf knapp 25 Quadrillion Btu (etwa 26 Exajoule) gestiegen ist.

Während in den 1950er-Jahren noch etwa 40 % der eingesetzten Energie aus Kohle stammten, reduzierte sich ihr Anteil bis 1970 auf unter 20 %. Im selben Zeitraum nahm Erdgas von rund 25 % auf knapp 40 % zu, und auch Petroleum wuchs von etwa 27 % auf gut 30 %.

Für das Jahr 2022 weisen die EIA-Daten einen industriellen Gesamtenergieverbrauch von knapp 27 Quadrillion Btu (zirka 28 Exajoule) aus. Rund 41 % davon entfallen auf Erdgas und 34 % auf Petroleum. Elektrizität macht etwa 13 % aus, erneuerbare Energien circa 9 %, und Kohle liegt bei knapp 4 % [1].

Vor diesem Hintergrund gewinnt Industrie 4.0 zunehmend an Bedeutung: Sie kennzeichnet auf eine hochflexible, auf Nachhaltigkeit ausgerichtete Produktionsweise. IoT (Internet of Things) ermöglicht hierbei, Energieverbrauchsdaten mithilfe von Sensorik und Smart Meters kontinuierlich zu erfassen und in Echtzeit zu analysieren. Exakte Prognosen und fortlaufende Optimierungen des Verbrauchs sind somit in hochautomatisierten Industrie-4.0-Umgebungen unverzichtbar, um sowohl die Kosten als auch die Umweltbelastung dauerhaft zu senken [2].

A. Herausforderungen

In hochautomatisierten Industrie-4.0-Umgebungen entstehen durch zahlreiche IoT-Sensoren umfangreiche Datenströme, deren gleichzeitige Erfassung und Analyse hohe Anforderungen an Speicher- und Verarbeitungskapazitäten stellt [3]. Hinzu kommt die Herausforderung, Daten aus unterschiedlichsten Quellen und Formaten – etwa von Maschinensteuerungen, Sensorclustern oder externen Systemen – in ein einheitliches Schema zu überführen, was flexible Datenmodelle voraussetzt [4]. Echtzeit-Anforderungen bilden dabei einen weiteren Engpass, da die Datenauswertung nahezu sofort erfolgen muss, um Ausfälle oder Qualitätsprobleme schnell zu erkennen und umgehend beheben zu können [5]. Die Kombination aus großen Datenmengen, Sensorheterogenität und Zeitdruck macht skalierbare Plattformen sowie robuste Analytikverfahren erforderlich, um Vorhersagen präzise zu treffen und rechtzeitig geeignete Maßnahmen einzuleiten. Hier setzt maschinelles Lernen, insbesondere Deep Learning, an. Es besitzt die Fähigkeit, aus großen, komplexen Datenmengen Muster und Zusammenhänge zu extrahieren, die durch IoT-Sensoren und andere Quellen generiert werden. Die Echtzeitanalyse industrieller Prozesse eröffnet zudem konkrete Lösungen für Herausforderungen wie prädiktive Wartung und die Optimierung von Energieverbrauchsprognosen [6].

Transformer-basierte Modelle, die ursprünglich für Sequenzdaten entwickelt wurden. Die Analyse von Zeitreihen erweist sich als ausgesprochen hilfreich, um zeitabhängige Strukturen wie Saisonalitäten und Trends zu erkennen. Mechanismen wie Masked Autoencoders und kontrastives Lernen werden zur verbesserten Repräsentation und Vorhersage von Zeitreihen eingesetzt. Durch die Kombination dieser Ansätze lassen sich relevante Muster wie Saisonalitäten und Trends präziser erfassen [7].

Im folgenden Abschnitt II wird der aktuelle Forschungsstand zusammengefasst. Anschließend beschreibt Abschnitt III

die Modellarchitektur, bevor Abschnitt IV die Ergebnisse präsentiert und diskutiert. Den Abschluss bildet Abschnitt V mit der Schlussfolgerung.

