TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

DNA sequence Classification

Sourajyoti Datta
Supervisor: **Muhammad Asim**
Project – Collaborative Intelligence (DFKI)
Winter Semester 2020/21
Department of Computer Science
Technische Universität Kaiserslautern, Germany

Introduction

- **Why DNA Sequence Classification?**

  - The genome of eukaryote species is embedded into the nuclei of their cells, packed as chromatin.
    - ***Nucleosomes*** form the first level of DNA compaction
      - About 147-150 base-pairs
      - Arrange themselves through successively higher-order structures to finally form the chromosomes (critical role in organization)
    - The genome-wide location of the nucleosomes is fundamental for many biological processes
      - Gene regulation, Co-transcriptional splicing, DNA replication, DNA repair
    - Nucleosomes are separated from each other by sequences of ***Linker*** DNA

  - **Need to understand:**
    - To what extent the DNA sequence specificity is solely responsible for nucleosome positioning.

  - **Task:**
    - Use only short DNA sequences (length of 150) to try to model a classification between the sequences being Nucleosomal vs. Linker

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Datasets

- **4 different datasets explored, finalized and results presented for one:**
  - Setting 1 Data:
    - Contains data from 4 different sets of DNA Sequences:
      - **Homo Sapiens**:
        - Nucleosomal: *2900* Linker: *2850*
      - **Caenorhabditis Elegans**:
        - Nucleosomal: *2567* Linker: *2608*
      - **Drosophila Melanogaster**:
        - Nucleosomal: *2273* Linker: *2300*
      - **Saccharomyces Cerevisiae**:
        - Nucleosomal: *1880* Linker: *1740*
    - Every DNA sequence is **150** characters long, containing only: Adenine, Cytosine, Guanine, and Thymine (ACGT) molecules
    - Fair division of classes in every species. Hence, no data bootstrapping/augmentation step performed.

  - For every model:
    - Data is divided into Stratified 10 Folds
    - Ensures better accuracy/precision of the results and their generalization

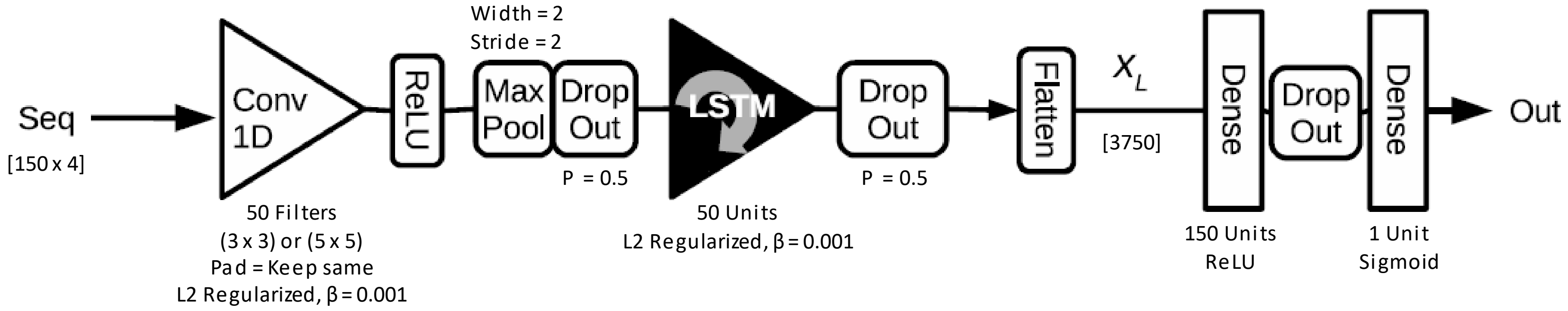
Approaches explored

- **CNN – RNN – FCNN hybrid architectures**
  - Convolutional LSTM DLNN
  - Modified version: Convolutional LSTM DLNN CORENup
- **Embeddings with ML/NN**
  - Global DNA Embedding with Random Forest
  - Word2Vec with FCNN
- **ML/NN with Statistical Metrics**
  - MathFeature
    - Random Forest
    - Extra Tree Forest
    - XGBoost
    - Convolutional LSTM DLNN
    - Deep Forest Architecture
- **AutoEncoders based**
  - Cosine Similarity based AE with Deep Forest architecture
  - Bi-LSTM AE with FCNN

