

Smart Grid LAB Hessen

WHITE PAPER

Smart Grid KI: Energieprognose mit maschinellem Lernen

8. JULI 2022

TRACTEBEL ENGINEERING GMBH

DR. LUCA PIZZIMBONE



In unserem früheren Artikel „[Maschinelles Lernen in Smart-Grid-Anwendungen](#)“ haben wir erörtert, wie maschinelles Lernen Smart-Grid-Anwendungen unterstützen kann, und eine Einführung in die Künstliche Intelligenz (KI), das klassische maschinelle Lernen und das Deep-Learning gegeben. Wir haben uns die aktuellen und vielversprechendsten Anwendungen dieser Technologien angesehen und die Ergebnisse eines Anwendungsfalls zur Energieprognose diskutiert.

Dieser Artikel bietet weitere Einblicke in den gleichen Anwendungsfall und stellt die architektonischen Entscheidungen, den Analyseprozess und einige zusätzliche Informationen zum Feature Engineering und zur Modellbewertung vor, die in der Studie verwendet wurden.

1 Einleitung

Vor etwa zehn Jahren wollte die britische Regierung, dass Energieversorger intelligente Zähler in jedem Haus in England, Wales und Schottland installieren. Es gab mehr als 26 Millionen Haushalte, mit dem Ziel, dass jedes Haus bis 2020 einen intelligenten Zähler hat.

Der Datensatz, den wir in dieser Analyse verwendet haben, ist eine refaktorierte Version der Daten aus dem Londoner Datenspeicher, der die Energieverbrauchswerte für etwa zwei Jahre für ein Stichprobe von 5.567 Londoner Haushalten enthält, die an dem von UK Power Networks geleiteten Low-Carbon London-Projekt teilgenommen haben.

Die Wahl des besten Machine-Learning-Algorithmus ist nicht immer offensichtlich. Manchmal ist die Verwendung klassischer einfacher maschineller Lernmodelle in Ordnung und es besteht keine Notwendigkeit, die Komplexität und die Rechenkosten zu erhöhen. Es gibt jedoch herausfordernde Situationen, in denen es notwendig ist, komplexere Modelle zu verwenden, um die gewünschte Genauigkeit zu erreichen.

Klassische Machine-Learning- und Deep-Learning-Modelle wurden mit dem Ziel definiert, ihre Leistung bei der Vorhersage des täglichen Energieverbrauchs zu vergleichen und eine Energieprognose mit einem mittleren absoluten prozentualen Fehler unter 2% zu erreichen.

1.1 ML Algorithmus

Zunächst wird ein einfacher linearer Regressionsalgorithmus an Datensätzen mit unterschiedlichen Funktionen getestet. Ein erster Datensatz enthält nur Umgebungsdaten wie min. und max. Temperatur, Taupunkt, Luftdruck, Luftfeuchtigkeit, Sichtbarkeit, UV-Index, Windgeschwindigkeit, tägliche Lichtdauer und Unterscheidung zwischen Arbeits- und Nicht-Arbeitstagen, während der zweite Datensatz zusätzliche Funktionen in Bezug auf die Energiedatenstatistiken des vorherigen Tages, der Woche und des Monats aufweist, die von den intelligenten Zählern erfasst werden.

Die Analyse bewertete den Einfluss der Feature-Auswahl (Eingabedaten) auf das lineare Regressionsmodell. Daher werden die Funktionen, die mit der linearen Regression eine bessere Leistung aufweisen, mit Gradient-Boosted-Trees und Random-Forest Regression weiter getestet. Schließlich wurde ein Deep-Learning-Algorithmus implementiert, um die Leistung von Deep-Learning- und Nicht-Deep-Learning-Algorithmen in diesem Anwendungsfall zu vergleichen.

Zusammenfassend wurden folgende Algorithmen getestet:

- Nicht-Deep-Learning-Algorithmen (klassisches maschinelles Lernen):
 - lineare Regression mit nur anfänglichen Merkmalen
 - lineare Regression mit zusätzlichen Funktionen
 - Gradient-Boosted-Trees (GBT) Regression (für den Datensatz mit zusätzlichen Funktionen)
 - Random-Forest-Regression (nur für den Datensatz mit zusätzlichen Funktionen)
- Deep-Learning-Algorithmen:
 - LTMS (nur für den Datensatz mit zusätzlichen Funktionen)

1.2 ML Architektur

Die Ergebnisse des leistungsstärksten Modells wurden zur Diskussion mit einem Basisfall einer naiven Vorhersage verglichen, der einfach davon ausgeht, dass die an einem bestimmten Tag verbrauchte Energie der am Vortag verbrauchten Energie entspricht.