II. RELATED WORK

Energieverbrauchsprognosen stellen eine essenzielle Grundlage dar, um sowohl betriebswirtschaftliche als auch ökologische Ziele zu erreichen. In jüngerer Zeit haben sich dabei unterschiedliche Ansätze herauskristallisiert. Während statistische Modelle wie ARIMA oder SARIMA in manchen Fällen bereits eine solide Vorhersagegüte bieten, stoßen sie bei stark nichtlinearen Mustern oft an ihre Grenzen [8].

Mehrere der untersuchten Studien setzen auf Machine-Learning-Methoden wie Random Forest (RF), Support Vector Regression (SVR) oder auch Boosting-Ansätze (z. B. AdaBoost), um komplexe und dynamische Lastprofile zu modellieren. Random Forest wird dabei oft wegen seiner Ensemble-Struktur aus Entscheidungsbäumen eingesetzt, was eine vergleichsweise robuste Vorhersage bei kurz- bis mittelfristigen Prognosen ermöglicht [9]. Gleichzeitig zeigen SVR-basierte Ansätze ihr Potenzial bei hochdimensionalen Daten, erfordern jedoch gewissen Aufwand bei der Parametrierung des Kernels [10].

Als besonders leistungsfähig haben sich in den letzten Jahren Deep-Learning-Modelle erwiesen [11]. LSTM-Netze (Long Short-Term Memory) können beispielsweise langanhaltende Abhängigkeiten innerhalb von Zeitreihen erfassen und eignen sich daher sehr gut für (kurz- bis) mittelfristige Prognosen, in denen sowohl saisonale Effekte als auch plötzliche Schwankungen eine Rolle spielen [8]. In neueren Untersuchungen werden darüber hinaus CNN-basierte (Convolutional Neural Network) Modelle oder sogar hybride Ansätze aus CNN und LSTM eingesetzt, um sowohl lokale Muster (durch Faltungsschichten) als auch zeitliche Abhängigkeiten (durch rekurrente Schichten) zu erkennen [9].

Ein weiteres Merkmal aktueller Forschung ist die vermehrte Nutzung IoT-Plattformen (Internet of Things), über die eine Echtzeit-Datenerfassung und -verarbeitung erfolgen kann. Das ist besonders für kurzfristige Lastprognosen und Anomalieerkennungen von Vorteil [10]. Hinsichtlich der Modellgüte hat sich gezeigt, dass Ensemble- und Hybridmodelle häufig bessere Vorhersageergebnisse erzielen als reine Einzelmodelle [10]. Beispielsweise kombinieren einige Arbeiten ein Vorverarbeitungsmodell wie Wavelet-Transformation oder PCA (Principal Component Analysis) mit einem anschließenden LSTM-Netz, um sowohl Rauschen zu reduzieren als auch nichtlineare Beziehungen zu erkennen. Auch Random Forest in Verbindung mit Feature-Engineering oder Support Vector Machines (SVM) mit optimierten Kernel-Parametern sorgen für eine robustere Performance [9].

Insgesamt hängt die Wahl der Vorhersagemethode maßgeblich von Faktoren wie Datenverfügbarkeit, Vorhersagehorizont und der benötigten Granularität ab. Statistische Methoden können bei kleinen Datensätzen oder relativ stabilen Lastprofilen weiterhin sinnvoll sein, während Machine-Learning- und Deep-Learning-Verfahren ihre Stärken bei hochfrequenten oder stark variierenden Lastkurven ausspielen. Für die Praxis ist zudem relevant, dass die Resultate reproduzierbar sind und die eingesetzten Modelle kontinuierlich an neue Gegebenheiten (z. B. technische Innovationen, Wetterextreme oder wirtschaftliche Veränderungen) angepasst werden können [11].

Angesichts dieser bereits sehr leistungsfähigen Ansätze rücken jedoch zunehmend Transformer-Modelle in den Fokus. Transformermodelle finden in jüngster Zeit große Beachtung bei der Prognose von Energieverbrauchszeitreihen. Ein Ansatz aus der Literatur beschreibt ein multivariates Transformerkonzept mit erweiterter Aufmerksamkeit für verschiedene Eingangsvariablen, das eine robuste Prognosegüte erreicht und klassische Rekurrente Netze übertrifft [12]. Ein weiterer Vergleich mit LSTM-Modellen zeigt, dass Transformer mindestens gleichwertig oder sogar besser abschneiden und zudem längere Zeithorizonte effizienter verarbeiten können [13]. Darüber hinaus existiert eine Erweiterung, die explizite Zeitembeddings (Time2Vec) nutzt, um insbesondere bei monatlichen Verbrauchsdaten in China die Prognosegenauigkeit deutlich zu steigern [14].