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Approaches and Results

- **CNN – RNN – FCNN hybrid architectures**
  - Convolutional LSTM DLNN

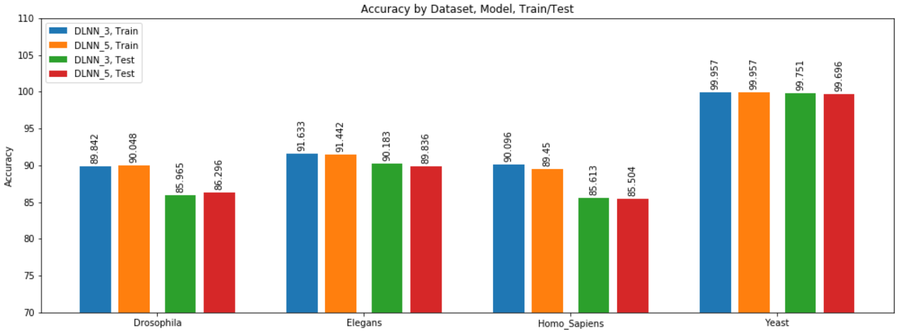


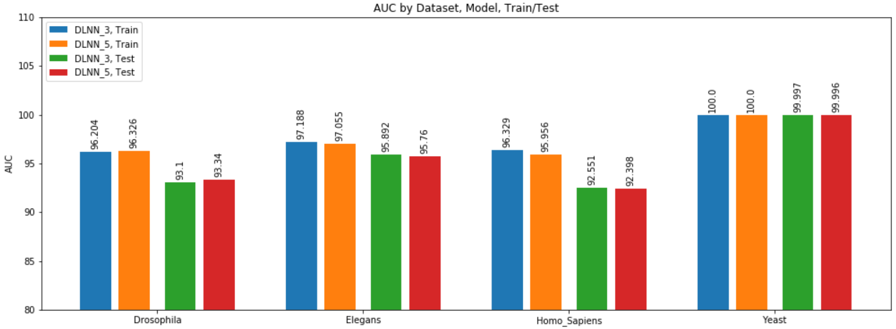
- Total parameters: **584,051**
- Trained using:
  - **Adam optimizer** with **Learning rate = 0.0003**
  - For, **200 Epochs** with **Batch size of 64**
  - **Early Stopping** criterion on validation-loss
  - **Binary Cross-Entropy** loss

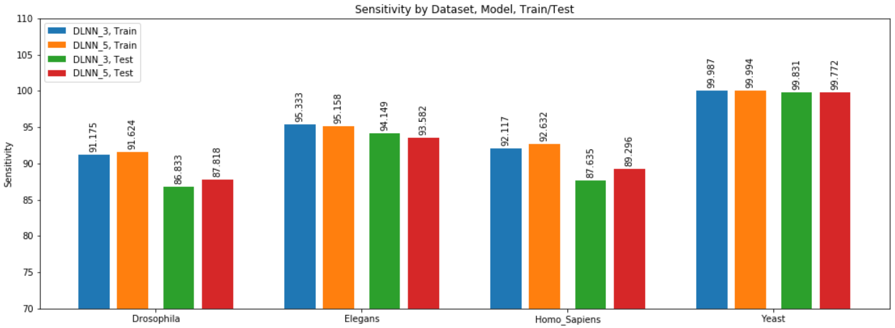
** The Hyperparameters: Epoch, Batch size and Early stopping criterion, is kept invariant across all NN models in the future