Die Analyse wurde in mehrere sequenziellen Aufgaben unterteilt, was beim maschinellen Lernen üblich ist:

- Erste Datenuntersuchung
- Extraktion der Transformationslast
- Feature-Engineering
- Modelldefinition
- Modelltraining
- Modellbewertung

Für eine bessere Rechenleistung untersuchten wir die Bereitstellung der Modelle in einem Cloud-System (IBM Cloud-System, durch Watson Studio-Ausführungsumgebung mit Python 3.7 und Spark 3.0).

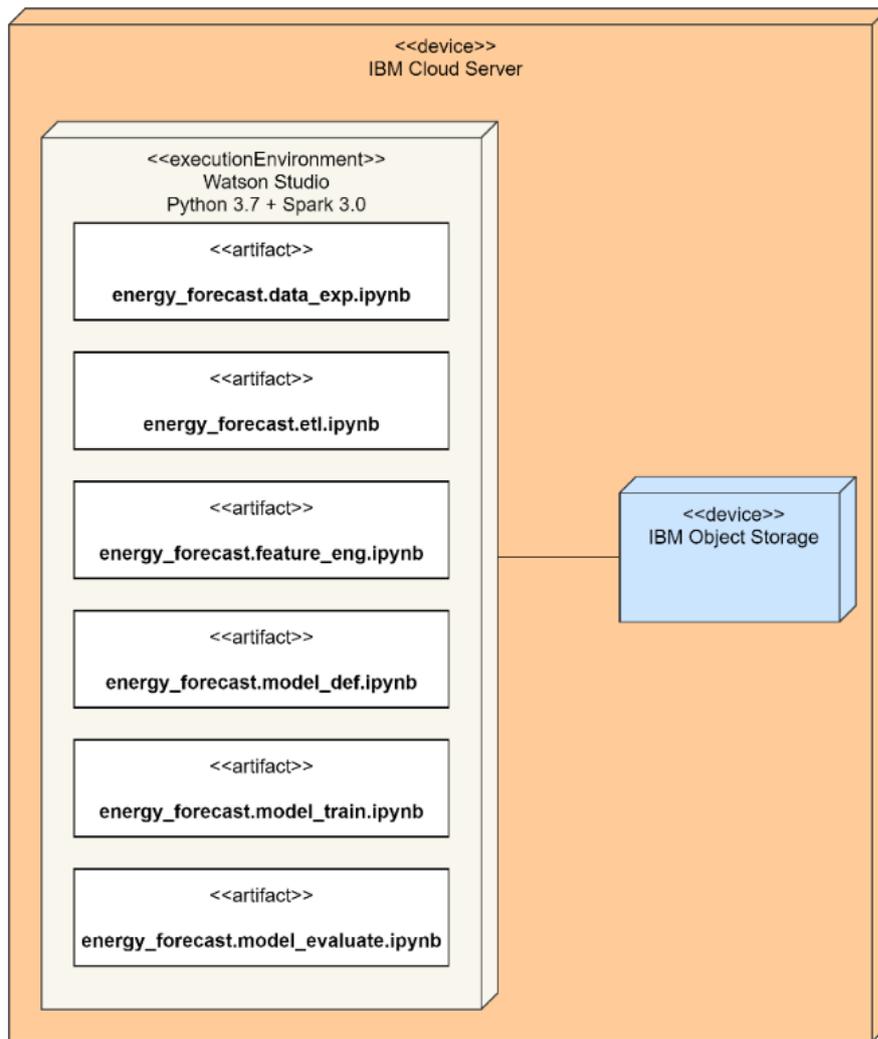


Abbildung 1 – ML Architektur

1.3 Metrik

Das Modell wurde während des Trainings und Tests mit Hilfe von R-Quadrat-Metriken bewertet, die als Index angesehen werden können, der zeigt, wie gut das Modell zu den Daten passt. Das R^2 bewegt sich von 0 auf 1. Im Allgemeinen zeigt ein höheres r-Quadrat eine bessere Anpassung an das Modell an.

1.4 Ergebnisse

Die Ergebnisse der Studie in Bezug auf die Modellleistung sind in Tabelle 1 und Abbildung 2 dargestellt. Die leistungsstärksten Modelle sind Modelle mit einem höheren R^2 , aber auch einer guten Balance zwischen Zug- und Testsatzergebnissen. Modelle mit hohen r-quadrierten Ergebnissen für Trainingsdaten, aber deutlich geringerer Leistung bei Testdaten sind überanpassungsfähig und müssen möglicherweise weitere Optimierungsprozesse durchlaufen.

