SYSTEMS ENGINEERING RESEARCH CENTER

Reliability Engineering of Autonomous Systems incorporating Machine Learning

Aiden Gula, Christian Ellis, Saikath Bhattacharya, and Lance Fiondella

University of Massachusetts, MA, USA

Outline

- Motivation
- Relationship of machine learning to system and software reliability engineering
- Reliability growth modeling and reliability engineering of machine learning
- Software and system testing considering machine learning
- Conclusions
- Future Research

Motivation

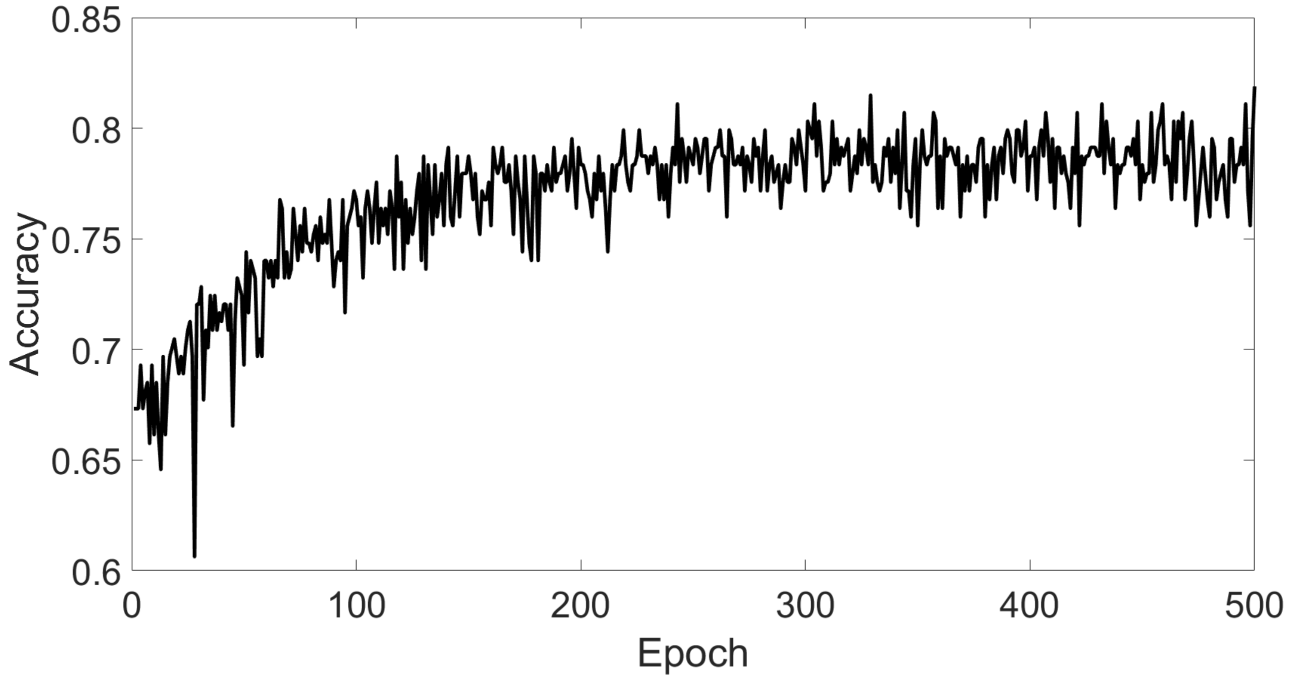
- Artificial intelligence (AI) and machine learning (ML) recognized as enablers of autonomy
  - Susceptible to a variety of failures and adversarial attacks
  - Pressing need to understand how ML capabilities can be incorporated into existing system engineering processes
- Provide Test & Evaluation community with familiar framework in which to assess autonomous systems
- Facilitate effective communication among stakeholders
  - System engineers, advanced algorithm designers, testers, leadership

Relationship of machine learning to reliability engineering

- Machine learning
  - Typically resides in software
    - Software reliability problem
  - Often characterized by perceive, decide, execute loop
  - Resides in software architecture with traditional software components
    - Necessitates test of
      - Traditional and ML components as well as interactions
    - Increases complexity and need for realism in
      - Hardware/software reliability, architecture-based software reliability, and software reliability growth models

Reliability growth modeling

- Accuracy - (Correct predictions)/(predictions attempted)



Appropriate for classification algorithms that may indirectly inform autonomy

Probably Approximately Correct (PAC) learning framework

- Rich theoretical framework overshadowed by recent empirical success
- Defines class of learnable concepts in terms of sample size
- Problem is PAC-learnable if there is an algorithm with $\varepsilon > 0, \delta > 0$

$$\Pr\{R(m) > 1 - \varepsilon\} \geq 1 - \delta$$

  - $R(\cdot)$ - Reliability of fitted model
  - m – sample size (polynomial in $1/\varepsilon$, $1/\delta$, cost of representing inputs, and size of concept to be learned)
  - $(1 - \delta)$ - Confidence
- Efficiently PAC-learnable also runs in polynomial time

