How to read a scientific paper (instructor)

How to Read an Engineering Research Paper (instructor)

DIRE: a neural approach to decompiled identifier naming (sample presented by TA)

Attention is All You Need (presented by guest speaker) [also see
1. Recurrent Neural Networks : Slides from Stanford CS224n.
2. LSTMs and Bidirectional RNNs : Slides from Stanford CS224n.
3. Encoder-Decoders and Attention : Slides from Stanford CS224n.]

Metamorphic Testing: Testing the Untestable (sample presented by instructor)

The following papers were all presented by students. Every student presented at least twice; some presented two papers from this list, some presented one of these papers plus their own own midterm paper or final project instead of a second research paper.

Understanding security mistakes developers make: Qualitative analysis from Build It, Break It, Fix It

DRIFT: Deep Reinforcement Learning for Functional Software Testing

Automatic Generation of Acceptance Test Cases from Use Case Specifications: an NLP-based Approach

code2vec: Learning Distributed Representations of Code

Neural code comprehension: a learnable representation of code semantics

code2seq: Generating Sequences from Structured Representations of Code

Where is the bug and how is it fixed? an experiment with practitioners

DeFlaker: Automatically Detecting Flaky Tests

Order Matters: Semantic-Aware Neural Networks for Binary Code Similarity Detection

Oreo: detection of clones in the twilight zone

FairFuzz: a targeted mutation strategy for increasing greybox fuzz testing coverage

CodeBERT: A Pre-Trained Model for Programming and Natural Languages

DeepBugs: a learning approach to name-based bug detection

SapFix: Automated End-to-End Repair at Scale
Asm2Vec: Boosting Static Representation Robustness for Binary Clone Search against Code Obfuscation and Compiler Optimization

Fairness testing: testing software for discrimination

Learning Semantic Program Embeddings with Graph Interval Neural Network

Shaping Program Repair Space with Existing Patches and Similar Code

Do Transformer Modifications Transfer Across Implementations and Applications?

Context-Aware Patch Generation for Better Automated Program Repair

Using recurrent neural networks for decompilation

InFix: Automatically Repairing Novice Program Inputs

Debin: Predicting Debug Information in Stripped Binaries

DeepTLE: Learning Code-Level Features to Predict Code Performance before It Runs

Running symbolic execution forever

PhASAR: An Inter-procedural Static Analysis Framework for C/C++

Local Differential Privacy for Deep Learning