Model	R Square (Train data)	R Square (Test data)
0 Linear Regression (Basic Features)	0.850061	0.682555
1 Linear Regression (Extra Features)	0.934283	0.918921
2 Gradient Boosted Trees (Extra Features)	0.985304	0.884309
3 Random Forest (Extra Features)	0.972449	0.920160
4 Gradient Boosted Trees (Cross Validation)	0.985069	0.910467
5 LSTM	0.977853	0.962497

Tabelle 1 - Vergleich der Modellleistung unter Verwendung von R^2 -Metriken (Tabellenformat).

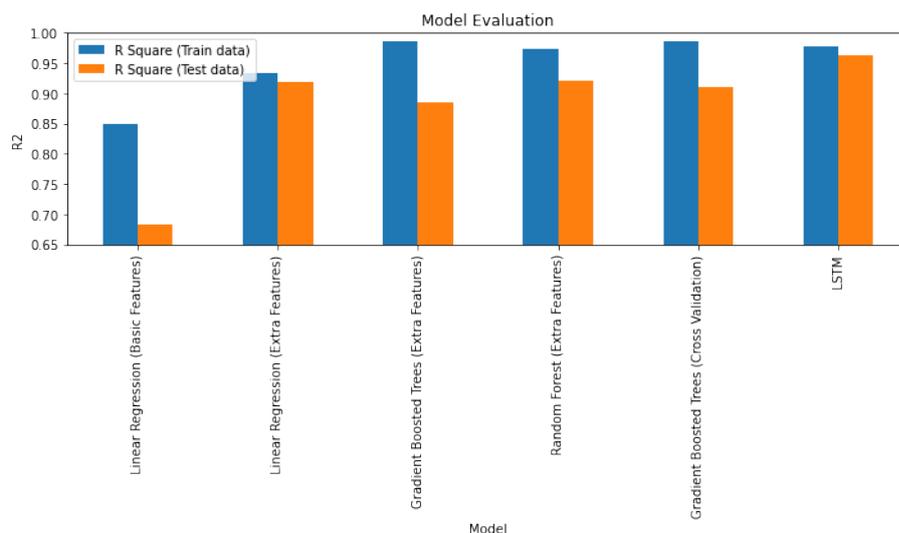


Abbildung 2 - Vergleich der Modellleistung unter Verwendung von R-Quadrat-Metriken (Diagrammformat).

Lassen Sie uns die Ergebnisse von Abbildung 2 diskutieren. Erwartungsgemäß ist eine gute Feature-Auswahl beim klassischen maschinellen Lernen sehr wichtig. Die Leistung des Modells mit grundlegenden Funktionen (nur Wetterdaten) ist ziemlich schlecht und weitgehend überanpassungsfähig (erste paar Balken - blau / orange - von links). Das Hinzufügen zusätzlicher Funktionen, wie z. B. Statistiken über den Energieverbrauch am Vortag, in der Vorwoche und im Vormonat, scheint eine gute Wahl zu sein. Mit diesen zusätzlichen Funktionen liefert die einfache lineare Regression ein r -Quadrat von etwa 0,934 auf Zugdaten und 0,918 auf Testdaten.

Durch die Verwendung der gleichen zusätzlichen Funktionen wurden zwei weitere klassische Machine Learning-Modelle definiert, trainiert und bewertet: Gradient-Boosted-Trees und Random-Forest.

Gradient-Boosted-Trees erzielten das höchste r -Quadrat in den Zugdaten (0,985), schnitten aber bei den Testdaten recht schlecht ab (0,884). Auch hier ist das Modell überanpassungsfähig. Diesmal wurde eine Technik namens Kreuzvalidierung verwendet, um die Hyperparameter des Modells zu optimieren, um eine Überanpassung zu vermeiden. Die Ergebnisse der Kreuzvalidierung zeigen eine Verbesserung des r -Quadrats in den Testdaten (0,910), ohne dass die Leistung der Zugdaten signifikant reduziert wird. Die Punktzahl für Testdaten liegt jedoch immer noch unter dem Random-Forest, der das leistungsstärkste klassische maschinelle Lernmodell auf den Testdaten zu sein scheint.