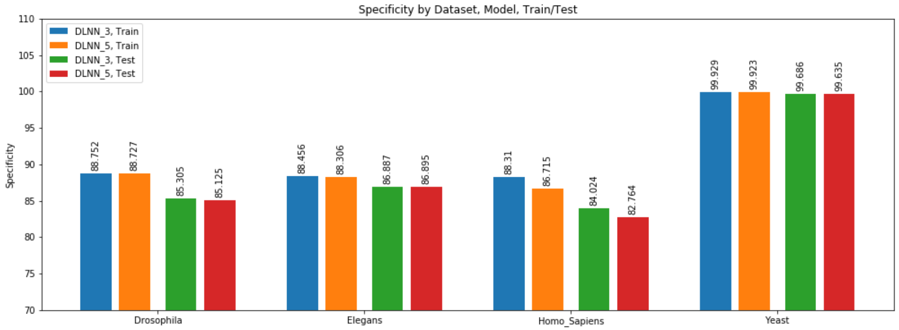
Approaches and Results

- **CNN – RNN – FCNN hybrid architectures**
  - Convolutional LSTM DLNN


Accuracy by Dataset, Model, Train/Test

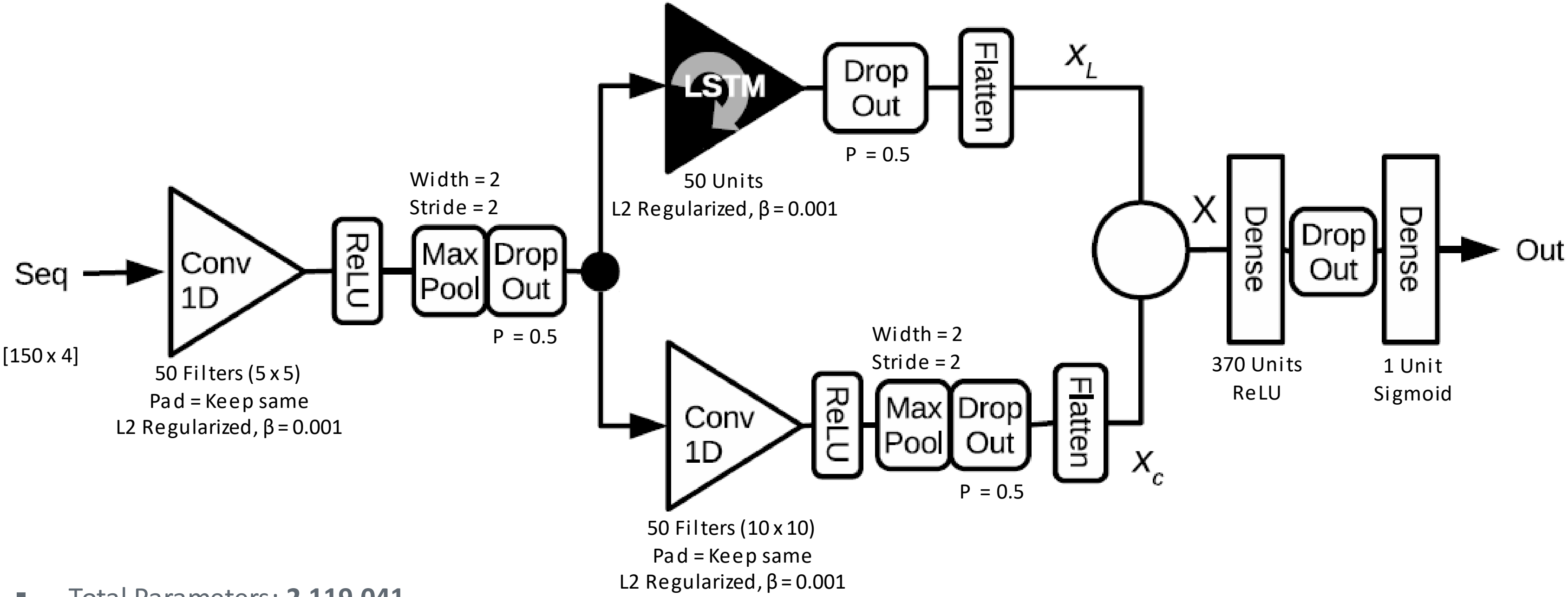

AUC by Dataset, Model, Train/Test


Sensitivity by Dataset, Model, Train/Test


Specificity by Dataset, Model, Train/Test

Approaches and Results

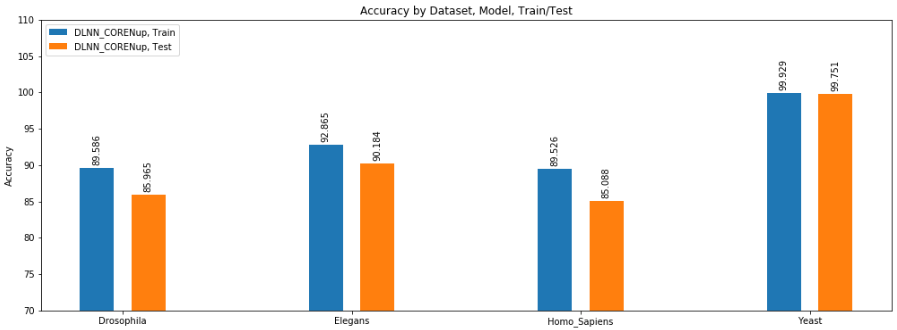
- **CNN – RNN – FCNN hybrid architectures**
  - Modified Convolutional LSTM DLNN – CORENup model **(State-of-the-art Architecture)**

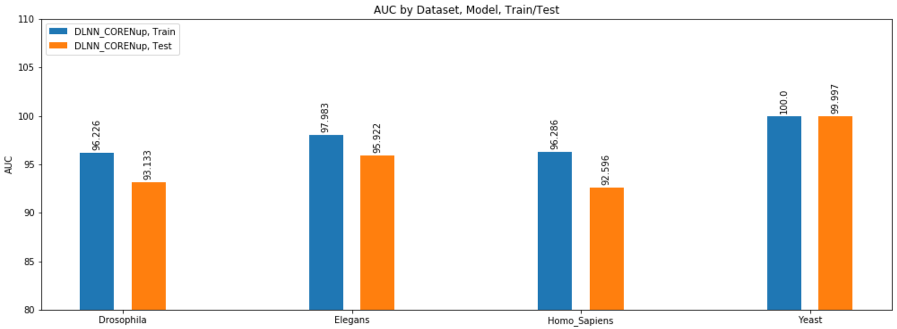


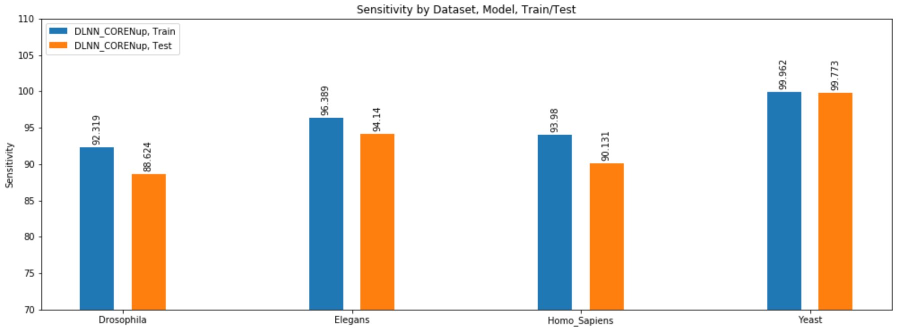
  - Total Parameters: **2,119,041**
  - Similar training Hyper-Parameters as the parent architecture
  - Basis for all future model comparisons

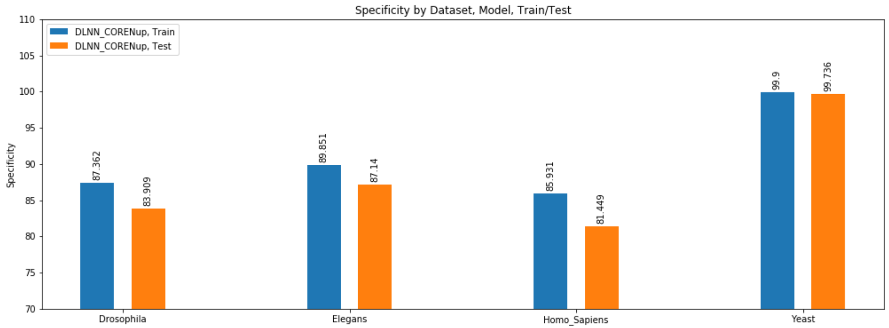
Approaches and Results

- **CNN – RNN – FCNN hybrid architectures**
  - Modified Convolutional LSTM DLNN – CORENup model **(State-of-the-art Architecture)**


