



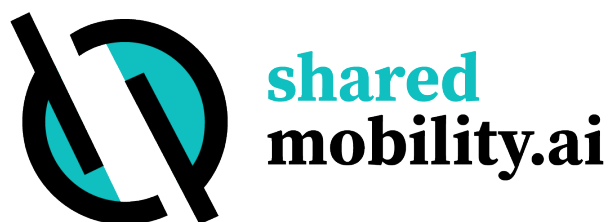
SharedMobility.ai

Endbericht | Call 13 | Projekt ID 3881

Lizenz: CC-BY-SA

Inhalt

1	Einleitung	3
2	Projektbeschreibung	3
3	Verlauf der Arbeitspakete.....	5
3.1	Arbeitspaket 1 - <i>Detailplanung und Formales am Projektstart</i>	5
3.2	Arbeitspaket 2 – <i>Öffentlichkeitsarbeit / Kommunikation / Grafik</i>	5
3.3	Arbeitspaket 3 – <i>Data Ingestion</i>	6
3.4	Arbeitspaket 4 – <i>AI Modelling</i>	6
3.5	Arbeitspaket 5 – <i>Web-Schnittstelle</i>	7
3.6	Arbeitspaket 6 – <i>Projektabschluss</i>	8
4	Liste Projektendergebnisse	9
5	Verwertung der Projektergebnisse in der Praxis.....	9
6	Öffentlichkeitsarbeit / Vernetzung	10
7	Eigene Projektwebsite	10
8	Geplante Aktivitäten nach netidee-Projektende	11
9	Anregungen für Weiterentwicklungen durch Dritte	12



1 Einleitung

Geteilte Mobilitätsangebote werden immer beliebter und bilden mittlerweile einen wichtigen Faktor im städtischen Angebot an flexibler Last-Mile-Mobilität. SharedMobility.ai versucht intelligente Prognosen für stationsgebundene Verleihsysteme, wie z.B. Citybike Wien oder die SeestadtFLOTTE, zu erstellen. NutzerInnen sollen in Zukunft nicht mehr vor vollen Rückgabeboxen oder leeren Entleihstationen stehen, wenn sie nach einer Route durch urbane Räume suchten. Wir stellen hierfür einen ersten wichtigen Baustein bereit, der das Fundament für weitere Anwendungen bilden kann. Außerdem zeigen wir mit einer Proof-of-Concept Webseite eine mögliche Implementierung für NutzerInnen von Shared Mobility Angeboten.

Hierfür erarbeiteten wir im Rahmen der netidee einerseits Schnittstellen und Tools zum Verarbeiten und Aufbereiten der Stationsdaten, andererseits berechneten wir TensorFlow-Modelle zur Vorhersage der Auslastung. Diese Modelle können später weiter verfeinert werden und mit neuen Datenströmen verbessert werden.

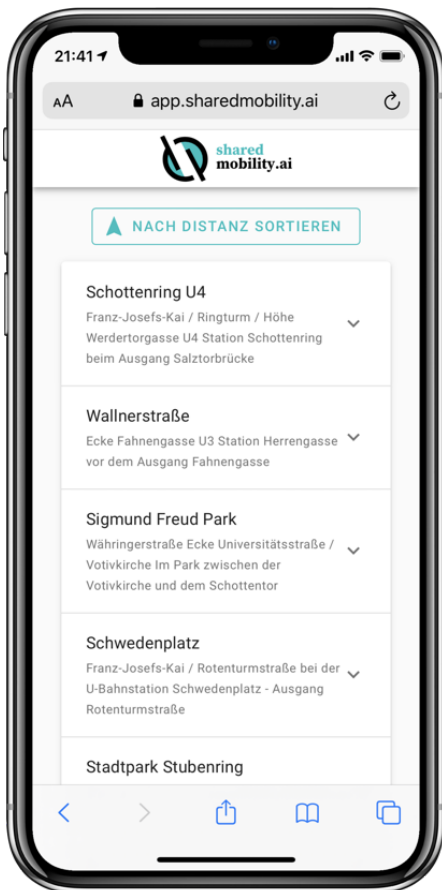
2 Projektbeschreibung

SharedMobility.ai richtet sich zuerst einmal an EntwicklerInnen aus dem Bereich Maschinelles Lernen / AI, die eigene Vorhersagemodelle für urbane Mobilitätsangebote erstellen möchten. Für

sie sind wir eine erste Anlaufstelle wenn es um die dafür benötigte Datengrundlage geht. Wir geben Hinweise für eine möglich Umsetzung und welche Faktoren zu berücksichtigen sind. Außerdem liefern wir Rohdaten in Form von CSV-Datasets, die die Entwicklung von eigenen Modellen beträchtlich erleichtert.