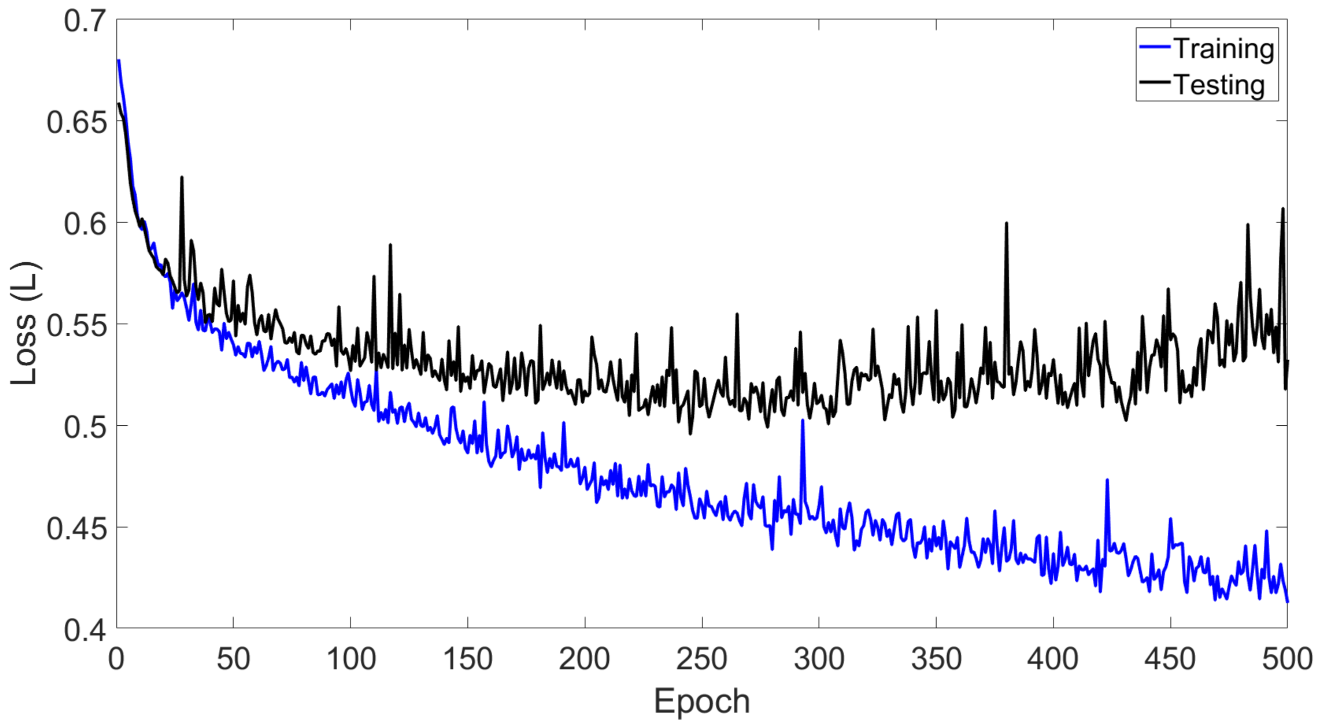
Can inform feasibility of attaining desired accuracy, including cost of data

Reliability Engineering

- Like other statistical models, machine learning prone to overfitting

- Model selection
  - Attempts to reduce error, decomposed into estimation and approximation error
  - Estimation error
    - Function of model fit
  - Approximation error
    - Cannot be estimated
    - Describes how well model fit approximates Bayes error or average noise
  - Empirical risk minimization
    - Seeks to minimize error on training sample
  - Tradeoff between estimation and approximation error required
    - Related to classical dilemma of model complexity vs. predictive goodness of fit