Accuracy by Dataset, Model, Train/Test


AUC by Dataset, Model, Train/Test


Sensitivity by Dataset, Model, Train/Test

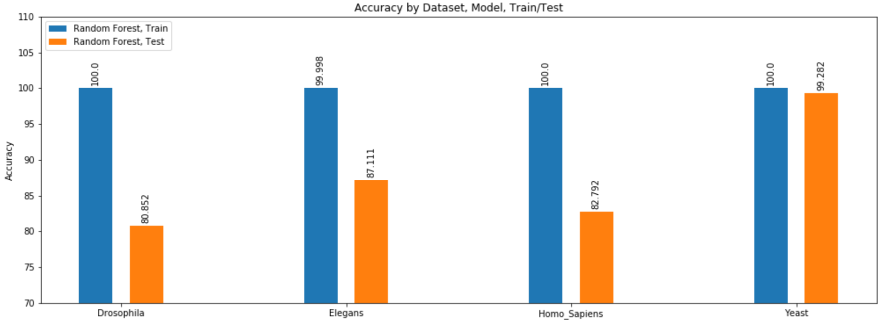

Specificity by Dataset, Model, Train/Test

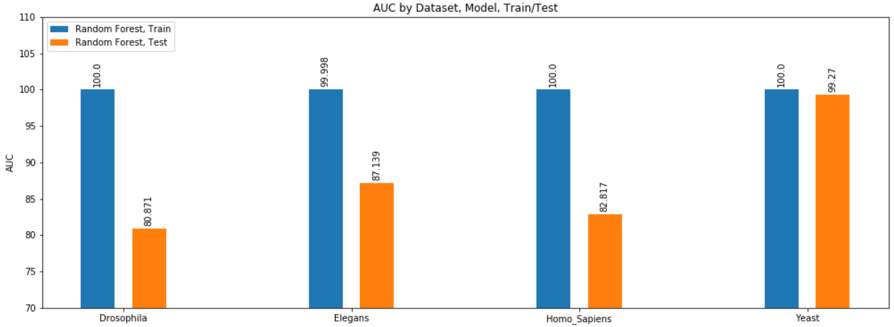
Approaches and Results

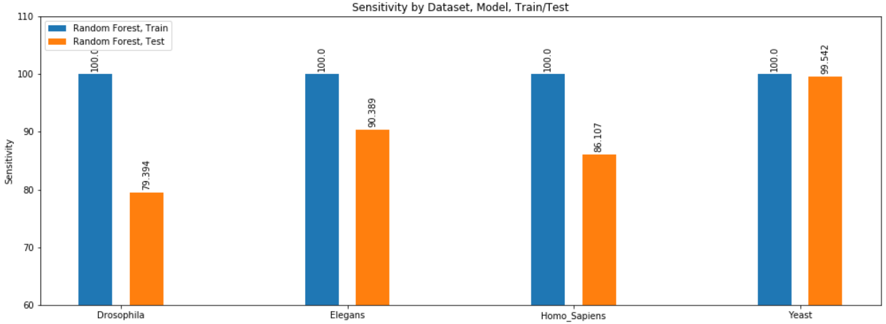
- **Embeddings**
  - **Global DNA Embedding with Random Forest**