Wir haben bewusst einen Blackbox-Ansatz gewählt und sind nicht auf Details der einzelnen Anbieter eingegangen. Damit wollen wir die Chance auf eine Weiterentwicklung durch Dritte erhöhen, denn so lassen sich die Ergebnisse leichter auf andere Mobilitätssysteme bzw. -anbieter umlegen.

Es gibt mit der Web-Applikation (siehe Screenshot links) unter app.SharedMobility.ai ein interessantes Angebot für Citybike-NutzerInnen. Hier können sie unsere Vorhersagen abrufen und mit dem aktuellen Bestand aus den Echtzeitdaten vergleichen. Diese Web-App ist als Proof-of-Concept / Demonstration zu verstehen.



Um überhaupt ein Vorhersagemodell zu erstellen, braucht es ein gutes Datenmodell mit der notwendigen Abstraktion, gleichzeitig aber auch mit einem ausreichenden Detailgrad. Wir stellen eine Referenzimplementierung bereit, auf deren Basis auch andere EntwicklerInnen eigene Datensammlungen aufbauen können. [Unser Datenmodell](#) ist in der Praxis erprobt und ließ sich problemlos mit dem ML-Framework TensorFlow verwenden.

Die Vorhersagemodelle erstellten wir in zwei Programmiersprachen und mit jeweils einem anderen Framework. Zuerst starteten wir mit Scikit-learn für die Programmiersprache Python spezielle für die SeestadtFLOTTE und anschließend für Citybike Wien. Diese Modelle geben Auskunft über den Grad der Auslastung einer Station. Im weiteren Verlauf verwendeten wir das bekannte ML-Framework TensorFlow bzw. dessen JavaScript-Ableger TensorFlow.js zur Berechnung der komplexeren Citybike-Modelle. Für jede Citybike-Station wurden mehrere Vorhersagemodelle berechnet, teilweise unter enormen Rechenaufwand mit 60 virtuellen Cloud-CPU's im Parallelbetrieb über die Zeitspanne eines ganzen Tages.

Zu Beginn wollten wir uns auf Prognosen für die Seestadt und das dort über einen Mobilitätsfonds finanzierte Verleihsystem „SeestadtFLOTTE“ spezialisieren. Allerdings stellte sich nach einiger Zeit heraus, dass wir gerade in den Wintermonaten viel zu geringe Rohdaten gewinnen konnten und darauf aufbauend keine zielführenden Berechnungen möglich waren. Deshalb auch der spätere Fokus auf Citybike Wien als Mobilitätsanbieter.

Im Laufe der Umsetzung sind wir über das aspern.mobil LAB auch in direkten Kontakt mit der Forschungscommunity getreten. Wir liefern Einblicke in die SeestadtFLOTTE, einem modernen Bikesharing in der Seestadt Aspern (Wien-Donaustadt). War es anfangs unser vorrangiges Ziel ein Modell für die SeestadtFLOTTE zu entwickeln, so konnten wir während der Umsetzung schnell feststellen, dass die Flottengröße einfach zu klein ist für stabile Vorhersagen. Daher haben wir uns in den letzten drei Arbeitsmonaten vor allem auf Citybike Wien als Datengrundlage konzentriert. Deren Flotte umfasst rund 120 Stationen im Zentrum von Wien und ist bereits ein etabliertes Angebot, das deutlich mehr NutzerInnen aufweisen konnte als die relativ neue und räumlich stark beschränkte SeestadtFLOTTE.