III. PRELIMINARIES

In einer Smart Factory ermöglichen Smart Meters und Sensoren die Erfassung von Echtzeitdaten über den Energieverbrauch auf Maschinen- und Produktionslinienebene, einschließlich aktiver und reaktiver Leistung. Diese Daten werden mithilfe von IoT-Technologie und Cloud-Plattformen gesammelt, analysiert und in das Produktionsmanagement integriert, um Energieeffizienz zu verbessern. Durch die Verbindung mit Smart Grids können energieintensive Prozesse auf kostengünstigere Zeiten verlagert werden, wodurch Produktionskosten gesenkt und Energieverschwendung minimiert werden [2]. Um die in einer Smart Factory generierten Energieverbrauchsdaten effizient zu analysieren und zukünftige Verbrauchsmuster vorherzusagen, können wie in den Arbeiten [12],[13],[14] moderne Machine-Learning-Modelle wie den Transformer-Netzwerken eingesetzt werden.

A. Transformer Networks

Der Transformer ist ein vollständig auf Attention-Mechanismen beruhendes Sequenz-zu-Sequenz-Modell, das konsequent auf rekurrente und konvolutionale Schichten verzichtet. Es nutzt eine Encoder-Decoder-Architektur, bei der der Encoder eine Eingabesequenz $(x_1, \ldots, x_n)$ in eine Reihe kontinuierlicher Repräsentationen $(z_1, \ldots, z_n)$ überführt, während der Decoder diese Repräsentationen schrittweise in die Ausgabesequenz $(y_1, \ldots, y_m)$ umwandelt. Grundlage jeder Schicht ist die *Multi-Head Self-Attention*, bei der alle Positionen gleichzeitig betrachtet werden. Dazu werden pro Schicht die Queries (Q), Keys (K) und Values

(V) gebildet, um mithilfe einer *Scaled Dot-Product Attention* die wichtigsten Informationen im Kontext zu identifizieren:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\Big(\frac{Q\,K^\top}{\sqrt{d_k}}\Big)\,V, \qquad (1)$$

wobei d_k die Dimension der Key- und Query-Vektoren ist. Durch *Multi-Head*-Aufspaltungen paralleler "Köpfe" werden unterschiedliche Aspekte der Eingabe gleichzeitig erfasst.

Der Decoder erweitert dieses Prinzip um eine *Encoder-Decoder-Attention*: Die Queries stammen aus der vorherigen Decoder-Schicht, während Keys und Values aus dem Encoder-Ausgang $(z_1, \ldots, z_n)$ entnommen werden. Zusätzlich maskiert der Decoder künftige Tokens ("Masking"), um nur bereits bekannte Ausgaben im autoregressiven Prozess zu berücksichtigen. Da der Transformer keine rekurrenten oder konvolutionalen Schritte enthält, werden *Positional Encodings* addiert, damit die Modellschichten Informationen über Positionsabstände der Tokens erhalten.

Jede Schicht umfasst überdies ein *Feed-Forward-Netz*, das jede Position separat verarbeitet

$$\mathrm{FFN}(x) = \max(0,\, x\,W_1 + b_1)\,W_2 + b_2. \qquad (2)$$

Residual-Verbindungen und Layer Normalization umrahmen diese Blöcke, um Stabilität und bessere Konvergenz zu gewährleisten [15].