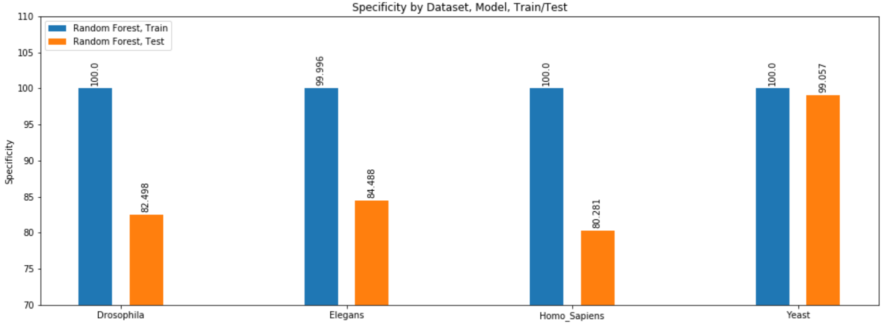
Approaches and Results

- **Embeddings**
  - Global DNA Embedding with Random Forest


Accuracy by Dataset, Model, Train/Test

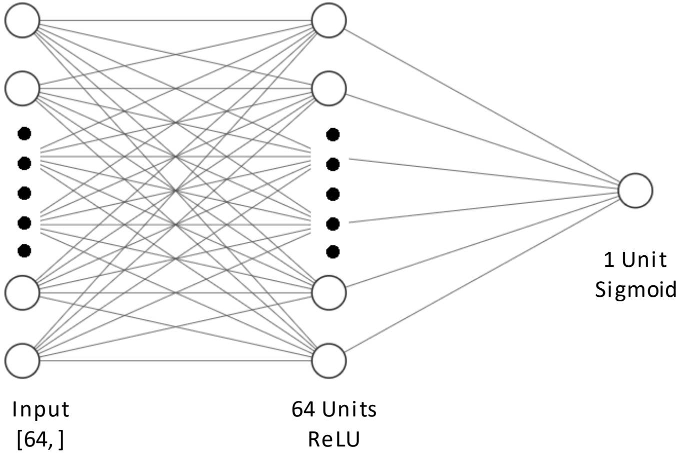

AUC by Dataset, Model, Train/Test


Sensitivity by Dataset, Model, Train/Test


Specificity by Dataset, Model, Train/Test

Approaches and Results

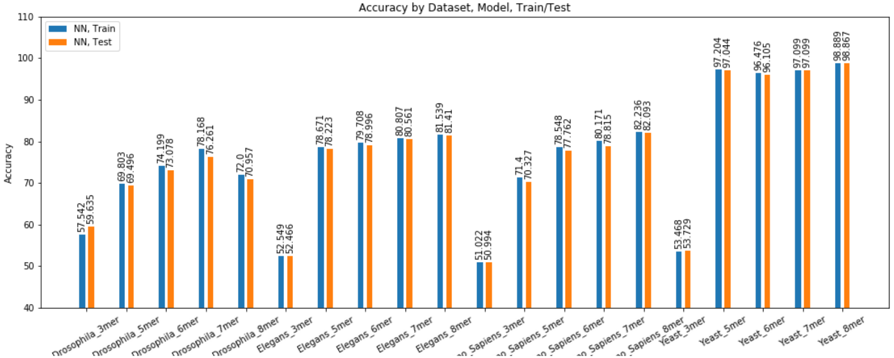
- **Embeddings**
  - **Word2Vec with FCNN**
    - Uses **Skip-Gram word2vec** to embed **k-mers**
      - Then, leverages an existing sentence embedding technique to embed all sequences of specific samples
      - Finally, SVD is performed to limit the dimensions
      - Preserves relevant information about sequencing, such as k-mer context, sequence taxonomy, and sample class
      - Embedded using k-mer values: {3, 4, 5, 6, 7, 8}
    - Finally, classification is performed using a shallow FCNN

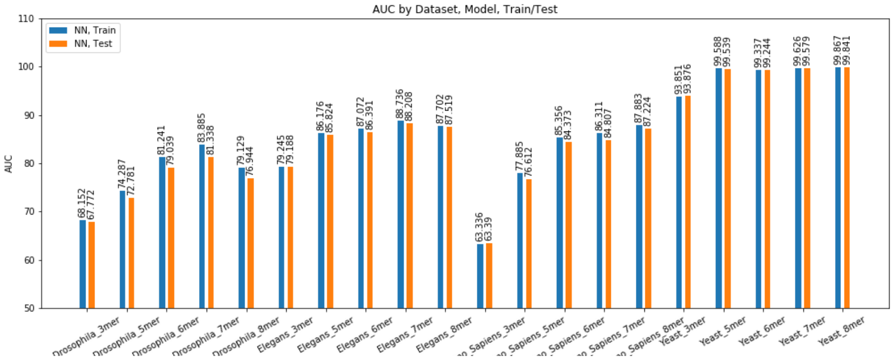


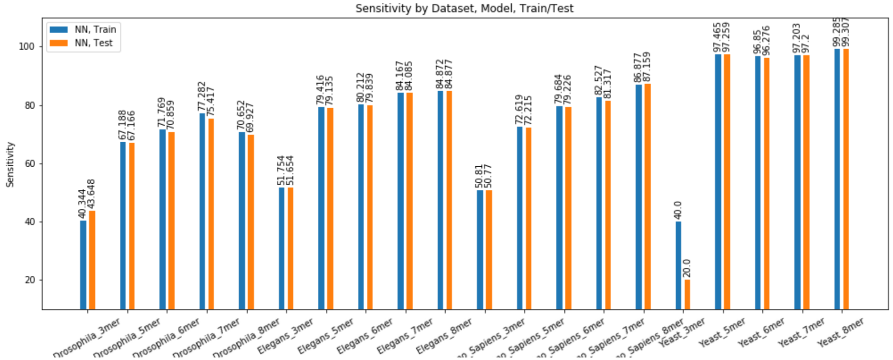
- Trained using:
  - **Adam optimizer** with **Learning rate = 0.0003**
  - For, **200 Epochs** with **Batch size of 64**
  - **Early Stopping** criterion on validation-loss
  - **Binary Cross-Entropy** loss

