While security is a significant concern for almost everyone worldwide, network or website server managers must be even more cautious. Allowing intentionally or accidentally malicious requests into a network can potentially destroy data, cause outages, or leak sensitive data. This can cause irreparable damage to a company’s reputation and success, or worse, individual lives. Therefore, ensuring harmful requests cannot enter specific networks is critical.

While network monitoring systems are in place, there is always room for improvement. Moreover, these systems are often costly, and the free or low-cost alternatives may not offer the same detection level. This leaves individuals and small businesses at a higher risk of attacks due to financial constraints. Therefore, there is a pressing need for more open-source research and experiments in the field of network request detection. This will not only expand the detection range and accuracy but also level the playing field for those with limited resources.

My experiments and models can be used directly for malicious network detection or combined with existing open-source systems to improve performance. Even if the models created in these experiments are not used directly in any tools, the results from comparing different supervised learning methods can be helpful to other data scientists in their choices for future detection system creation or improvements. Secondarily, there may be trends in the data found during experiments that could inform the decisions of network security specialists or improve other detection algorithms.

The data was found on [Kaggle.com](https://www.kaggle.com/datasets/rudrakumar96/web-firewall-good-and-bad-request/data) as a recently uploaded dataset. An individual has captured it using a scanning tool, and it includes over 8,000 instances. While it is mostly clean, the results of models would significantly improve if other columns were created by running computations on different columns. For example, URL paths and body values should be checked for common malicious words, and the result should be placed in a new column. It caught my eye as I could only find one documented experiment using this dataset, and it was a minimal experiment, leaving the possibilities almost untouched.