AIR POLLUTION FORECASTING

Simon Spitzer
business informatics student
DHBW Stuttgart
simon.spitzer@live.de
Projektdokumente Link: Github Repository

ABSTRACT

Diese Studie entwickelt ein Framework zur Vorhersage von Luftverschmutzung mittels maschinellen Lernens. Basierend auf historischen Daten und meteorologischen Faktoren wird ein Modell trainiert und evaluiert. Das anschließend optimierte Modell zeigt eine hohe Vorhersagegenauigkeit und erfasst zeitliche Abhängigkeiten effektiv. Dieses Modell könnte so politische Entscheidungsträger und Umweltbehörden bei der Planung von Maßnahmen zur Luftqualitätskontrolle unterstützen.

***K*eywords** Air Pollution Forecasting · LSTM · Regression · MLFLow

1 Charakterisierung & Aufbereitung Datensatz

1.1 Studienziel und Datensatz

Diese Studie beschäftigt sich mit der Vorhersage von Luftverschmutzung einer US-Botschaft in Peking, China durch Einsatz maschinellen Lernens.[1] Der Datensatz streckt sich über fünf Jahre und enthält ca. 43.800 Datenpunkte, wobei jeder Datenpunkt einer stündlichen Aufnahme entspricht. Der vorherzusagende Wert ist die Luftverschmutzung, die als PM2.5-Konzentration angegeben wird und als Feinstaub bekannt ist. Feinstaub bezeichnet Staubpartikel mit einer Korngröße von weniger als zehn Mikrometern.[2] Der darin enthaltene PM2.5-Feinstaub trägt wesentlich zur Entstehung vieler Krankheiten bei. Zu den negativen Folgen von PM2.5-Feinstaub gehören unter anderem Atemwegserkrankungen und Herz-Kreislauf-Erkrankungen.[3] Das konkrete Ziel dieser Studie ist es, das Modell mit Daten der ersten vier Jahre zu trainieren, um die Luftverschmutzung im fünften Jahr möglichst genau vorherzusagen. Im Detail wird in diesem fünften Jahr **immer der nachfolgende Tag vorhergesagt** mit dem Input von X Tagen in der vergangenheit, welche abhängig von der Batch Size sind. Der benutzte Datensatz enthält dabei 9 verschiedene Eingabespalten ("Features"), die wie folgt aufgebaut sind:

Feature	**Erklärung**
Datum	Datum in yyyy-mm-dd, hh:mm:ss
Verschmutzung	Luftverschmutzung in µg/m³
Taupunkt	Taupunkt in °C
Temperatur	Temperatur in °C
Druck	Luftdruck in hPa
Windrichtung	Windrichtung, z.B. SE
Windgeschwindigkeit	Windgeschwindigkeit in m/s
Schnee	Schneefall in mm
Regen	Regenfall in mm

Table 1: Datenübersicht

[1] vgl. 1, p. 1.
[2] vgl. 3, p. 1.
[3] vgl. 4, p. 1.

1.2 Feature Engineering und Aufbereitung des Datensatzes

Diese Studie beutzt ein LSTM Modell als neuronales Netz, auf das später im Text genauer eingegangen wird. Aufgrunddessen soll der Datensatz entsprechend der Anforderungen des Modells aufbereitet werden. Nach Erläuterung des Datensatzes werden die Datenpunkte auf **Qualität** überprüft. Hierbei werden zuerst Nullwerte untersucht um fehlende Datenpunkte zu identifizieren. Es stellt sich dabei heraus, dass keine Nullwerte vorhanden sind und der Datensatz somit vollständig ist. Des Weiteren kann gesagt werden, dass keine inkonsistenz in den Daten existiert, da jedes Jahr durch die stündliche Datengenerierung die gleiche Anzahl an Datenpunkten sichergestellt ist. Durch visualisieren der Verteilung der Datenpunkte auf ihrer Skala pro Kategorie können keine Ausreißer festgestellt werden. Weiterhin ist eine Balanciertheit der Daten gegeben. Ein Ausschnitt der Daten für die Luftverschmutzung (lila) und die Temperatur (grün) ist in folgender Visualisierung Abb1. sichtbar:

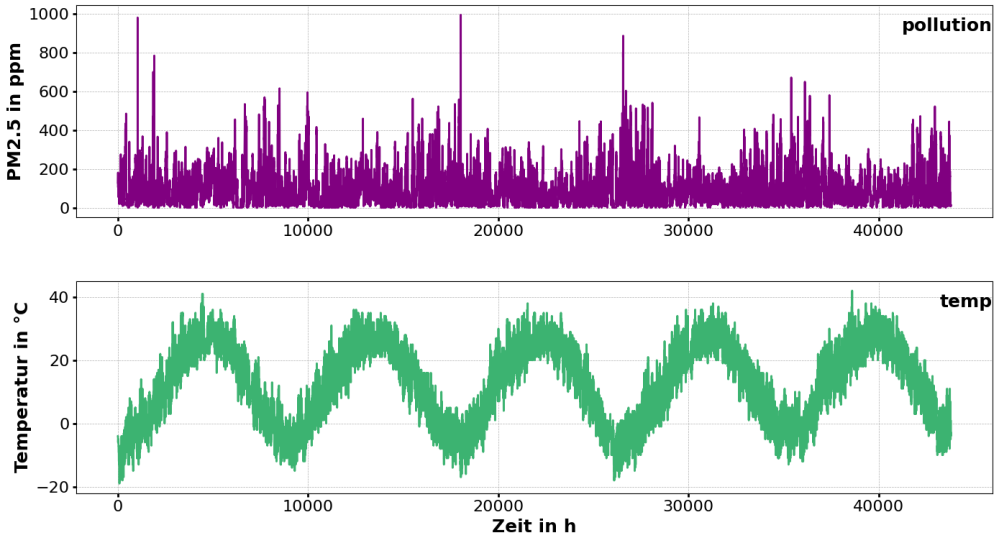

Figure 1: Datenkorrelation zwischen der Luftverschmutzung und Temperatur

Bei genauerer Betrachtung ist zu erkennen, dass sich die gewünschte Zielmetrik Luftverschmutzung antikorelliert verhält zur Temperatur. Zusätzlich ist bei Werten wie der Temperatur, Regen oder Windgeschwindigkeit eine Abhängigkeit zu Jahreszeiten erkennbar und somit existieren periodische Zyklen.

Generell werden alle **Features** mit Außnahme des Datums als Input benutzt. Dieses wird gelöscht und stattdessen eine Indexierung von 0 bis 43.800 Stunden verwendet, da dies die weitere Verarbeitung vereinfacht und zusätzlich Modellstabilität, sowie Skalierbarkeit positiv beeinflusst. Im nächsten Schritt steht die **Datenaufbereitung** im Fokus. Zu Beginn ist eine Umwandlung der aktuellen Windrichtungen (NE, SE, NW, cv) in numerische Werte nötig, um sie für das LSTM nutzbar zu machen, wobei "cv" eine calm variable für Wind aus mehreren Richtungen darstellt. Somit sind die neuen label encoded Werte wiefolgt: [’1’, ’2’, ’3’, ’4’]. Danach wird eine Datentyp Standardisierung vorgenommen, die den Datentyp “float32” für alle Spalten festlegt. Somit wird sichergestellt, das alle Daten im gleichen Format vorliegen, was für viele Machine-Learning-Algorithmen wichtig ist. Zusätzlich wird ein MinMaxScaler angewendet, der die Daten in einen Bereich von [0 , 1] skaliert und so für numerische Stabilität, sowie eine Vermeidung von Dominanz großer Werte sorgt. Im nächsten Schritt muss der Datensatz noch auf das bevorstehende LSTM Modell vorbereitet werden. Dabei wird der Datensatz als überwachtes Lernproblem formuliert und die Eingabevariablen werden normalisiert. Die dazugehörige Funktion zur Transformation und Normalisierung der Daten ist übernommen aus einem Machine-Learning Mastery Blog.[4] Der Output enthält zusätzliche Spalten, die um eine Position verschoben sind, sodass mit den aktuellen Eingabespalten der jeweils nächste Wert vorhergesagt werden kann. Eine Visualisierung der Transformation kann aus folgender Tab.2 entnommen werden.