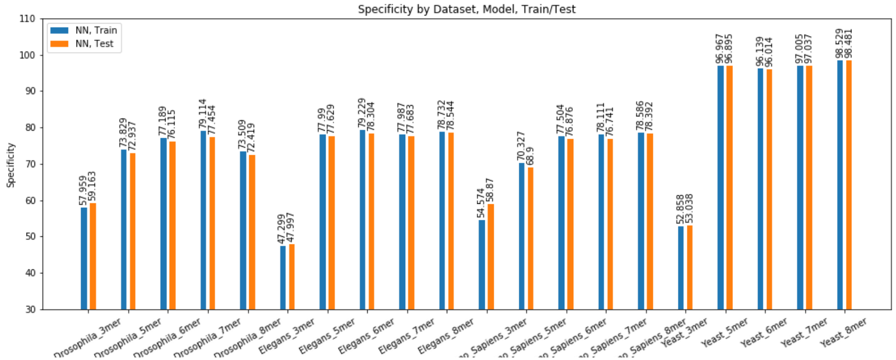
Approaches and Results

- **Embeddings**
  - Word2Vec with FCNN


Accuracy by Dataset, Model, Train/Test


AUC by Dataset, Model, Train/Test


Sensitivity by Dataset, Model, Train/Test

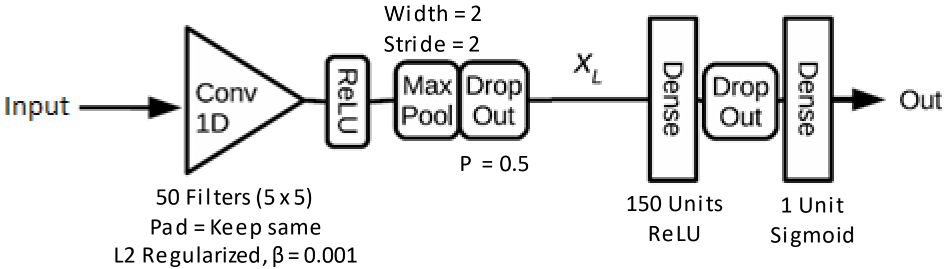

Specificity by Dataset, Model, Train/Test

Approaches and Results

- **Classification with Numerical Measures:**
  - **MathFeature**:
    - Provides **20** approaches based on:
      - Various numeric mappings
      - Genomic signal processing
      - Chaos game theory
      - Entropy
      - Complex networks
      - Fourier Transformation of certain embeddings from the above
    - In total, about **34** different mappings, with a total of about **3217** factors
      - The total can vary depending on parameters for the numerical mappings, like k-mer values, strides, etc.

Approaches and Results

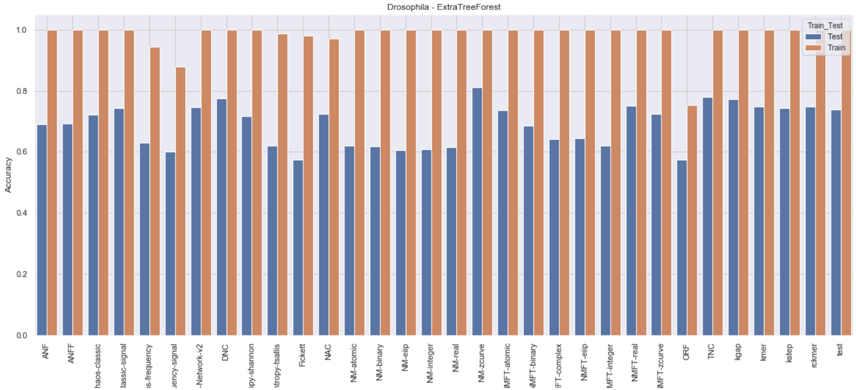
- **Classification with Numerical Measures**
  - **First approach:** Train ML models with each of the *32 MathFeature* mappings individually
    - **Random Forest Classifier**
      - Ensemble of 100 Decision Trees, with **Gini** impurity as criterion
    - **Extra-Trees Classifier**
      - Ensemble of 100 Trees, with **Gini** impurity as criterion
    - **XGBoost**
      - With a Binary Logistic objective
  - **Second Approach:** Train ML model by combining all the *32 MathFeature* mappings into one vector of length 3217
    - **Random Forest**
      - Ensemble of 100 Decision Trees, with **Gini** impurity as criterion
    - **Conv FCNN**



