Predicting Capture-to-Control Delay in an Automated UAV System

Rahul Thapa\textsuperscript{1}, Davide Callegaro\textsuperscript{2}, and Marco Levorato\textsuperscript{2}
\textsuperscript{1} Department of Computing Sciences, Villanova University
\textsuperscript{2} Donald Bren School of Information and Computer Science, UC Irvine

With the advent of edge computing, the hardware limitation imposed upon Unmanned Aerial Vehicles (UAV) has been mitigated. Robust frameworks such as HyDRA further improve the performance of the system pipeline despite using interconnected resources rather than onboard resources. However, the lag between sensing and control in this pipeline, referred to as capture-to-control delay, still possesses a problem. The performance of this system can be drastically improved if the factors causing significant increases in capture-to-control delay can be identified. This project deals with understanding the capture-to-control delay in the pipeline and selecting the highly correlated features with the delay so that it can be predicted and mitigated in future.

**Edge Computing**
- A powerful computing device is connected to a system via a network such as WiFi or LTE (the edge server)
- Handles heavy computing (edge computing)
- Benefits: Reduces load from Drone and energy consumption
- Problems: Unpredictable Capture-to-Control Delay

**HyDRA**
- A middleware architecture for resilient computing in Heterogenous Autonomous Systems \cite{1}
- Enables the adaptive distribution of computing tasks within a network of devices
- The modular architecture allows flexibility in deploying sensing-to-control pipelines, and also to assist dynamic activation

**Preliminary Data Analysis**
- Delay above 0.11s considered an anomaly
- Dataset includes telemetry data such as acceleration, inclination, and network data such as TCP packets sent and received

**Feature Selection**
Feature selection helps machine learning algorithms run faster and produce more accurate results. We utilized the concept of mutual information to select the best features from our feature space:

\[
P(\mathbf{x}, \mathbf{y}) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \left( \frac{p(x, y)}{p(x)p(y)} \right)
\]

\(p(x, y)\) and \(p(x)\) and \(p(y)\) are the marginal probability mass functions.

We used various approaches such as mutual information max relevance min redundancy (mRMR) based on \cite{2} as well as scikit-learn based algorithms to calculate mutual information.

**Modeling**
- Sequence to sequence prediction, where we use the features selected in the previous step to predict future delays.
- Used Recurrent Neural Networks with Long Short Term Memory layers
- Used the hidden state to propagate the contextual information through the experiment

**Future Works**
- Training the model on more data collected from various drone flights
- Optimizing the model by tuning hyperparameters

**References**
\begin{itemize}
\item \textsuperscript{1} Davide Callegaro, Sabur Baidya, and Marco Levorato. A measurement study on edge computing for autonomous UAVs. Proceedings of MageSys Workshop (SIGCOMM) 2019.
\end{itemize}

**Acknowledgments**
The author would like to thank NSF for providing research grant for the summer, IOT-SITY program for hosting the program, and Intelligent and Autonomous Systems Lab, lead by Prof. Levorato for allowing me in the lab.