Da uns ein besseres Bild der – oft als veraltet angesehen – stationsgestützten Services wichtig ist, publizierten wir auch einige professionelle Fotos unter der CC-BY-SA Lizenz. Damit können gute Fotografien auch für wissenschaftliche Publikationen im Bereich der Stadtplanung und Wohnbauforschung eingesetzt werden.

3 Verlauf der Arbeitspakete

3.1 Arbeitspaket 1 - *Detailplanung und Formales am Projektstart*

Es gab zu Beginn keine größeren Auffälligkeiten in diesem Arbeitspaket. Die Projektwebseite samt Domain wurde angelegt und initial befüllt. Außerdem wurden soweit die Blogbeiträge für die Webseite der netidee erstellt. Die formalen Anforderungen in diesem Arbeitspaket verliefen soweit reibungslos, auch wenn zu Beginn kleinere Korrekturen an der Arbeitsaufteilung vorgenommen wurden. Allerdings war der zu Beginn erstellte Zeitplan zu ambitioniert und wurde mehrmals angepasst. Ein Überblick der Soll- und Ist-Termine:

Zwischenbericht	8. April 2019	18. Juni 2019
Abschluss AP 4	19. April 2019	28. Juni 2019
Abschluss AP 5	21. Juni 2019	16. September 2019
Projektabschluss / AP 6	31. Juli 2019	1. Oktober 2019

3.2 Arbeitspaket 2 – *Öffentlichkeitsarbeit / Kommunikation / Grafik*

Das Logo und Icon wurden von [Lisa Vietze](#) soweit wie geplant erstellt und in mehreren Varianten umgesetzt. Sie hat dazu auch [einen Blogbeitrag](#) verfasst und die Hintergründe genauer erläutert. Die Grafiken und Fotos auf der Netidee-Webseite wurden entsprechend angepasst.

Das gesamte Corporate Design für SharedMobility.ai umfasst mehrere Einzelteile. Dazu gehört die Erstellung eines Logos in verschiedenen Anwendungsvarianten: 4c-Farben, schwarz-weiß, negativ. Darüber hinaus wurde ein ausführliches Corporate Design Manual entwickelt, das die weiteren Anwendungsmöglichkeiten genauer ausführt.

Die Entwicklung des Corporate Design war ein längerer Prozess als erwartet, da die eigentliche Zielgruppe genauer definiert werden musste und die tatsächliche Darstellung schließlich angepasst werden musste – da das finale Ergebnis von SharedMobility.ai vor allem Schnittstelle für Informationen ist, sollte dies auch im Logo widerspiegelt werden.

Das Logo hat einen Spagat zwischen modernem Design und technischem Auftritt geschafft – mehrere Varianten im Bezug auf Dynamik, Schriftart und Farben waren dabei unumgänglich und auch in einem so technischen Bereich sind persönlicher Geschmack und Befindlichkeiten nicht ausgeschlossen. Die Frage in wie weit das Corporate Design daher ein Erfolg ist, bleibt jede/m Nutzer/in selbst gestellt. Für uns als EntwicklerInnen hat es trotz einer intensiven Entwicklungsphase gut funktioniert.

Anschließend wurde das Design der Webseite angepasst und leicht adaptiert. Zusätzlich wurden die Grafiken auch im Github-Repository entsprechend eingebaut, um eine leichter Identifikation zu schaffen.

Auf der Projektwebseite wurden erste Texte erstellt, der Blogbereich angelegt und unter Beachtung der Canonical URL auch alle Blogbeiträge von der Netidee-Seite erneut veröffentlicht. Wir haben außerdem erste Kontakte zu MedienvertreterInnen aufgenommen, allerdings hat sich bislang leider noch kein Bericht dazu ergeben. Wir hoffen aber, dass sich nach Projektende noch einige Möglichkeiten für Erwähnungen oder Interviews ergeben.

3.3 Arbeitspaket 3 – *Data Ingestion*

Dieses Arbeitspaket umfasste die Einholung der Rohdaten von Verleihsystemen, deren Aufbereitung und Umwandlung in ein generalisiertes Datenformat. Wir konnten die Rohdaten der SeestadtFLOTTE über ein knappes Jahr aufbereiten und in unser Standardformat umwandeln. Herausforderung war die hohe Anzahl an Datensätzen, immerhin fallen pro Woche 81.000 Snapshots alleine für dieses Verleihsystem an.