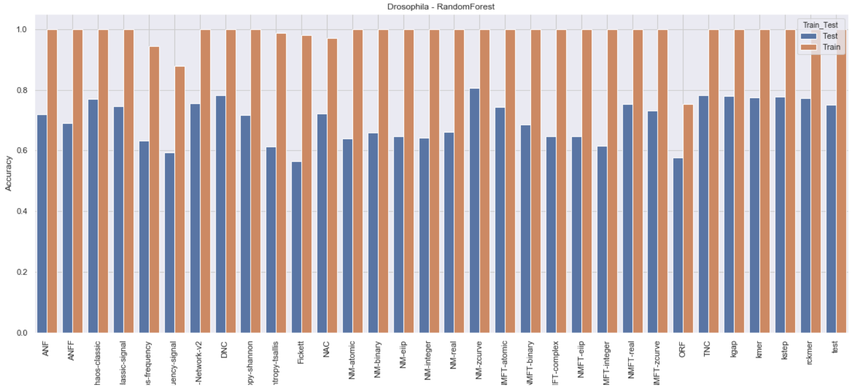
Approaches and Results

- **Classification with Numerical Measures**
  - First approach

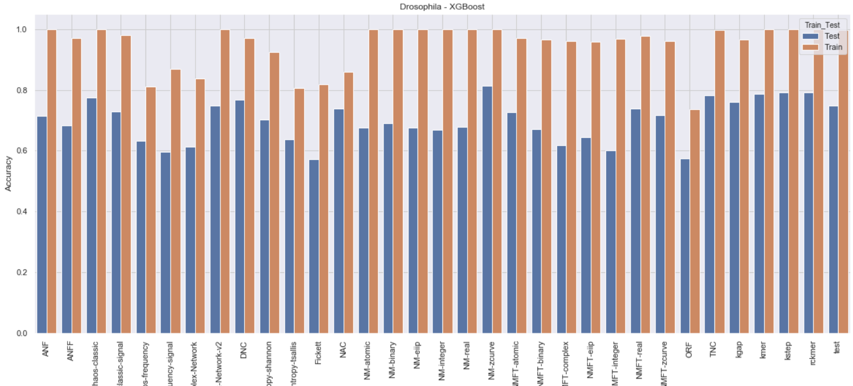
Drosophila - ExtraTreeForest



Drosophila - RandomForest



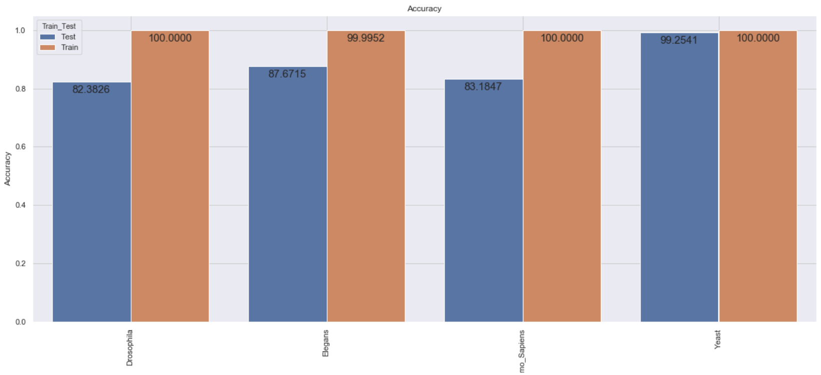
Drosophila - XGBoost

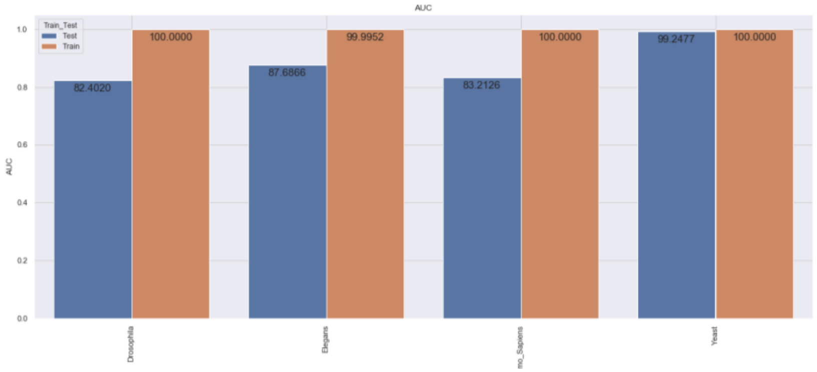


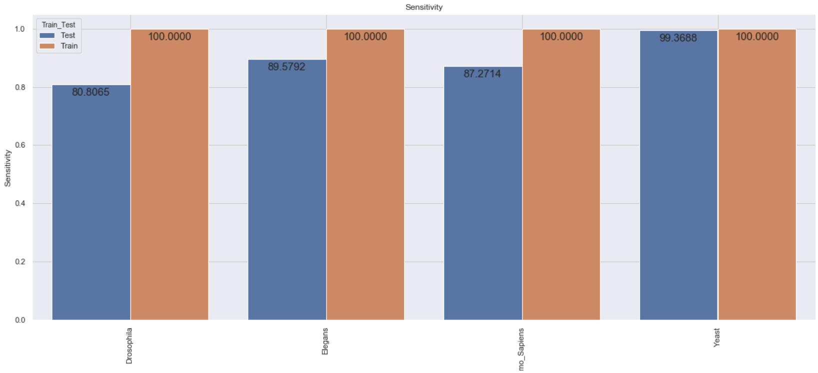
Results here indicates the comparison of predictive power between the different approaches incorporated in the MathFeature