Schließlich wurde ein Deep-Learning-Modell implementiert, um die Ergebnisse mit klassischen Machine-Learning-Modellen zu vergleichen. Zum Einsatz kommt das bekannte Long-Short-Term-

Memory (LSTM) rekurrente neuronale Netz, das sich besonders für die Vorhersage von Zeitreihendaten eignet. Diese Art von neuronalem Netzwerk kann die Ordnungsabhängigkeit zwischen Elementen in einer Sequenz lernen, was bedeutet, dass es die Daten für eine genauere Vorhersage kontextualisieren kann.

Das LSTM-Modell hatte das höchste R im Quadrat des Testdatensatzes und die Leistungsergebnisse für Zug- und Testdaten sind sehr gut ausbalanciert.

Das Diagramm des vorhergesagten Wertes im Vergleich zu den wahren Werten des Testsatzes ist in Abbildung 3 für die klassischen Modelle für maschinelles Lernen (Random-Forest) und Deep-Learning (LSTM) dargestellt.

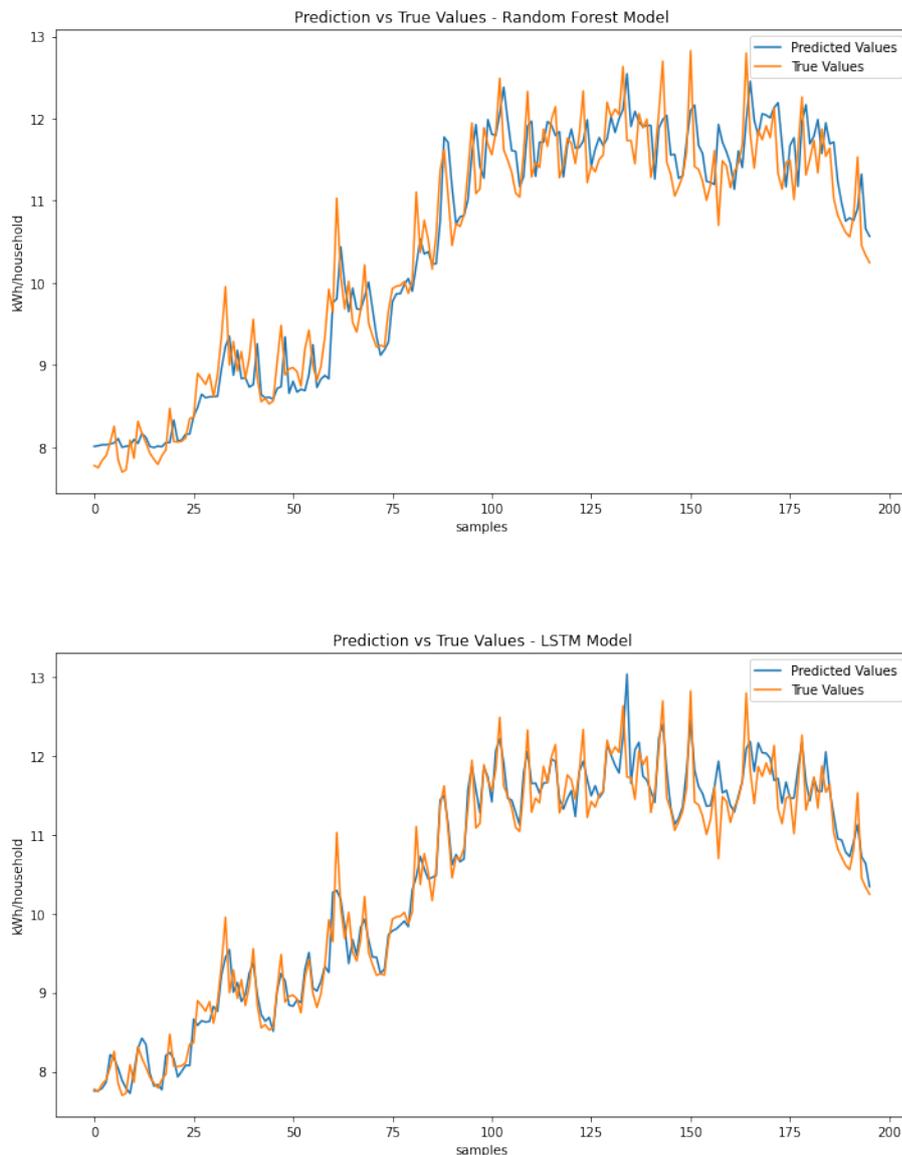


Abbildung 3 – Vorhersage im Vergleich zu wahren Werten für Random-Forest- und LSTM-Modelle. Die beiden Diagramme mögen sehr ähnlich erscheinen, aber mit Aufmerksamkeit bemerken wir mehrere verschiedene Verhaltensweisen in der Vorhersage. Das LSTM-Modell war dasjenige, das besser abgeschnitten hat.