B. Self-Attention Mechanism

Der Self-Attention-Mechanismus (häufig auch Intra-Attention" genannt) beschreibt in neuronalen Netzen ein Verfahren, bei dem jedes Positionselement einer Sequenz eine gewichtete Summe über alle Elemente derselben Sequenz bildet. Formal ordnet man jeder Position i drei Vektoren zu: Q_i (*Query*), K_i (*Key*) und V_i (*Value*). Die Aufmerksamkeit (*Attention*) für Position i gegenüber einer anderen Position j wird durch den Wert

$$\mathrm{Attention}(i, j) = \mathrm{softmax}\Big(\frac{Q_i \cdot K_j^\top}{\sqrt{d_k}}\Big) \qquad (3)$$

bestimmt, wobei d_k die Dimension der Key- und Query-Vektoren ist. Über alle Positionen wird eine gewichtete Summe der Value-Vektoren $\{V_j\}$ gebildet, sodass jedes *Token* im Satz lernt", wie stark es mit jedem anderen Token verknüpft ist. Im Vergleich zu rein rekurrenten Netzen (z. B. LSTM) oder konvolutionalen Verfahren können mit Self-Attention selbst weit auseinanderliegende Elemente in einem Schritt berücksichtigt werden. Dadurch wird die Berechnung hochparallelisierbar und kontextuelle Abhängigkeiten lassen sich besonders effizient erfassen [15].

IV. MAIN PART

Die Prognose des Energieverbrauchs ist angesichts steigender Elektrifizierung und wachsender Volatilität im Energiesektor besonders bedeutsam [1]. Genaue Verbrauchsvorhersagen ermöglichen eine effiziente Kapazitätsplanung, unterstützen das Management von Lastspitzen und tragen dazu bei, wirtschaftliche sowie ökologische Ziele miteinander zu verbinden [16].

Vor diesem Hintergrund wird im Folgenden exemplarisch ein Modell zur Energieverbrauchsprognose in Industrie-4.0-Umgebungen vorgestellt, das auf Transformer-Netzwerken basiert. Da die dafür erforderlichen Datensätze aus realen Industrie-4.0-Anwendungen nicht öffentlich zugänglich sind, wird stattdessen ein Datensatz der *American Electric Power Company* genutzt, der stündliche Messungen zum Energieverbrauch in Megawattstunden (MWh) für den Zeitraum von 2014 bis 2018 enthält [17].

A. Data Set and Implementation Environment

Zur Entwicklung und Ausführung des vorgestellten Transformer-Netzwerks wurden zahlreiche Python-Bibliotheken genutzt, darunter `torch`, `transformers`, `keras` und verschiedene weitere Abhängigkeiten. Durch den Einsatz der Vorab-Version (*nightly build*) von `torch` in Kombination mit `torchvision` und `torchaudio` lassen sich neueste Funktionen der PyTorch-Plattform nutzen. Da auf dem verwendeten macOS-System der `mps`-Support aktiviert ist, kann die GPU-Beschleunigung über die Metal Performance Shaders genutzt werden.

Hardware und Betriebssystem Das Projekt wurde auf einem *Apple M4* System mit den in Tabelle I aufgeführten Spezifikationen durchgeführt.

TABLE I
SYSTEMKONFIGURATION

Komponente	Beschreibung
Prozessor	Apple M4 Chip
CPU	10-Core CPU
GPU	10-Core GPU
Neural Engine	16-Core Neural Engine
Arbeitsspeicher	32 GB gemeinsamer Speicher
Betriebssystem	macOS Sequoia 15.1

B. Transformer Model and Hyperparameters

Ein zentrales Element des hier vorgestellten Transformer-basierten Modells sind verschiedene Hyperparameter, die dessen Lernverhalten steuern und den Umfang der Trainings- und Vorhersagefenster bestimmen. Mithilfe von `input_window = 100` wird festgelegt, dass das Modell jeweils die letzten 100 Zeitschritte als Eingabe (Vergangenheit) erhält, um daraus seine Prognosen abzuleiten. Die Anzahl der vorherzusagenden Zeitschritte wird durch `output_window = 24` definiert, sodass das Modell die nächsten 24 Stunden vorhersagt. Die Trainings-Batches umfassen dabei je 32 Sequenzen (`batch_size = 32`), was bedeutet, dass in jedem Trainingsschritt 32 Eingabe-Ausgabe-Paare verarbeitet werden, bevor eine Aktualisierung der Gewichte erfolgt. In den fünf Trainingsdurchläufen (`epochs = 5`) wird der gesamte Datensatz wiederholt durchlaufen, sodass das Modell

die zugrunde liegenden Muster mehrfach erlernen kann.