Wir haben in dieser Phase nur die Quantität der Daten überprüft, also ob für jeden Tag die korrekte Anzahl an Snapshots vorhanden ist. Außerdem wurde nur eine Validierung des Rohdatenformats vorgenommen, nicht aber eine inhaltliche Prüfung. Im nachfolgenden Arbeitspaket 4 wird das zu einem Problem: Wir erkannten erst sehr spät, dass durch äußere Umstände über einen längeren Zeitraum im Verleihsystem zu wenige Fahrräder für aussagekräftige Vorhersagen vorhanden waren. Wir haben darauf später reagiert und als Fallback noch die Daten des Gewista-Systems „Vienna CityBikes“ über die Open Data Schnittstelle Citybik.es laufend mitgesichert.

Es wurde analysiert, welche Informationen unterschiedliche Verleihsysteme bekanntgeben und wie diese in ein übergreifendes Datenmodell abstrahiert werden können. Das erarbeitete generalisierte Datenformat für Stations-Snapshots geht auch auf während der Entwicklung aufgedeckte Probleme ein. So müssen Snapshots stets mit der korrespondierenden Zeitzone abgespeichert werden, ansonsten können später weitere Datensätze, z.B. Wetterdaten oder Verkehrsinformationen, nur unzureichend und mit manuellen Eingriffen mit den Snapshots zusammengeführt werden. Zeitzonen dürfen dabei nicht als absolute Abweichung in Minuten (z.B. +120 bzw. +02:00) abgespeichert werden, sondern nur über ihren Zeitzonen Identifier (z.B. „Europe/Vienna“), ansonsten kann es bei historischen Auswertungen in Folge zu unnötigen Heuristiken / Konvertierungsskripten kommen.

3.4 Arbeitspaket 4 – *AI Modelling*

Während die erste Aufbereitung der Daten, die Erstellung eines Datenmodells und der prinzipielle Austausch der Daten rückblickend sehr gut voranschritten, gab es besonders bei der Erstellung der Prognosemodell doch deutlich mehr Arbeit als ursprünglich angenommen. Wir haben uns außerdem entschieden mehr in kurzen intensiven Sprints zu arbeiten, statt über einen längeren Zeitraum kontinuierlich kleinere Arbeitspakete zu verrichten.

Der erste Schritt zur Modellierung war die Evaluierung der Daten, die mit den statischen Methoden durchgeführt wurde. Es wurden dazu die NumPy und Pandas Python's Bibliotheken verwendet. Mit Hilfe von Matplotlib wurden auch Grafiken erstellt, die bei der Erkennung der täglichen Muster hilfreich gewesen sind. Nach dem Schritt war auch noch Literaturrecherche notwendig um zu entscheiden, welches AI-Modell am besten für Vorhersagen geeignet ist.

Letztendlich wurde die Entscheidung getroffen, dass die ersten Vorhersagen mit einem Multilayer Perceptron (MLP) Modell geliefert werden. Für das Implementieren vom neuronalen Netz wurde die freie Software Maschinelles Lernen Bibliothek Scikit-learn für die Programmiersprache Python verwendet. Über die Keras-API kann man die so ermittelten Modelle später jederzeit in TensorFlow-kompatible Modelle serialisieren.

Folgende Schritte wurden mit Scikit-learn abgearbeitet:

- scikit-learn zur ersten Evaluierung der Daten
- Tensorflow-Umgebung aufsetzen
- Aus Datenbank ein Trainings- und Testdatenset generieren
- AI-Modelle generieren und validieren
- Erweiterte Input-Variablen mit geeigneten Open Data Datensätzen finden (Wetter aus offenen Quellen, Verkehrsdaten, etc.)

Im nichtlinearen Regressionsmodell ist die Anzahl der verfügbaren Räder die abhängige Variable. Mehrere Eingangsfaktoren und deren Kombinationen sind als unabhängige Variablen getestet worden. Auch die Struktur des neuronalen Netzes ist in mehreren Konfigurationen genauer untersucht worden.

