There is no limit to the amount of data a connected vehicle produces every day. Even a very conservative estimate of a few dozen gigabytes per vehicle per hour for a small fleet of cars shows that it is infeasible to perform centralized data processing. The reason is that it is not efficient to transfer big data via the network. Instead, alternative approaches for distributed data analysis are needed. Federated Learning is one such example of decentralized data processing.

**Contribution**

- Creating an Erlang-based prototype of a framework for distributed data processing
- Highlighting the feasibility of a purely functional style for the aforementioned framework
- Creating a purely functional implementation of an artificial neural network in Erlang
- Conducting a performance comparison of a Federated Learning implementation exclusively in Erlang with one in which client processes depend on computations being executed in C
- Demonstrating the suitability of Erlang for decentralized machine learning tasks via performance comparisons

**Implementation**

We implemented a general framework in Erlang for distributing machine learning tasks and added an implementation of an artificial neural network to it. We created four prototypes, using the following approaches: (1) concurrent Erlang, (2) concurrent Erlang plus NIFs, (3) distributed Erlang, (4) distributed Erlang and C nodes. As the source code on the right shows, message passing is a very good fit for distributed machine learning in general, and Federated Learning in particular. The code is discussed further in our paper. Here, we draw attention to how elegantly the general framework can be implemented in Erlang.

**Future Work**

Building on this research, future avenues to pursue are, first, to scale the performance comparison and determine at which point a collection of edge devices outperforms a powerful server. Second, one could implement other machine learning algorithms and assess how well they map to distributed computing. Third, there is the issue of scaling when using very large data sets, which one may want to look into. Lastly, we could like to point out that the results of this paper led to currently ongoing work on designing and implementing a real-world framework for distributed data analysis in our research lab, written in Erlang and Python. We intend to share details in a future paper.

$
\begin{align*}
\text{Concurrent Execution} & \\
\text{Time in s} & \\
\text{Erlang-only} & \\
\text{Erlang + NIFs} & \\
0 & 1,000 & 2,000 & 3,000 & 4,000 & 5,000 & 6,000 & 7,000 & 8,000 & 9,000 & 10,000 & 11,000 & 12,000 \\
\text{Erlang-only} & \\
\text{Erlang + NIFs} & \\
0 & 1,000 & 2,000 & 3,000 & 4,000 & 5,000 & 6,000 & 7,000 & 8,000 & 9,000 & 10,000 & 11,000 & 12,000 \\
\end{align*}
$

**Acknowledgements**

Our interns Adrian Nilsson and Simon Smith assisted with the implementation. This research was supported by the project Onboard/Offboard Distributed Data Analytics (OODIDA) in the funding program FFI: Strategic Vehicle Research and Innovation (DNR 2016-04260), which is administered by VINNOVA, the Swedish Government Agency for Innovation Systems.

**Source Code**

Source code artifacts accompanying this paper are available at [https://gitlab.fraunhofer-chalmers-centro/federated_learning_erlang](https://gitlab.fraunhofer-chalmers-centro/federated_learning_erlang), released under the MIT license.