Die Lernrate (`lr = 1e-3`) legt fest, in welcher Schrittweite die Gewichte des Modells im Laufe des Trainings angepasst werden. Darüber hinaus wird die Dimensionalität des Transformer-Encoders durch `d_model = 32` gesteuert, indem die Eingabe auf 32 Dimensionen projiziert wird. Gleichzeitig ist `nhead = 4` so gewählt, dass der *Multi-Head Attention*-Mechanismus die Eingabe parallel in vier Köpfen“ verarbeitet, während `num_layers = 2` eine Tiefe von zwei aufeinanderfolgenden Encoder-Blöcken bereitstellt. Das Modell nutzt zudem eine Dropout-Rate von `dropout = 0.1`, um Overfitting zu reduzieren. Abschließend ermittelt das Skript selbstständig, ob die Nutzung von `mps` (Metal Performance Shaders) auf `macOS` verfügbar ist; falls ja, erfolgt das Training auf dem Apple-GPU-Backend, ansonsten kommt die CPU zum Einsatz.

C. Results and Conclusion

Im folgenden Abschnitt wird die Güte der Modellanpassung anhand des Verlaufes der Fehlermaße im Training und in der Validierung illustriert. Abbildung 1 stellt diesen zeitlichen Verlauf dar, wobei eine kontinuierliche Abnahme beider Kurven auf eine fortschreitende Konvergenz des Modells schließen lässt.

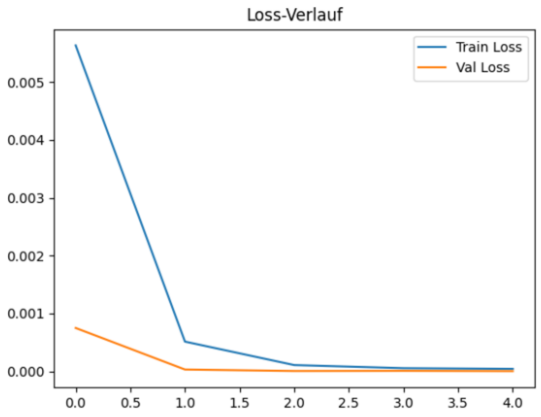

Fig. 1. Verlauf des Trainings- und Validierungsverlustes.

Die Abnahme beider Kurven verdeutlicht, dass das Modell mit fortschreitenden Trainingszyklen zunehmend präzise Vorhersagen trifft. Das kontinuierliche Sinken des Validierungsverlustes legt zudem nahe, dass das Modell nicht nur die Trainingsdaten gut repräsentiert, sondern auch für bisher unbekannte Daten verlässliche Prognosen liefert.

Bei der anschließenden Bewertung von Vorhersagemodellen kommen häufig etablierte Metriken zur Fehlerbestimmung zum Einsatz. Dazu gehören:

- **MAE (Mean Absolute Error)** Ermittelt die durchschnittliche absolute Abweichung zwischen vorhergesagtem und tatsächlichem Wert.
- **RMSE (Root Mean Square Error)** Misst die quadratische Abweichung, wodurch größere Fehler stärker gewichtet werden als bei der MAE.
- **MAPE (Mean Absolute Percentage Error)** Gibt den prozentualen Fehler an und ermöglicht dadurch einen relativen Vergleich verschiedener Prognosen.

Diese Kennzahlen erfassen sowohl absolute als auch relative Fehlergrößen und liefern somit ein umfassendes Bild der Vorhersagegüte. Auf deren Basis lassen sich fundierte Rückschlüsse auf die Modellqualität ziehen und gegebenenfalls Anpassungen bei Hyperparametern oder der Modellauswahl vornehmen [18]. Entsprechende Rückschlüsse auf die Modellgüte können folglich gezogen und gegebenenfalls Optimierungen am Modell (z. B. Anpassung von Hyperparametern oder Modellauswahl) vorgenommen werden.