Für die Berechnungen mit TensorFlow bzw. TensorFlow.js sind wir auf virtuelle Server auf Basis der Google Cloud Platform umgestiegen. Dies war notwendig, da der Berechnungsaufwand für 120 Citybike-Stationen einfach nur mehr mit parallelen CPUs in akzeptabler Zeit zu bewältigen war. Wir empfehlen die Verwendung eines CPU- und Memory-optimierten Maschinentyps in einer europäischen Region.

Für unsere Berechnungen haben wir im ersten Schritt den Typ c2-standard-60 mit 60 virtuellen CPUs (Intel Xeon bei 3.8 GHz) und 240 GB an Speicher verwendet. Dafür wurde die TensorFlow.js-Anwendung in 60 parallelen Prozessen gestartet und für jede Station ein eigenes Modell berechnet.

Wir haben für jede Citybike-Station folgende vier Modelle errechnet und publiziert:

- **adagrad-192-150-epochs100**
Adagrad-Optimizer mit 3 Layer und 38.962 trainierbaren Parametern in 100 Epochen.
- **adam-150-80-500epochs-64batch-learning0_001**
Adam-Optimizer mit 3 Layer und 19.754 trainierbaren Parametern, aber einer deutlich kleineren Lernrate, dafür statt 150 mit 500 Durchläufen und einer kleineren Batchgröße von 64.
- **adam-192-150-epochs100**
Adam-Optimizer mit 3 Layer und 38.962 trainierbaren Parametern in 100 Epochen.
- **rmsprop-192-150-epochs25**
rmsprop-Optimizer mit 3 Layer und 38.962 trainierbaren Parametern in 25 Epochen. Diese Konfiguration hat sich für die meisten Citybike-Station als die beste Variante erwiesen.

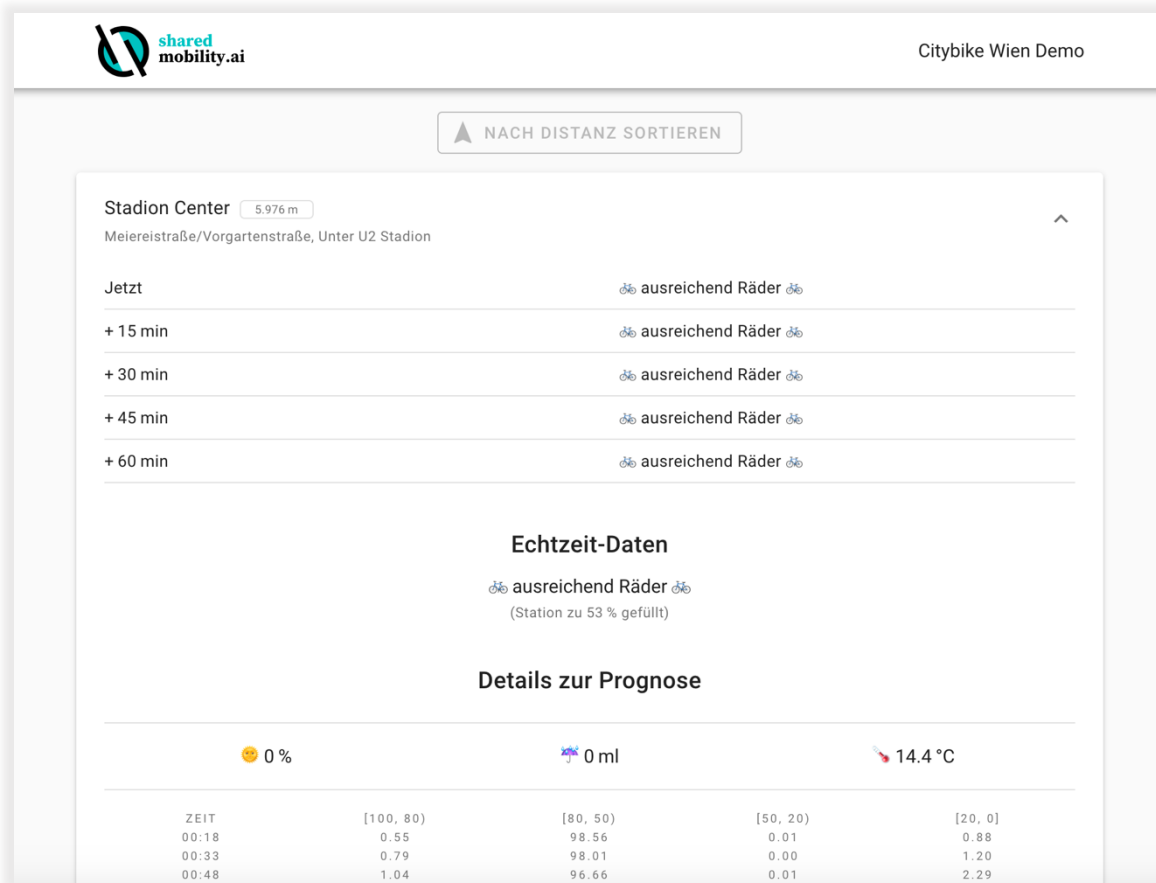
In dem Modell sind reine Stationsdaten und Open Data Wetterwerte der ZAMG als Eingangsdaten verwendet worden. Es sollten aber weitere Variablen inkludiert werden, u.a. generelle Verkehrsdaten und simple Wetterprognosen, um auch die Erwartungshaltung der Nutzerschaft abzubilden. Außerdem berücksichtigen die Modelle keinerlei Besonderheiten des Anbieters, z.B. Wartungsintervalle oder Radtransporte zwischen „heißen“ Stationen mit hoher Frequenz.

