Richtlinie zum „Software-Sprint“

FedCast – Federated Learning for Time-Series Forecasting

Schlussbericht

Zuwendungsempfänger:

Nicolas Kuhaupt

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Forschung, Technologie und Raumfahrt unter dem Förderkennzeichen 16IS25S11 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei dem Autor.

Kurze Darstellung der Aufgabenstellung und Motivation in der Förderphase

Motivation und Problemstellung

Die zunehmende Digitalisierung und Vernetzung unserer Lebensbereiche erzeugt eine immense Menge an Zeitreihendaten, die wertvolle Informationen für personalisierte Dienste und datenbasierte Entscheidungen bergen. Ob im Gesundheitswesen zur individuellen Therapieplanung, in der Energieversorgung zur Optimierung des Verbrauchs oder in der Industrie zur vorausschauenden Wartung von Maschinen – präzise und personalisierte Vorhersagen sind von entscheidender Bedeutung.

Bisherige Ansätze erfordern jedoch oft die zentrale Speicherung sensibler Daten, was nicht nur ein erhebliches Risiko für den Datenschutz darstellt, sondern auch das Vertrauen der Nutzer in solche Technologien untergräbt. FedCast adressiert diese Herausforderung, indem es eine datenschutzfreundliche Alternative basierend auf Federated Learning bietet. Durch die dezentrale Verarbeitung der Daten auf den Geräten der Nutzer und den Austausch lediglich aggregierter Modellupdates wird die Privatsphäre geschützt und gleichzeitig das Potenzial personalisierter Anwendungen voll ausgeschöpft.

Geplante Vorgehensweise

FedCast wurde als modulares, in Python entwickeltes Open-Source-Toolkit mit einer klaren API realisiert. Die modulare Architektur ermöglicht eine flexible Kombination verschiedener Algorithmen und Methoden. Konkret wurden Komponenten für verschiedene Aspekte des personalisierten Federated Learning für Zeitreihen Prognosen implementiert.

Wichtigste Meilensteine

1. **Implementierung und umfassende Validierung der Kernkomponenten** für personalisiertes Federated Learning von Zeitreihendaten mit verschiedenen Personalisierungsstrategien
2. **Integration und Evaluation robuster Aggregationsmethoden** und adaptiver Lernmechanismen für unterschiedliche Szenarien
3. **Entwicklung einer detaillierten Dokumentation**, praxisnaher Tutorials und aussagekräftiger Anwendungsbeispiele
4. **Veröffentlichung des stabilen und gut dokumentierten Toolkits** als Open-Source-Projekt auf GitHub und Präsentation auf relevanten Fachveranstaltungen

Beitrag des Projektes zu den Zielen der Förderinitiative „Software Sprint“

Bezug zu Infrastruktur

FedCast trägt zur **Infrastruktur** bei, indem es ein Open-Source-Framework bereitstellt, das als grundlegende Software-Infrastruktur für die Forschung und Entwicklung im Bereich des Federated Learning mit Zeitreihendaten dient. Das Framework bietet:

- **Wiederverwendbare Komponenten**: Modulare Architektur mit austauschbaren Datensätzen, Modellen und Aggregationsstrategien
- **Standardisierte Schnittstellen**: Klare APIs für die Integration in bestehende ML-Workflows
- **Reproduzierbare Experimente**: MLflow-Integration für umfassendes Experiment-Tracking und Vergleichbarkeit
- **Skalierbare Architektur**: Aufbau auf dem bewährten Flower-Framework, das sowohl für kleine als auch große Federated Learning Szenarien geeignet ist

Bezug zu Sicherheit

FedCast adressiert **Sicherheit** durch datenschutzfreundliche Technologien:

- **Privacy-by-Design**: Daten bleiben dezentral auf den Geräten der Nutzer, nur Modellupdates werden aggregiert
- **Keine zentrale Datenspeicherung**: Sensible Zeitreihendaten werden nie zentral gesammelt
- **Robuste Aggregationsmethoden**: Implementierung von Strategien, die auch bei unvollständigen oder manipulierten Daten funktionieren
- **Transparenz durch Open Source**: Vollständige Nachvollziehbarkeit der Algorithmen und Datenschutzmechanismen