Die abschließende Auswertung des Testdatensatzes, in dem die stündlichen Messwerte im Bereich von etwa 18,000 bis 22,000 MW liegen, weist auf eine hohe Modellgüte hin: Konkret wurden ein MSE von 157, ein RMSE von rund 12,5, ein MAE von etwa 9,2 und ein MAPE von nur 0,06 % ermittelt. Angesichts des hohen absoluten Lastniveaus zeigt sich somit eine äußerst geringe Abweichung zwischen den modellbasierten Prognosen und den tatsächlichen Messwerten, was auf ein präzises und robustes Vorhersageverhalten schließen lässt. Abbildung 2 illustriert hierzu den Vergleich zwischen echten Messwerten und durch das Modell erzeugten Vorhersagen für die letzten 200 Stichproben. Die nahezu deckungsgleichen Verläufe beider Kurven verdeutlichen, dass das Modell den tageszeitabhängigen Zyklus der Energieabnahme präzise abbildet und auf hohem Lastniveau zuverlässige Prognosen liefert. Diese visuell erkennbar hohe Vorhersagegüte deckt sich mit den zuvor vorgestellten Fehlerkennzahlen, die trotz eines Verbrauchsbereichs von 18,000 MW bis 22,000 MW sehr niedrige Abweichungen ausweisen.

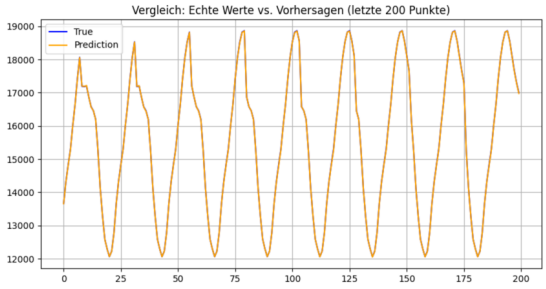

Fig. 2. Vergleich zwischen echten Messwerten und Modellprognosen (letzte 200 Stichproben).

Das in den vorherigen Abschnitten beschriebene Transformer-Netzwerk, einschließlich der zugrunde liegenden

Datensätze sowie der verwendeten Evaluationsroutinen, ist öffentlich auf GitHub verfügbar. Unter folgendem Link können sämtliche Implementierungsdetails, Konfigurationsdateien und Trainingsprotokolle eingesehen und nachvollzogen werden: https://github.com/Maddi02/EnergyPrediction-Tranformer-Network

V. CONCLUSION

Das vorgestellte Transformer-basierte Modell zeigt die Möglichkeit, Energieverbrauchszeitreihen präzise vorherzusagen, indem Mechanismen wie Multi-Head Attention und Encoder-Decoder-Architekturen genutzt werden. Mit einem MAPE von 0,06% auf einem Testdatensatz liefert das Modell nachvollziehbare Ergebnisse. Es bleibt jedoch zu berücksichtigen, dass der verwendete Datensatz nicht direkt auf Industrie-4.0-Umgebungen übertragbar ist, da reale Produktionsdaten häufig komplexere und variierende Muster aufweisen, die spezifische Daten erfordern. Die Literatur zeigt, dass Transformer-Modelle bereits erfolgreich für Energieverbrauchsprognosen eingesetzt werden [12],[13],[14]. Gleichzeitig bleiben wichtige Herausforderungen unberücksichtigt, wie die Verarbeitung heterogener und unvollständiger Daten, Echtzeitanforderungen, externe Einflussfaktoren und die Integration in bestehende industrielle Systeme. Zukünftige Arbeiten sollten sich auf die Nutzung besser geeigneter Datensätze, eine höhere Skalierbarkeit und eine kontinuierliche Anpassbarkeit der Modelle konzentrieren. Das Modell zeigt dennoch, dass Transformer-Ansätze eine praktikable Grundlage für Energieverbrauchsprognosen bieten und in weiteren Forschungen vertieft werden können.

REFERENCES

[1] U.S. Energy Information Administration, *Monthly Energy Review: December 2024*, Released December 23, 2024, Washington, DC, Dec. 2024. [Online]. Available: https://www.eia.gov/totalenergy/data/monthly.

[2] F. Shrouf, J. Ordieres, and G. Miragliotta, "Smart factories in industry 4.0: A review of the concept and of energy management approached in production based on the internet of things paradigm," in *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, 2014, pp. 697–701. DOI: 10.1109/IEEM.2014.7058728.