3.5 Arbeitspaket 5 – *Web-Schnittstelle*

Die Web-API wurde mit Node.js und dem Framework Express umgesetzt und ist auf einen virtuellen Server in der Google Cloud Platform gehostet. Sie ist auch öffentlich erreichbar und unter der URL <https://api.sharedmobility.ai/> abrufbar. Im Hintergrund verwendet diese Schnittstelle die im smai-public-

models Bucket veröffentlichen Modelle. Daher kann die Umsetzung auch als Referenzimplementierung für andere EntwicklerInnen dienen. Sie können so mit TensorFlow.js ihre eigenen Schnittstellen anbieten und unsere grundsätzliche Architektur als Vorbild nehmen.

Ursprünglich war vorgesehen diese Schnittstelle im [Seestadt.bot](#) zu verwenden. Da dies aber durch den „Ausfall“ der SeestadtFLOTTE als Datengrundlage nicht möglich war, haben wir uns kurzerhand dazu entschieden, eine Citybike-Miniapplikation zu erstellen. Diese wurde mit Vue.js und Vuetify umgesetzt und für mobile Nutzung am Smartphone optimiert. Sie steht unter [app.SharedMobility.ai](#) zur Verfügung.



3.6 Arbeitspaket 6 – Projektabschluss

Wir haben in diesem Arbeitspaket die Zukunft des Projekts und der Idee dahinter abgesichert. Hierfür wurden einerseits die entsprechenden Hosting-Ressourcen auf längere Zeiträume vorbereitet (unser Dienst sollte für zumindest vier weitere Jahre problemlos online abrufbar bleiben), andererseits wurde die Dokumentation erweitert. Es steht nun eine umfassende und detaillierte [EntwicklerInnen-Dokumentation auf unserer Projektwebseite](#) zur Verfügung, in der wir auch die unterschiedlichen Kostenaspekte einer cloud-basierten Berechnung transparent auflisten. Außerdem haben wir die Code-Dokumentation selbst verbessert, um auch eine Nachvollziehbarkeit zu gewährleisten bzw. Wartung der Software zu ermöglichen.

