Predicting Capture-to-Control Delay in Automated UAV Systems

Rahul Thapa[1], Marco Levorato[2], and Benjamin Mitchell[1]

[1] *Department of Computing Sciences, Villanova University*

[2] *Donald Bren School of Information and Computer Science, UC* Irvine

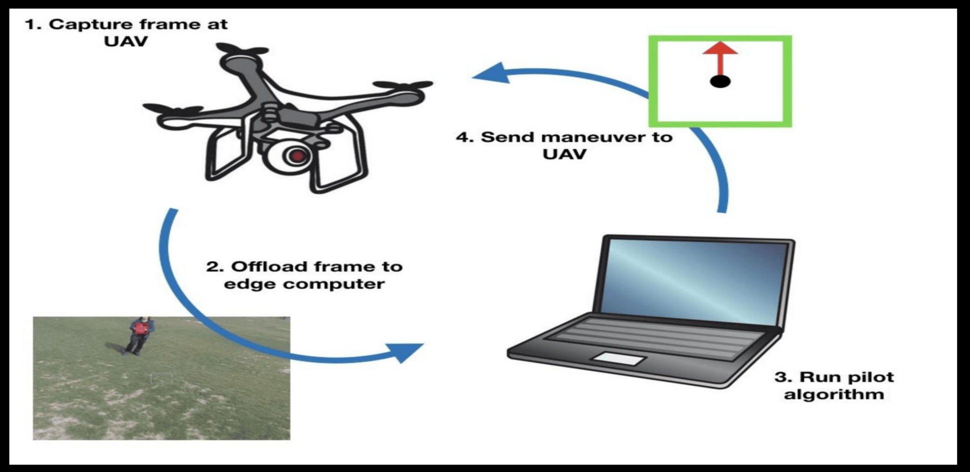
With the advent of edge computing, the hardware limitation imposed upon Unmanned Aerial Vehicles (UAV) has been mitigated. Robust frameworks such as HyDRA further improve the performance of the system pipeline despite using interconnected resources rather than onboard resources. However, the lag between 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

Adapted from Callegaro et al. (2019)

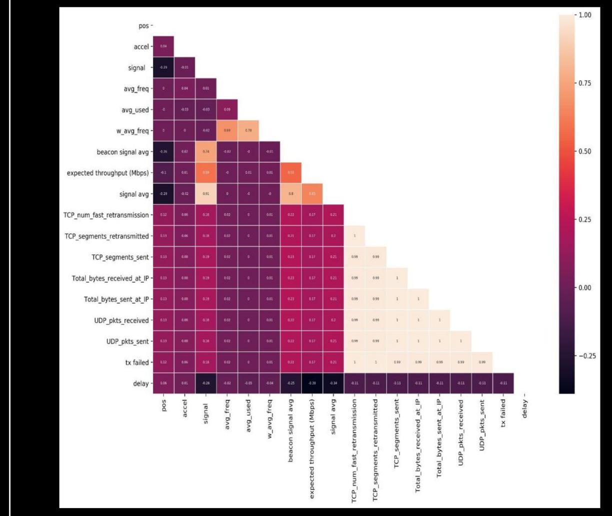
HyDRA

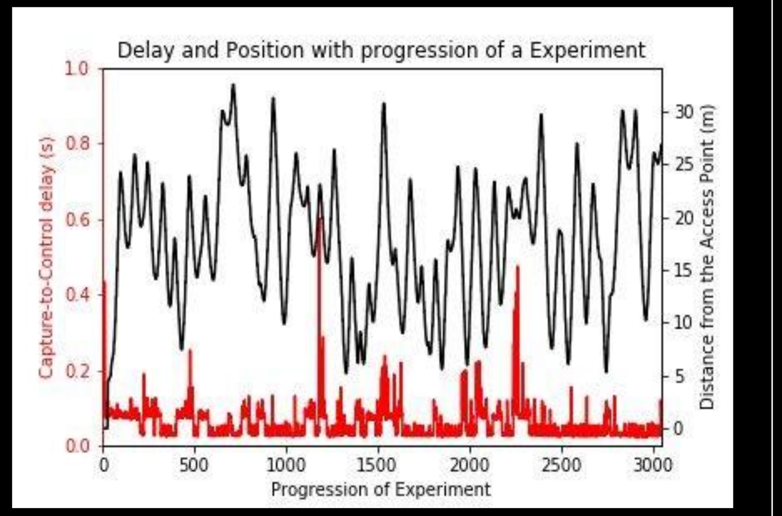


- A middle-ware architecture for resilient computing in Heterogenous Autonomous Systems [1]
- Enables the adaptive distribution of computing tasks within a network of devices
- The modular architecture allows flexibility in deploying sensing-to-control pipelines, and also to assist dynamic activation of pipelines