Weitere gesellschaftlich relevante Ziele

- **Förderung von Innovation**: Ermöglicht Forschern und Entwicklern, datenschutzfreundliche ML-Lösungen zu entwickeln
- **Anwendungen in kritischen Bereichen**: Unterstützt Anwendungen in Medizin, Finanzwesen, Energieversorgung und IoT, wo Datenschutz essentiell ist

Ausführliche Darstellung der Ergebnisse in der Förderphase

Implementierte Federated Learning Strategien

Das Framework implementiert **9 verschiedene Aggregationsstrategien** für Federated Learning:

1. **FedAvg** (Federated Averaging) - Basisstrategie für standardmäßige Aggregation
2. **FedProx** - Proximale Regularisierung zur Behandlung von Heterogenität
3. **FedTrend** - Zeitreihen-bewusste Strategie mit Trajektorien-Informationen
4. **FedLAMA** - Layer-wise Adaptive Model Aggregation, reduziert Kommunikationsaufwand um bis zu 70%
5. **FedNova** - Normalized Averaging zur Lösung der Objective Inconsistency in heterogenen Umgebungen
6. **FedDyn** - Dynamische Regularisierung zur Behandlung von Client-Drift
7. **SCAFFOLD** - Stochastic Controlled Averaging mit Control Variates
8. **FedOpt** - Adaptive Federated Optimization mit Adam, AdaGrad und Yogi
9. **Partial Sampling** - Partielle Client-Auswahl für Kommunikationseffizienz

Implementierte Datensätze

Das Framework unterstützt **7 verschiedene Zeitreihendatensätze** aus unterschiedlichen Domänen:

1. **Synthetischer Sinus-Datensatz** - Für Simulationen und Tests mit kontrollierter Heterogenität
2. **MIT-BIH ECG-Datensatz** - Medizinische Zeitreihen von 48 Patienten (PhysioNet)
3. **Electricity Load Dataset (ELD)** - Energiedaten für Smart Grid Anwendungen
4. **Intel Berkeley IoT-Sensor-Datensatz** - Temperaturdaten von 54 Sensoren
5. **UNSW-NB15 Netzwerkverkehr-Datensatz** - 2.5 Millionen Netzwerkflow-Datensätze mit 10 Verkehrskategorien
6. **S&P 500 Aktien-Datensatz** - Finanzielle Zeitreihen von 50 Unternehmen (Yahoo Finance)
7. **210 US-Wetterstationen-Datensatz** - Meteorologische Daten von 50 NOAA-Wetterstationen

Modellarchitekturen

- **MLP (Multi-Layer Perceptron)** - Neuronales Netzwerk mit mehreren versteckten Schichten
- **Linear Model** - Einfaches lineares Modell für Baseline-Vergleiche
- **Modulares Modellsystem** - Erweiterbares Framework für benutzerdefinierte Architekturen

Experiment- und Evaluationsinfrastruktur

- **MLflow-Integration**: Umfassendes Experiment-Tracking mit automatischer Protokollierung von Parametern, Metriken und Artefakten
- **Automatische Visualisierung**: Generierung von Verlaufsplots für Verluste und Metriken pro Runde
- **Grid-Experimente**: Automatisierte Ausführung aller Kombinationen von Datensätzen, Modellen und Strategien
- **Umfassende Tests**: Test-Suite für alle Datensätze und Strategien zur Qualitätssicherung

Erreichte Meilensteine

Alle geplanten Meilensteine wurden erfolgreich erreicht:

1. **Kernkomponenten implementiert und validiert**: Alle geplanten Personalisierungsstrategien und Aggregationsmethoden wurden implementiert und umfassend getestet
2. **Robuste Aggregationsmethoden integriert**: 9 verschiedene Strategien implementiert, inklusive spezialisierter Ansätze für Zeitreihen (FedTrend, FedLAMA)
3. **Dokumentation und Tutorials entwickelt**:

- Umfassendes README mit Architektur-Übersicht
- Tutorial-System mit strukturierter Lernprogression
- Installation & Setup Tutorial
- "Your First Federated Experiment" Tutorial
- Contribution Guide (CONTRIBUTING.md)

4. **Open-Source-Veröffentlichung**:
- Projekt auf GitHub veröffentlicht (https://github.com/NKDataConv/FedCast)
- PyPI-Package erstellt und als Beta-Version (0.1.1b1) veröffentlicht
- GitHub Pages Dokumentation eingerichtet
- auf Fachkonferenz MLCon am 26.11.25 vorgestellt