Impact of model fitting on loss in training and testing data



Minimizing loss on training data overfits model

Regularization

- Method to avoid overfitting

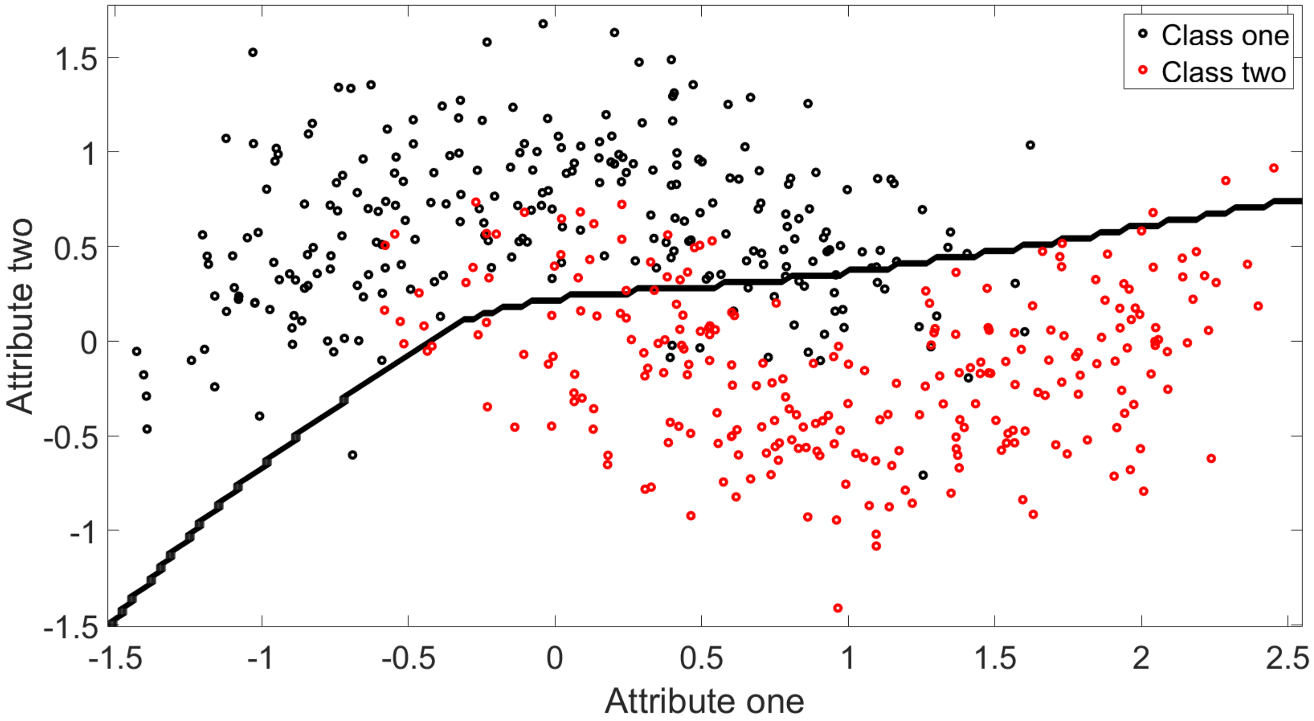
$$R(h) = L(h) + \lambda * C(h)$$

  - h - Hypothesis (fitted model)
  - L - Empirical loss
  - $\lambda > 0$ - Penalty applied to complexity function
  - $C(h)$ – Complexity of hypothesis h

Cross-validation, fault tolerance, and bootstrapping

- K-fold cross-validation
  - —Used when data too small to reserve subset for validation
  - —Uses data for both training and testing
  - —Divides dataset of size m into n subsets of equal size
  - —Learning algorithm trained on $(n-1)$ subsets and validated with remaining subset
  - —Applied iteratively to improve a model's predictive accuracy
  - —Employed in conjunction with regularization
- Fault-tolerance (ensemble learning)
  - —Includes both unweighted and variety of weighted majority voting techniques
  - —Bootstrapping popular technique to avoid overfitting in context of ensemble classifiers

Input-domain view of testing machine learning algorithm



Related to concept of coverage from traditional software testing as well as model complexity in machine learning

Failure modes, effects and criticality analysis (FMECA)

- How system or subsystem fails, consequences, and severity
- Coupled with fault tree analysis to characterize logical structure of failure propagation, quantify risk, and prioritize mitigation
- Autonomous vehicle example

		Actual Values	
		Pedestrian	No Pedestrian
Predicted Values	Pedestrian	TP	FP (Minor)
	No Pedestrian	FN (Catastrophic)	TN

Consequences of misclassification can vary

Cost sensitive learning

- Trains classifier in light of cost

$$F(C) = \sum_{j=0}^{n} \sum_{i=0}^{n} c_{ij} p_{ij}$$

  - n – Number of classes
    - Class 0 corresponds to 'nothing'
  - c_{ij} - Cost of classifying object of class i in class j
  - p_{ij} - Probability of classifying object of class i in class j
- Data-based method (Class rebalancing)
  - Under samples more prevalent data and oversamples underrepresented data
- Algorithmic methods
  - Modify learning process to improve sensitivity to catastrophic misclassifications

Conclusion and Future Work

- Identified relationships between
  - Machine learning and system and software reliability engineering
- Mapped machine learning methods to traditional reliability concepts
  - Reliability growth modeling
  - Reliability engineering
  - Fault tolerance
  - Software testing
  - Failure modes, and effects criticality analysis
- Intended to assist individuals familiar with reliability engineering communicate with machine learning experts to support engineering of autonomous systems incorporating machine learning

Future Research

- Further elaborate connections between reliability engineering and machine learning methods
- Explore
  - Relationship between adversarial machine learning and failure modes, effects and criticality analysis
  - Application of techniques from machine learning to support reliability engineering

Acknowledgement and Disclaimer

- This material is based upon work supported by the US Army Research Laboratory (ARL) under a Joint Faculty Appointment

- Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ARL