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:

$$\mathrm{I}(X;Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p_{(X,Y)}(x,y) \log \left(\frac{p_{(X,Y)}(x,y)}{p_X(x)\, p_Y(y)} \right)$$

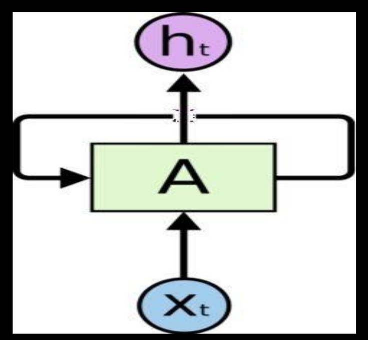
P(x, y) is the joint probability mass function and Px and Py are the marginal probability mass functions.

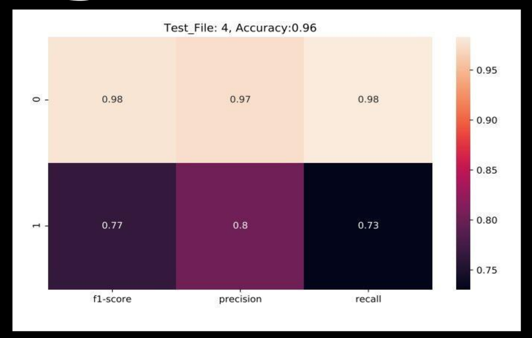
We used various approaches such as minimal redundancy maximal relevance (mRMR) based on [2] as well as scikit-learn based algorithms to calculate mutual information.

Using these approaches we selected a shortlist of features that show good performances in predictability. They can be described as:

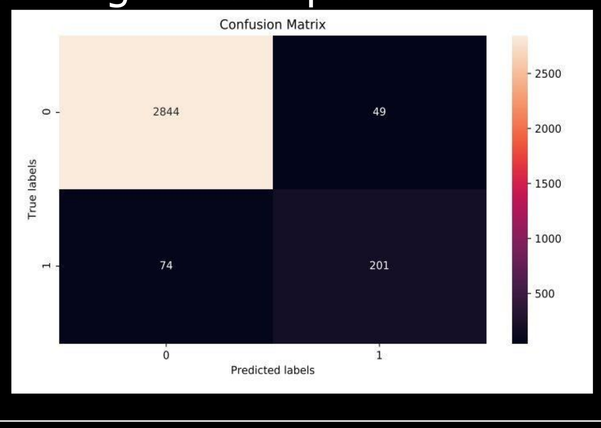
- TCP segments received
- Failed retransmissions
- IP packets received

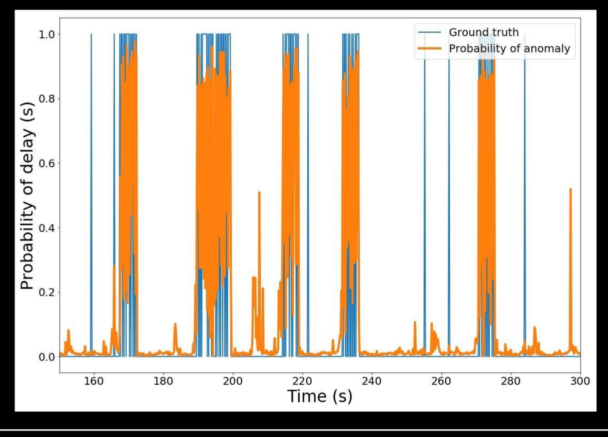
Modeling





- **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 Work

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

References

[1] Davide Callegaro et al., **Information Autonomy: Self-Adaptive Information Management for Edge-Assisted Autonomous UAV Systems**. Proceedings of IEEE Military Communications Conference (MILCOM), 2019.

[2] Hanchuan Peng, Fuhui Long, and Chris Ding. **Feature selection based on mutual information**. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2005.

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 hosting and mentoring.