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Zielgruppe

Die Zielgruppe des Projekts sind **Forscher_innen, Entwickler_innen und Data Scientists** im Bereich Machine Learning und Zeitreihenanalyse, die Federated Learning in ihren Projekten einsetzen oder evaluieren möchten. Dies umfasst:

- **Akademische Forscher**: Wissenschaftler, die Federated Learning für Zeitreihenprognosen erforschen
- **Industrieentwickler**: Entwickler in Unternehmen, die datenschutzfreundliche ML-Lösungen implementieren möchten
- **Data Scientists**: Praktiker, die Zeitreihenmodelle auf verteilten Daten trainieren müssen
- **Studenten**: Lernende, die Federated Learning praktisch erlernen möchten

Nutzen für die Zielgruppe

Sofortiger Nutzen

- **Sofort einsatzbereites Framework**: Keine Notwendigkeit, Federated Learning von Grund auf zu implementieren
- **Vorgefertigte Datensätze**: 7 verschiedene Zeitreihendatensätze aus verschiedenen Domänen, sofort verwendbar und zum Experimentieren geeignet
- **Vergleichbare Strategien**: 9 implementierte Strategien ermöglichen direkten Vergleich verschiedener Ansätze
- **Reproduzierbare Experimente**: MLflow-Integration ermöglicht vollständige Reproduzierbarkeit

Langfristiger Nutzen

- **Wissensbasis**: Das Framework dient als Referenzimplementierung für Federated Learning mit Zeitreihen
- **Forschungsgrundlage**: Ermöglicht vergleichende Studien und Benchmarking
- **Lernressource**: Tutorials und Dokumentation unterstützen den Wissenstransfer

Weitergehende Effekte durch Open-Source-Stellung

- **Transparenz**: Vollständige Nachvollziehbarkeit der Algorithmen und Implementierungen
- **Gemeinschaftliche Weiterentwicklung**: Andere Entwickler können das Projekt erweitern und verbessern
- **Wissensverbreitung**: Open-Source-Verfügbarkeit fördert die Verbreitung von Federated Learning Technologien
- **Standardisierung**: Kann als Grundlage für Standardisierungsbemühungen dienen

Ideen für Weiterentwicklung

Kurzfristige Weiterentwicklungen (nächste 6 Monate)

1. **Erweiterte Modellarchitekturen**:
   - LSTM/GRU für sequenzielle Zeitreihenprognosen
   - Transformer-basierte Modelle für langfristige Abhängigkeiten
   - Attention-Mechanismen für wichtige Zeitpunkte

2. **Zusätzliche Datensätze**:
   - Weitere medizinische Datensätze (z.B. MIMIC-III)
   - Weitere finanzielle Datensätze (Kryptowährungen, Devisen)
   - Industrielle Sensordaten (Predictive Maintenance)

3. **Erweiterte Personalisierungsstrategien**:
   - Meta-Learning Ansätze (MAML, Reptile)
   - Transfer Learning zwischen Clients
   - Multi-Task Learning

Mittelfristige Weiterentwicklungen (6-12 Monate)

1. **Asynchrones Federated Learning**:
   - Unterstützung für asynchrone Kommunikation zwischen Clients
   - Staleness-Toleranz-Mechanismen

2. **Differential Privacy Integration**:
   - Integration von Differential Privacy Mechanismen
   - Privacy-Budget-Management

3. **Federated Learning für Streaming-Daten**:
   - Online-Learning Fähigkeiten
   - Adaptive Modellaktualisierung in Echtzeit

4. **Erweiterte Evaluationsmetriken**:
   - Zeitreihen-spezifische Metriken (MASE, sMAPE)
   - Kommunikationskosten-Analyse
   - Fairness-Metriken

Langfristige Vision (12+ Monate)

1. **Production-Ready Features**:
   - Deployment-Tools für Edge-Devices
   - Monitoring und Alerting-Systeme

- Skalierung auf tausende von Clients

2. **Multi-Modal Federated Learning**:
   - Unterstützung für verschiedene Datentypen (Bilder, Text, Zeitreihen)
   - Cross-Modal Learning

3. **Federated Learning für Graph-Daten**:
   - Graph Neural Networks in Federated Settings
   - Zeitreihen auf Graphen

Persönliche Weiterentwicklung

Die Arbeit an diesem Projekt hat mich in meiner persönlichen und fachlichen Weiterentwicklung unterstützt:

Fachliche Weiterentwicklung

- **Einarbeitung in Federated Learning**: Tiefgreifendes Verständnis der theoretischen Grundlagen und praktischen Herausforderungen des Federated Learning
- **Zeitreihenanalyse**: Umfassende Kenntnisse in Zeitreihenprognosen und deren spezifische Anforderungen
- **Software-Architektur**: Entwicklung von Fähigkeiten im Design modularer, erweiterbarer Frameworks
- **Open-Source-Entwicklung**: Erfahrung in Best Practices für Open-Source-Projekte, Community-Management und Dokumentation

Impulse durch Austausch

- **Demo Day**: Der Austausch beim Demo Day bot wertvolle Einblicke in andere Projekte und ermöglichte Feedback von Experten und Kollegen
- **Konferenzpräsentation**: Die Präsentation des Projekts auf der Fachkonferenz MLCon ermöglichte Austausch und konstruktive Diskussionen mit Data Scientist aus dem Bereich Machine Learning.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Im Rahmen der Entwicklung des FedCast-Frameworks wurden diverse Aggregationsmethoden für das Federated Learning implementiert. Hier gab es viele Ideen, und es konnte auch auf viele bereits bestehende Ideen aus wissenschaftlichen Veröffentlichungen zurückgegriffen werden.

Nicht weiterverfolgte Ansätze

Dabei bin ich mehrfach auf Ansätze gestoßen, die zu komplex waren, als dass ich sie im Rahmen dieses Projekts implementieren wollte. Beispiele hierfür sind:

1. **Komplexe Meta-Learning Ansätze**: Einige fortgeschrittene Meta-Learning Methoden (z.B. komplexe Varianten von MAML) erforderten erhebliche Implementierungsaufwände und zusätzliche Abhängigkeiten, die den Rahmen des Projekts überschritten hätten.

2. **Homomorphic Encryption Integration**: Während Homomorphic Encryption ein interessanter Ansatz für zusätzliche Privatsphäre-Garantien ist, wäre die Integration sehr komplex gewesen. Außerdem sind diese Methoden außerhalb des Fokus des FedCast Projektes.

Alternative Vorgehensweise

Durch die Fülle an Ideen und verfügbaren Methoden war es jedoch nicht zwangsweise notwendig, diese komplexeren Ansätze zu implementieren. Stattdessen konnte ich auf bewährte und gut dokumentierte Methoden zurückgreifen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer*innen

Interessent*innen können sich detailliert über die Projektergebnisse informieren über:

Hauptquellen

- **GitHub Repository**: https://github.com/NKDataConv/FedCast
  - Vollständiger Quellcode
  - Issue-Tracking
  - Contribution Guidelines
  - Release-Informationen

- **PyPI Package**: https://pypi.org/project/fedcast/
  - Installierbares Python-Package
  - Package-Dokumentation
  - Versionshistorie

- **Projektdokumentation**: https://nkdataconv.github.io/FedCast
  - GitHub Pages Website
  - Tutorials und Anleitungen

Weitere Ressourcen

- **README.md**: Umfassende Projektübersicht direkt im Repository
- **Tutorials**: Schritt-für-Schritt Anleitungen für erste Experimente
- **Beispiel-Experimente**: Vorgefertigte Experiment-Skripte im Repository
- **Test-Suite**: Umfassende Tests als Beispiele für die Verwendung der Komponenten

Kontaktmöglichkeiten

- **GitHub Issues**: Für Fragen, Bug-Reports und Feature-Requests
- **Repository Discussions**: Für Diskussionen und Community-Austausch

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Projektplanung konnte im Wesentlichen wie vorgesehen eingehalten werden. An ein paar Stellen wurden Aufwände zu klein eingeschätzt, an anderen Stellen zu groß.

Bei den Aggregationsmethoden hätte man beliebig viele implementieren können, da es eine große Fülle an wissenschaftlichen Veröffentlichungen zu verschiedenen Federated Learning Strategien gibt. Dadurch war etwas Flexibilität in der Planung gegeben. Mit den implementierten 9 Strategien wurde das gesetzte Mindestziel übertroffen.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Während der Projektlaufzeit gab es keine Entwicklungen anderer Personen oder Institutionen, die einen wesentlichen Einfluss auf die Arbeiten und die Zielsetzung des FedCast-Projekts hatten.