1.5 Weitere Modellbewertung

Für das Modell mit der besten Performance (LSTM) wurden Leistungsindizes wie Mean-Absolute-Error, Mean-Squared-Error und Mean-Absolute-Percentage-Error berechnet und in Abbildung 4 dargestellt.

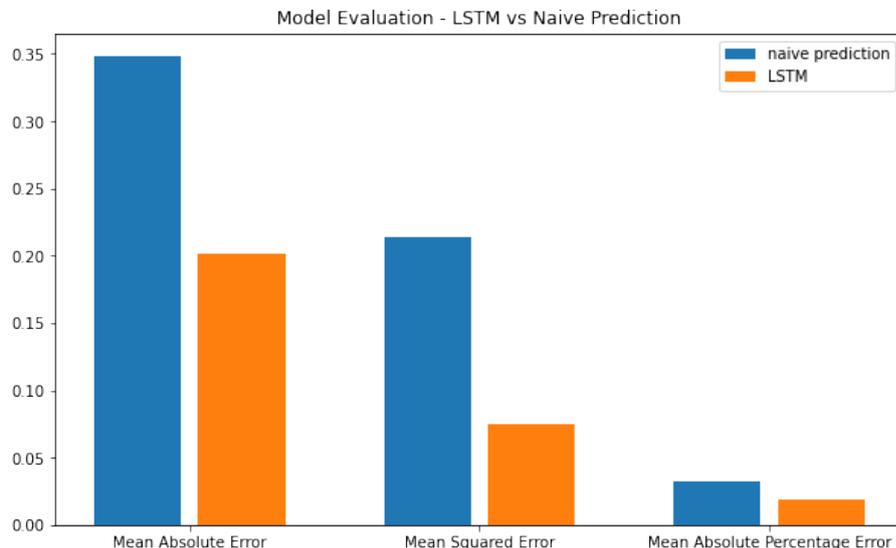


Abbildung 4 – Leistung des LSTM-Modells im Vergleich zur naiven Vorhersage. Eine signifikante Verbesserung bietet LSTM im Vergleich zu einer naiven Vorhersage, bei der davon ausgegangen wird, dass der Energieverbrauch eines bestimmten Tages dem des Vortages entspricht.

Der LSTM übertrifft eine naive Prognose in allen berechneten Performance-Indizes bei weitem. Im Falle des mittleren quadratischen Fehlers (MSE) ist die Verbesserung ziemlich beeindruckend, von etwa 0,213556 auf 0,074922. Dies bedeutet, dass das implementierte LSTM-Modell große Fehler recht gut einschränkt, hier durch das Quadrat der MSE-Metrik bestraft.

Obwohl der mittlere absolute prozentuale Fehler auch im LSTM-Modell deutlich niedriger ist (von etwa 3,3% auf 1,9%), ist seine Verbesserung weniger spektakulär. Dies deutet darauf hin, dass die Schwankungen des Energieverbrauchs zwischen zwei aufeinanderfolgenden Tagen nicht sehr groß sind. Dennoch gibt es genügend Beispiele für größere Variationen, um die in der naiven Prognose zu erhöhen. Hier leistet der LSTM seinen Hauptbeitrag.

Einzelne Zeitpunkte mit über 10% Fehlern sind auch im LSTM-Modell vorhanden (siehe Abbildung 5).

Dies spiegelt das Grundprinzip wider, dass Zug- und Testdaten tatsächlich Unterschiede aufweisen und dass die Merkmale (intelligente Zählerstände und Wetterdaten) die beobachtete Variable (Energieprognose) möglicherweise nicht vollständig beschreiben. Dies schließt die Nützlichkeit des Modells nicht aus, das die Energieprognose und die korrelierten Entscheidungen im Last- und Erzeugungsmanagement effektiv unterstützen kann.