[4] vgl. 2, p. 1.

var1(t-1)	var1(t)
0.129779	0.148893
0.148893	0.159960
0.159960	0.182093
0.182093	0.138833
0.138833	0.109658

Table 2: Verschobene Werte für ein Supervised Learning Problem

Generell ist zu erkennen, dass alle Werte skaliert, normiert und nun auch als überwachtes Lernproblem dargestellt sind. Wie farbig zu erkennen ist, ist der aktuelle Wert der Luftverschmutzung var1(t-1) und der zu vorhersagende Wert var1(t) um eins versetzt und gewährleistet so eine Vorhersarge der Verschmutzung mit zeitlich vorangegangenen Daten. Der letzte Schritt der Datenaufbereitung beinhaltet die Aufteilung der Daten in Trainingsdaten und Evaluationsdaten. Die Aufteilung ist klassisch durch 80% Trainings- und 20% Evaluationsdaten gekennzeichnet, wobei dies den oben genannten vier Jahren für Training und einem Jahr Evaluation entspricht. Nachfolgend werden beide Teildatensätze nochmals in Eingabedaten, die alle Merkmale außer dem vorherzusagenden "Luftverschmutzung", enthalten (train_X, test_X), und die vorherzusagenden Labels (train_Y, test_Y) unterteilt. Schlussendlich muss das Format der Daten noch in einem 3D-Array umgewandelt werden, da dies von einem LSTM benötigt wird. Der 3D-Array sieht dabei wie folgt aus: Anzahl der Datenpunkte, Anzahl der Zeitschritte, Anzahl der Merkmale.

2 Neuronales Netz

2.1 Zielmetrik

Für das Training des Modells wird die Zielmetrik der mittleren quaratischen Abweichung benutzt. Diese ist der Standard für Regressionsaufgaben, wie die Vorhersage der Luftverschmutzung. Der Mean Squared Error (MSE) wird durch die folgende Formel definiert:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{1}$$

wobei y_i die tatsächlichen Werte und $\hat{y}_i$ die vorhergesagten Werte sind. MSE eignet sich gut für Regressionsaufgaben, da sie große Fehler stärker gewichtet und somit das Modell dazu zwingt, größere Abweichungen zu minimieren, was zu präziseren Vorhersagen führt.[5] Zusätzlich werden zwei weitere Metriken erhoben: Mean Absolute Error (MAE) und Root Mean Squared Error(RMSE), jedoch werden diese nicht als loss zum Training benutzt. Dennoch ermöglichen Sie ein besseres Verständnis über die Performance des Modells während und nach dem Training. Der MAE gibt hierbei die durchschnittliche Abweichung der absoluten Differenzen zwischen vorhergesagten und tatsächlichen Werten an. Dies erlaubt Einblick in die durchschnittliche Größe der Fehler ohne zu stark auf Ausreißer zu reagieren und ermöglicht zudem ein einfacheres Verständnis, da die Abweichung in derselben Einheit wie die Daten angegeben ist. Der RMSE ist die Quadratwurzel des MSE und erfüllt den gleichen Zweck, jedoch steht er dadurch in der selben Einheit wie die Daten und ist somit leichter interpretierbar. Dennoch reagiert der RMSE stark auf größere Fehler, wie der eigentliche MSE. Nutzen mehrerer Metriken ermöglicht zudem ein besseres Verständnis der Performance beim Hyperparametertuning.

2.2 Aufbau des Modells

Das Benutzte neuronale Netzwerk ist ein sequentielles Modell zur Vorhersage von Luftverschmutzung bestehend aus insgesamt **fünf Schichten** und insgesamt 288,129 Parametern. Die Struktur ist inspiriert aus folgenden Papers, jedoch abgewandelt uns für diesen Datensatz und Aufgabenstellung angepasst[6]:

1. **LSTM-Schicht**: Diese Long Short Term Memory (LSTM) Schicht enthält 256 Neuronen und ist in der Lage, langfristige Abhängigkeiten in zeitlich geordneten Daten zu lernen. Die LSTM-Schicht verarbeitet die Eingabezeitreihen und extrahiert relevante Merkmale, die für die Vorhersage wichtig sind.