4 Liste Projektendergebnisse

1	Projektzwischenbericht	CC-BY-SA	Netidee-Projektseite
2	Projektendbericht	CC-BY-SA	Netidee-Projektseite
3	EntwicklerInnen-Dokumentation	CC-BY-SA	Link SharedMobility.ai
4	AnwenderInnen-Dokumentation <i>Wir stellen kein klassisches Produkt für EndanwenderInnen bereit. Jedoch gibt es eine Demo-Applikation für Citybike Wien NutzerInnen.</i>	CC-BY-SA AGPLv3	Demo: app.SharedMobility.ai Dokumentation: Installationsbeschreibung
5	Kurzzusammenfassung	CC-BY-SA	Netidee-Projektseite
6	Externe Kommunikation	CC-BY-SA	Netidee-Projektseite
7	TensorFlow-Modelle (inkl. von Keras konvertierte Modelle)	AGPLv3	Github (Code) GCS Bucket (Modelle)
8	Backend / Data-Ingest	AGPLv3 CC-BY-SA	Github (Code) GCS Bucket (Rohdaten)
9	<u>Frontend:</u> Öffentliche REST-Schnittstelle & Demo-Applikation / Web-Frontend	AGPLv3	Github (Code REST-API) Github (Code Web-App)
10	Testdaten für Citybike Wien	CC-BY-SA	GCS Bucket Daten-Dokumentation
11	Grafiken und Fotos zu Citybikes und SeestadtFLOTTE	CC-BY-SA	Projektwebseite

5 Verwertung der Projektergebnisse in der Praxis

EntwicklerInnen können unsere REST-API jederzeit für eigene Mobilitäts-Apps verwenden. Sie ist unter <https://api.SharedMobility.ai> erreichbar und [ausführlich dokumentiert](#).

Unsere Web-Applikation ist für Citybike-NutzerInnen jederzeit nutzbar. Sie wurde auch bereits von einigen RadfahrerInnen genutzt: <https://app.SharedMobility.ai>

Für die Forschungscommunity (u.a. dem aspern.mobil LAB) stellen wir alle aggregierten Daten und die TensorFlow-Modelle öffentlich und dauerhaft über Google Cloud Storage Buckets zur Verfügung. Eine detaillierte Übersicht dazu findet sich [auf der Projektseite](#).

6 Öffentlichkeitsarbeit / Vernetzung

Folgende Blog-Postings wurden im Projektverlauf veröffentlicht:

- [SharedMobility.ai – Mobilität intelligent teilen](#)
- [Datensammeln und Aufbereitung](#)
- [Logoentwicklung für SharedMobility.ai](#)
- [Verkehrsströme in der Seestadt](#)
- [Lessons Learned: Vier Fehler in unserer Projektplanung](#)
- [Auf dem Weg zu einem AI-Modell](#)
- [Lernen mit Multi-layer Perceptrons \(MLP\)](#)
- [Neuronale Netze mit TensorFlow.js](#)

Außerdem versuchen wir auch über Twitter und Instagram unser Projekt bekannter zu machen:

[Twitter-Suche „from:botic netidee“](#) | Instagram-Stories verschwinden nach 24 Stunden.

Es wurden zahlreiche professionell fotografierte Bilder rund um Citybike und SeestadtFLOTTE unter CC-BY-SA veröffentlicht. <https://SharedMobility.ai/kostenfreies-bildmaterial/>

Wir haben unser Projekt auch beim [aspermobil LAB](#) zweimal kurz vorgestellt und einen Ausblick auf die Implementierung im [Seestadt.bot](#) gegeben. Als diese durch den Wechsel auf die Citybike-Daten nicht mehr realistisch war, wurde im Zuge einer Mobillab-Stunde über die Ursachen diskutiert und wie man das lokale Verleihsystem in der Seestadt verbessern kann. Die gewonnenen Erfahrungen wurde daher im Rahmen vom aspermobil LAB dokumentiert und für wissenschaftliche Zwecke festgehalten. Außerdem wird es weiter einen laufenden Austausch mit dem Forschungsteam des Mobillabs geben. Ein nächster Termin ist bereits eine Woche nach Projektabschluss avisiert.

7 Eigene Projektwebsite

Wir betreiben unter [SharedMobility.ai](#) eine Projektwebseite.

Die Citybike Wien Demo ist unter [app.SharedMobility.ai](#) abrufbar.