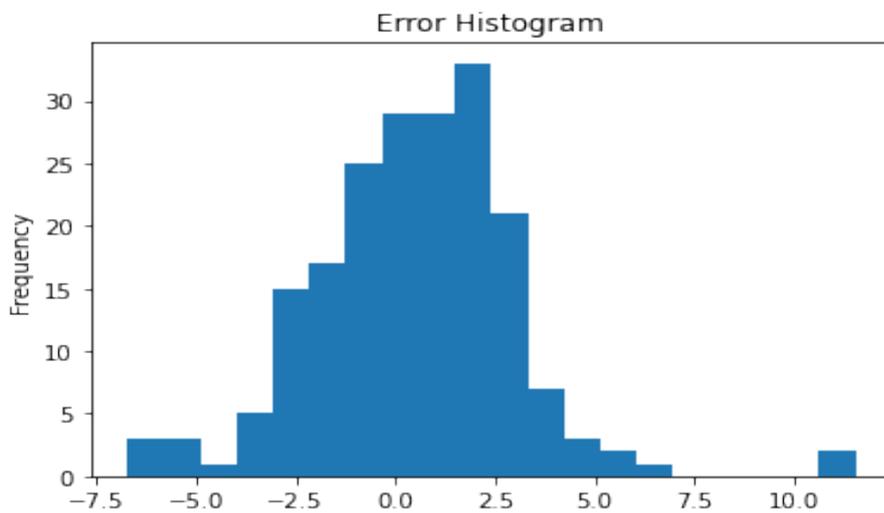
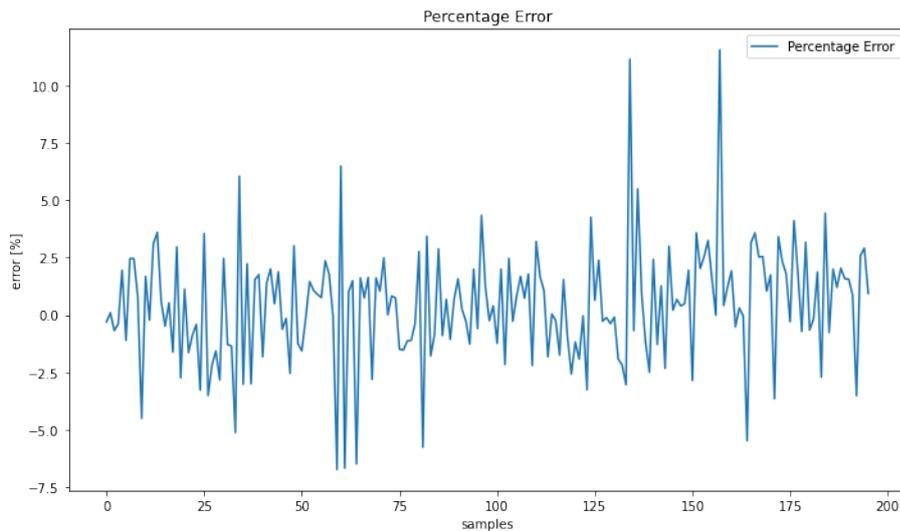


Abbildung 5 – Prozentualer Fehler von LSTM. Der mittlere absolute Fehler beträgt etwa 1,9%. Allerdings weisen nur wenige Beispielausnahmen viel höhere Fehler auf (über 10 %).

1.6 Schlussfolgerungen

In diesem Anwendungsfall fanden wir ziemlich signifikante Verbesserungen der Vorhersagen durch die Verwendung eines Deep-Learning-Algorithmus, mit einem durchschnittlichen absoluten prozentualen Zielfehler unter 2%, auch ohne eine vollständige Optimierung des Modells. Es ist vernünftig anzunehmen, dass die Kluft zwischen naiver Prognose und Deep-Learning-Modell in einem Stromumfeld, das stark von erneuerbaren Energien, Elektrofahrzeugen und anderen stochastischen Last- und Erzeugungsquellen beeinflusst wird, noch größer wäre.

Über den Autor:



Dr. Luca Pizzimbone ist ein wichtiger Experte für die Analyse von Energiesystemen, der in der Energiewende und der Entwicklung von Stromübertragungs- und Verteilnetzen weltweit tätig ist. Er hat ein Diplom in Elektrotechnik von der C.I.T.I. G. Galilei in Genua (Italien), einen Dokortitel (Laurea di Dottore) in Elektrotechnik von der Universität Genua und ist ein IBM zertifizierter Advanced Data Scientist.

Als Teil der Abteilung Power Transmission and Distribution der Tractebel Engineering GmbH leitete er Projekte in den Bereichen Netzplanung, erneuerbare Energien, Hybridsysteme und Smart Grids in Afrika, Asien, Europa und dem Nahen Osten.

Er lebt in Bad Vilbel, Deutschland.