2. **Dense-Schicht**: Diese Schicht besteht aus 64 Neuronen und transformiert die von der LSTM-Schicht extrahierten Merkmale. Sie ermöglicht es dem Modell, komplexe nichtlineare Beziehungen zu lernen.

3. **Dropout-Schicht**: Mit einer Rate von 0,25 wird ein zufälliger Prozentsatz der Neuronen während des Trainings deaktiviert. Dies hilft, Overfitting zu vermeiden und die Generalisierungsfähigkeit des Modells zu verbessern.

[5] vgl. 5, p. 672.
[6] vgl.[6], p. 2-4; vgl.[5], p. 673

4. **Batch-Normalisierungs-Schicht**: Diese Schicht normalisiert die Ausgaben der vorhergehenden Schicht. Sie beschleunigt das Training und verbessert die Stabilität des Modells, indem sie die Verteilungen der Inputs zu den nachfolgenden Schichten konsistent hält.

5. **Finale Dense-Schicht**: Die letzte Schicht besteht aus einem einzelnen Neuron, das den vorhergesagten Wert der Luftverschmutzung ausgibt. Diese Schicht aggregiert die verarbeiteten Informationen und liefert die endgültige Vorhersage.

Des Weiteren wird der **Adam-Optimizer** verwendet, um das Modell zu trainieren. Er kombiniert die Vorteile von AdaGrad und RMSProp, was zu einer schnellen und effizienten Konvergenz führt und somit die Optimierung des Modells verbessert. Die **Loss-Funktion** des Modells, während dem Training, ist der Mean Squared Error (MSE), der große Fehler stärker gewichtet und somit die Vorhersagegenauigkeit verbessert. Das Modell verwendet den mit einer Lernrate von 0.001.

2.3 Hyperparameter

Durch die einzelnen Schichten gibt es etliche Hyperparameter, die getuned werden können, um die Genauigkeit des Modells zu erhöhen. Die Hyperparameter sind:

Hyperparameter	**Benutzte Standardwerte**
lstm_units	256
dense_units_1	64
dense_units_2	1
dropout_rate	0.25
learning_rate	0.001
epochs	50
batch_size	128
activation_function (Dense-Schichten)	Linear
optimizer	Adam

Table 3: Hyperparameter und deren Standardwerte

3 Evaluation des Trainings und Hyperparametertuning

Generell wird für die Durchführung des Trainings Keras, als High-Level-API und Teil von Tensorflow zur Erstellung und Training von Deep-Learning-Modellen benutzt. Das Ergebnis des nicht Hyperparameter optimierten Modells ist in folgender Abb.2 dargestellt:

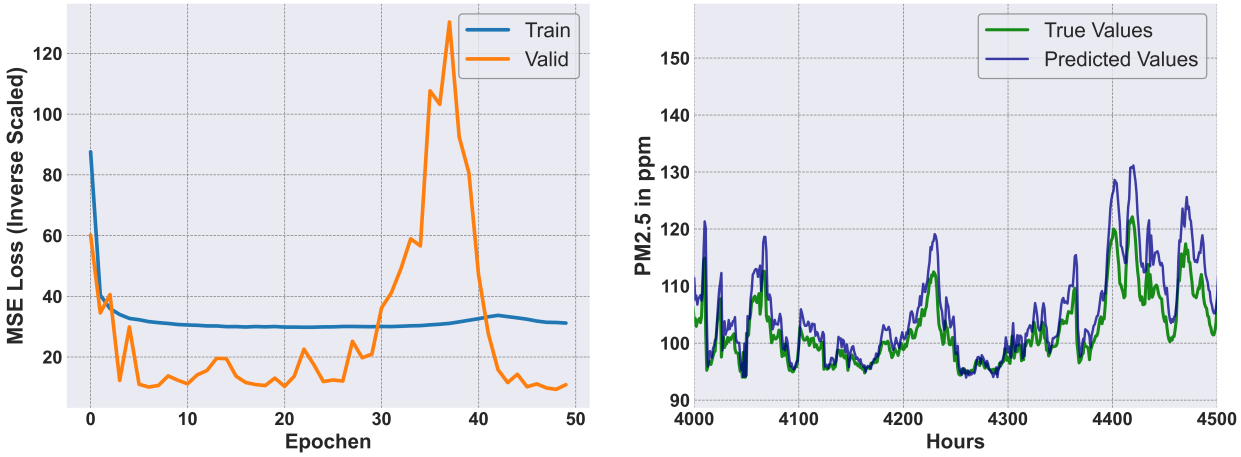

Figure 2: Initialergebnisse

In der linken Abb. ist zu erkennen, dass das Standardmodell nach 50 Trainingsepochen einen trainings-loss von 31.1 bezogen auf den MSE und sogar einen evaluation-loss von 10.8 erreicht, welche auf gute Performance deuten. Hierbei ist zu erwähnen, dass der MSE Loss invertiert auf die Originalskala ist um eine korrekte und verständliche Interpretation zu ermöglichen. Des Weiteren sind ein paar Ausreißer in höheren Epochen des Trainings zu erkennen, welche sich jedoch gegen Ende des Trainings vaporisieren und nicht genauer untersucht werden. Der Rechte plot veranschaulicht die vorhergesagten Werte im Vergleich zu den korrekten Luftverschmutzungswerten. Hierfür wurde zudem der RMSE berechnet, indem die Differenzen zwischen den tatsächlichen Werten $y_{\text{test\_true},i}$ und den vorhergesagten Werten $\hat{y}_{\text{testPredict2},i}$ quadriert, gemittelt und dann die Quadratwurzel genommen wird. Die Berechnung erfolgt somit nach folgender Formel:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{\text{test\_true},i} - \hat{y}_{\text{testPredict2},i})^2}, \tag{2}$$

mit dem Ergebnis von 3.29, was bedeutet, dass Vorhersagen hauptsächlich im Bereich von +-3.29ppm um die realen Werte schwanken.

Im nächsten Schritt findet ein Hyperparametertuning mit dem ML Lifecycle Mgt Tool "MLFLow" statt. Da es etliche, Hyperparameter gibt und der Zeitaufwand zu enorm ist, alle Hyperparameter zu betrachten, wird sich auf die Batch Size konzentriert. Diese gibt an, wie viele Trainingsbeispiele in einem einzelnen Vorwärts-/Rückwärtsdurchlauf verarbeitet werden. Speziell wird die Batch Size zu Beginn auf 2 gesetzt und anschließend verdoppelt, bis eine Batch Size von 256 erreicht wird. Der Hintergrund hierbei ist, dass ein Vielfaches von 2 oftmals recheneffizieter ist und mehr Hardwarekompatibilität bietet. Als Ausnahme sind drei Zwischenschritte bei 100, 150 und 200 eingefügt, um zu große Abstände der Batch Sizes vorzubeugen. MLFlow benötigt hierbei einen Experimentnamen, unter dem die Runs gespeichert werden, und zudem die zu speichernden Parameter, wie etwa der MSE Loss oder RMSE und natürlich die einzelnen Hyperparameter. Ebenfalls muss hier darauf geachtet werden, dass die Evaluationsmetrik auf die Originalskala inversiert werden kann, um Vergleichbarkeit und Interpretierbarkeit zu schaffen. Das Ergebnis des Hyperparametertunings ist in folgender Abb.3 ersichtlich:

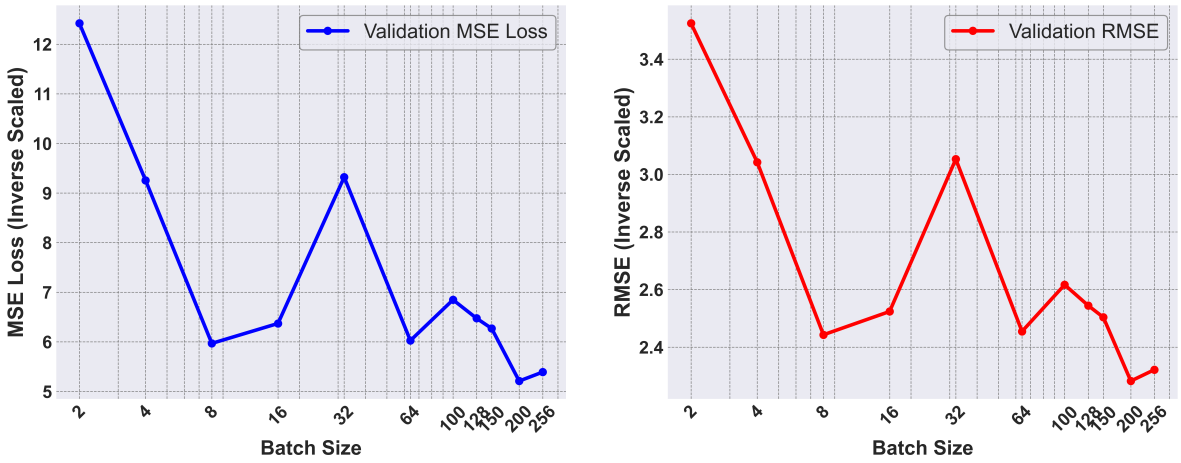

Figure 3: Hyperparametertuning der Batch Size

Aufgeführt in blau ist der MSE Loss und in rot der besser interpretierbare RMSE. Es ist zu erkennen, dass das Modell mit höheren Batch-Sizes, speziell einer Batch-Size von 200, seine Genauigkeit von den ursprünglichen +-3.29 (RMSE) auf +-2.28 (RMSE) senken kann, was einer Verbesserung von 31% entspricht. Diese Verbesserung durch die Verwendung größerer Batch-Sizes ist darauf zurückzuführen, dass mehr Daten aus der Vergangenheit genutzt werden, wodurch die Luftverschmutzung am Folgetag genauer vorhergesagt werden kann.

4 Fazit

Die Studie zeigt, dass ein LSTM-Modell effektiv zur Vorhersage der Luftverschmutzung genutzt werden kann. Durch die Aufbereitung und Skalierung der Daten, sowie der Anwendung von Feature Engineering, konnte ein robustes Modell entwickelt werden. Die Hyperparameteroptimierung, speziell der Batch Size, führte zudem zu einer signifikanten Steigerung der Vorhersagegenauigkeit, wobei der RMSE von ursprünglich +-3.29 ppm auf +-2.28 ppm gesenkt wurde.

Dies entspricht einer Genauigkeitsverbesserung von 31%. Zusammenfassend stellt das Modell eine wertvolle Ressource zur Unterstützung politischer Entscheidungsträger und Umweltbehörden dar, um die Planung von Maßnahmen zur Luftqualitätskontrolle zu verbessern.

References

[1] *Air Pollution Forecasting - LSTM Multivariate*. URL: https://www.kaggle.com/datasets/rupakroy/lstm-datasets-multivariate-univariate (visited on 05/28/2024).

[2] Jason Brownlee. *Multivariate Time Series Forecasting with LSTMs in Keras*. MachineLearningMastery.com. Aug. 13, 2017. URL: https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/ (visited on 06/04/2024).

[3] Bundesumweltministeriums. *Feinstaub*. Bundesministerium für Umwelt, Naturschutz, nukleare Sicherheit und Verbraucherschutz. URL: https://www.bmuv.de/WS538 (visited on 05/28/2024).

[4] *Feinstaub - PM2,5*. URL: https://www.umweltbundesamt.at/umweltthemen/luft/luftschadstoffe/staub/pm25 (visited on 05/28/2024).

[5] *Linear Regression vs LSTM for Time Series Data | IEEE Conference Publication | IEEE Xplore*. URL: https://ieeexplore.ieee.org/document/9848887 (visited on 06/26/2024).

[6] Fangbi Tan. "Regression Analysis and Prediction Using LSTM Model and Machine Learning Methods". In: *Journal of Physics: Conference Series* 1982.1 (July 2021), p. 012013. ISSN: 1742-6596. DOI: 10.1088/1742-6596/1982/1/012013. URL: https://dx.doi.org/10.1088/1742-6596/1982/1/012013 (visited on 06/26/2024).