8 Geplante Aktivitäten nach netidee-Projektende

Die Sammlung und Aufbereitung der Daten in TensorFlow-kompatible Datasets wird fortgeführt und die entsprechende Server-Infrastruktur weiterhin genutzt. Die Datensammlung für Citybike Wien läuft normal weiter und wird auch über die Wintersaison fortgesetzt. Unser aktueller Plan sieht vor, ein komplettes Jahr an Datasets abzudecken. Hierfür haben wir vorab die Kapazitäten des Sammel-Services entsprechend erweitert und alle Vorkehrungen (Domain, Cloud-Speicherplatz, vServer) schon während der letzten Projektphase getroffen. Durch den mit 3.000 Euro dotierten und unserem Projekt zuerkannten netidee-Sonderpreis für Mobilität sind auch die Mehrkosten für die kommenden Jahre abgedeckt und entsprechend z.B. schon die Domain auf weitere 4 Jahre bereits reserviert.

Vor dem Abschluss vom AP 4 (AI-Modelling) war noch kein Release von TensorFlow 2.0 absehbar, lediglich ein Release Candidate war vorhanden. Wir planen mit der Freigabe von TensorFlow 2.0 auch unsere Modelle auf diese neue Version anzupassen. Damit werden die verwendeten JSON-serialisierten Modelle für einen deutlich längeren Zeitraum nutzbar bleiben. Die entsprechenden Adaptionen sollten in einem überschaubaren Ausmaß sein, da viele der API und Formate bereits mit Blick auf TensorFlow 2.0 konzipiert wurden.

In der Kommunikation nach außen werden wir weiterhin auf persönlichen Social Media Kanälen über die Publikation von neuen Datasets, aber auch Entwicklungen im Bereich Shared Mobility, berichten. Wir hoffen, dass jemand unsere Idee und Datasets für eigene Projekte aufgreift und so zu einer weiteren Verbesserung der Vorhersagemodelle führt. Hierfür wird auch noch in den Wochen nach Projektabschluss die Dokumentation nach offenen Fragen durchsucht und um ein kurzes Anwendungsbeispiel erweitert werden.

Bereits fixiert ist ein weiteres Treffen und eine punktuelle Zusammenarbeit mit dem [aspern.mobil LAB](https://www.aspernmobil.at/). Das vom bmvit geförderte F&E-Projekt in der Seestadt Aspern forscht an der Etablierung einer neuen urbanen Mobilität. Es werden einerseits die von uns über die lokale SeestadtFLOTTE gewonnenen Informationen rezipieren, andererseits wollen wir uns auch über Shared Mobility Angebote regelmäßig austauschen. In Folge könnten sich weitere Kooperationen oder Perspektiven zur Weiterentwicklung aufbauen.

9 Anregungen für Weiterentwicklungen durch Dritte

Wir haben alle Daten und bereits fertig aufbereitete Datasets und unmittelbar einsetzbare TensorFlow-Modelle öffentlich und direkt einsetzbar publiziert. Sie befinden sich in folgenden Google Cloud Storage Buckets:

- [smai-public-assets](#) – Logos, Bildmaterial und Projektartefakte.
- [smai-public-datasets](#) – TensorFlow-kompatible Datasets zur direkten Verwendung.
- [smai-public-models](#) – komplette JSON-seralisierte TensorFlow-Modelle.

Die Datasets können direkt als Input-Pipeline zur Erstellung von neuen TensorFlow-Modellen verwendet werden. Sie sind im Grunde voll aufbereitete CSV-Dateien, die in unserem Fall eine denormalisierte Form der SMAI-Datenbank enthalten. Interessierte können so mit der [tf.data.Dataset](#)-API direkt eigene Vorhersagemodelle für Citybike Wien erstellen.

Die öffentlichen TensorFlow-Modelle können direkt mit TensorFlow.js geladen werden. So kann jede/jeder eine eigene Applikation erstellen, die direkt unsere Vorhersagemodelle einsetzt.

Zur schnellen Nutzung steht eine REST-API unter [api.SharedMobility.ai](#) bereit. Wir haben [eine detaillierte Dokumentation](#) über die einzelnen Methoden erstellt und die genaue Verwendung kann [in der Demo-Applikation](#) nachvollzogen werden.

Es bleibt abzuwarten, ob auch Projektergebnisse direkt vom aspern.mobil LAB verwendet werden können. Wir sind jedenfalls in einem kontinuierlichen Austausch mit dem Forschungsteam.