[3] S. Ren, J.-S. Kim, W.-S. Cho, S. Soeng, S. Kong, and K.-H. Lee, "Big data platform for intelligence industrial iot sensor monitoring system based on edge computing and ai," in *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2021, pp. 480–482. DOI: 10.1109/ICAIIC51459.2021.9415189.

[4] X. Xu and Q. Hua, "Industrial big data analysis in smart factory: Current status and research strategies," *IEEE Access*, vol. 5, pp. 17 543–17 551, 2017. DOI: 10.1109/ACCESS.2017.2741105.

[5] S. K. Jagatheesaperumal, M. Rahouti, K. Ahmad, A. Al-Fuqaha, and M. Guizani, "The duo of artificial intelligence and big data for industry 4.0: Applications, techniques, challenges, and future research directions," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 12 861–12 885, 2022. DOI: 10.1109/JIOT.2021.3139827.

[6] H. Wang, W. Zhang, D. Yang, and Y. Xiang, "Deep-learning-enabled predictive maintenance in industrial internet of things: Methods, applications, and challenges," *IEEE Systems Journal*, vol. 17, no. 2, pp. 2602–2615, 2023. DOI: 10.1109/JSYST.2022.3193200.

[7] Y. E. Midilli and S. Parshutin, "A review for pre-trained transformer-based time series forecasting models," in *2023 IEEE 64th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS)*, 2023, pp. 1–8. DOI: 10.1109/ITMS59786.2023.10317721.

[8] R. Rane, M. Desai, A. Pandey, and F. Kazi, "Energy consumption forecasting using a deep learning energy-level based prediction," in *2021 IEEE International Power and Renewable Energy Conference (IPRECON)*, 2021, pp. 1–6. DOI: 10.1109/IPRECON52453.2021.9640748.

[9] S. S. Arnob, A. I. M. S. Arefin, A. Y. Saber, and K. A. Mamun, "Energy demand forecasting and optimizing electric systems for developing countries," *IEEE Access*, vol. 11, pp. 39 751–39 775, 2023. DOI: 10.1109/ACCESS.2023.3250110.

[10] S. Balaji and S. Karthik, "Energy prediction system using internet of things," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 1131–1135. DOI: 10.1109/ICACCS48705.2020.9074299.

[11] T. C. Brito and M. A. Brito, "Forecasting of energy consumption : Artificial intelligence methods," in *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, 2022, pp. 1–4. DOI: 10.23919/CISTI54924.2022.9820078.

[12] H. S. Oliveira and H. P. Oliveira, "Transformers for energy forecast," *Sensors*, vol. 23, no. 15, 2023, ISSN: 1424-8220. DOI: 10.3390/s23156840. [Online]. Available: https://www.mdpi.com/1424-8220/23/15/6840.

[13] G. Sreekumar, J. P. Martin, S. Raghavan, C. T. Joseph, and S. P. Raja, "Transformer-based forecasting for sustainable energy consumption toward improving socioeconomic living: Ai-enabled energy consumption forecasting," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 10, no. 2, pp. 52–60, 2024. DOI: 10.1109/MSMC.2023.3334483.

[14] X. Li, Y. Zhong, W. Shang, X. Zhang, B. Shan, and X. Wang, "Total electricity consumption forecasting based on transformer time series models," *Procedia Computer Science*, vol. 214, pp. 312–320, 2022, 9th International Conference on Information Technology

and Quantitative Management. DOI: https://doi.org/10.1016/j.procs.2022.11.180.

[15] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1706.03762.

[16] T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, and H. Zareipour, "Energy forecasting: A review and outlook," *IEEE Open Access Journal of Power and Energy*, vol. 7, pp. 376–388, 2020. DOI: 10.1109/OAJPE.2020.3029979.

[17] R. Mulla, *Hourly energy consumption: Over 10 years of hourly energy consumption data from pjm in megawatts*, https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption, CC0: Public Domain. Zugriff am 28.12.2024, 2018.

[18] J. Cai, X. Zheng, J. Wang, and S. Meng, "Comprehensive analysis and research on the mainstream algorithm of machine learning in stock trend prediction," in *2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*, 2023, pp. 818–822. DOI: 10.1109/ICCECT57938.2023.10141254.