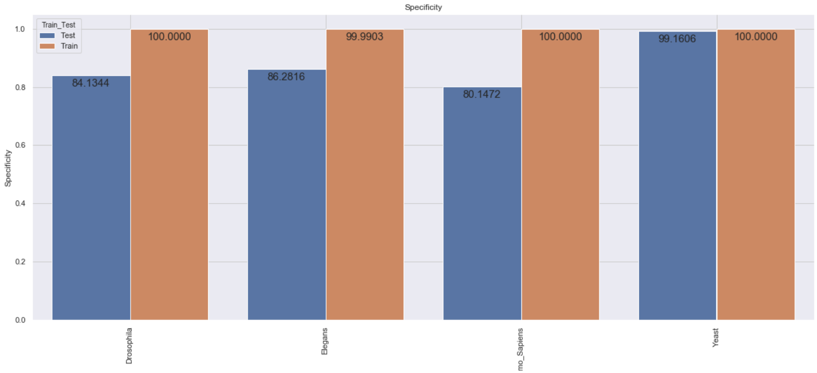
Approaches and Results

- **Classification with Numerical Measures**
  - Second approach: Random Forest


Accuracy

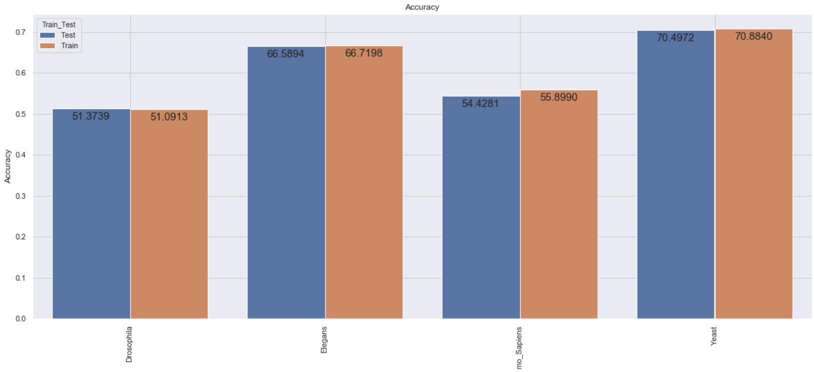

AUC

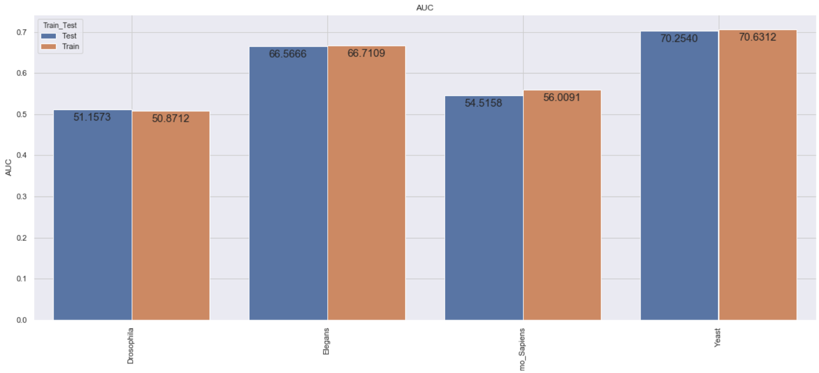

Sensitivity

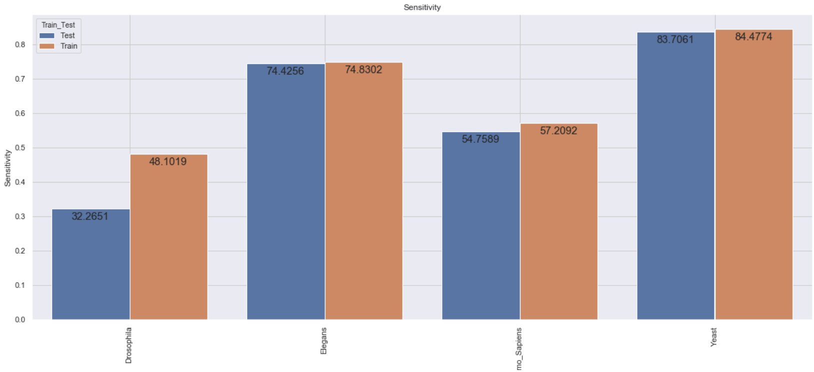

Specificity

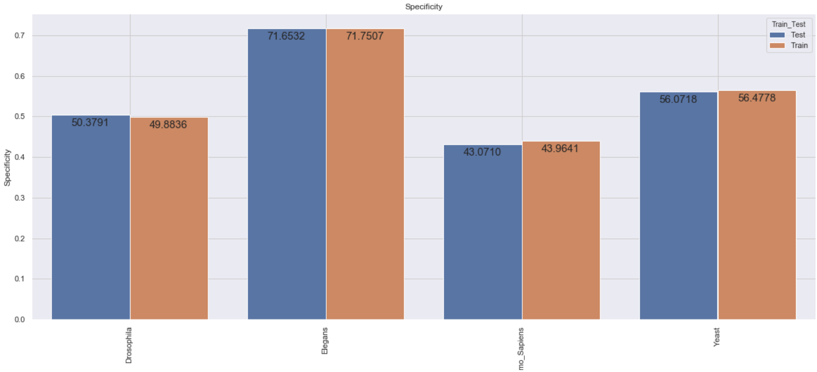
Approaches and Results

- **Classification with Numerical Measures**
  - **Second approach: Conv FCNN**



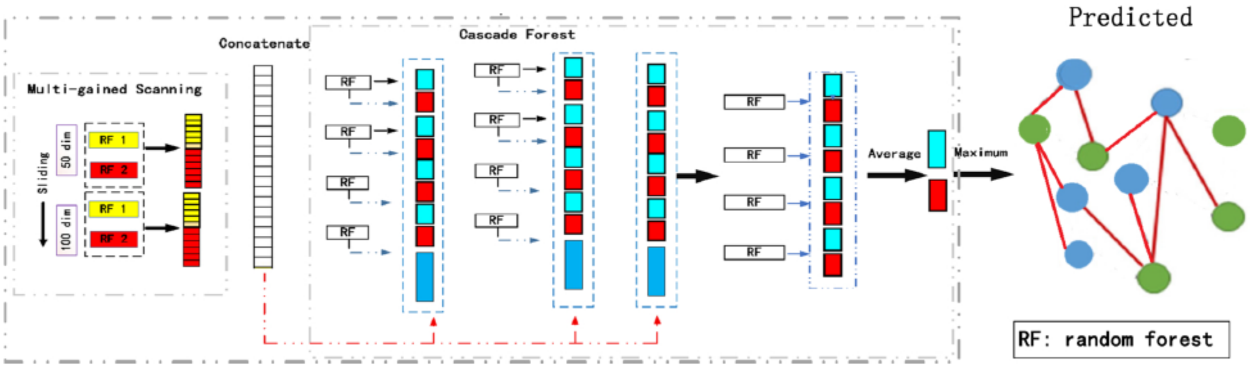






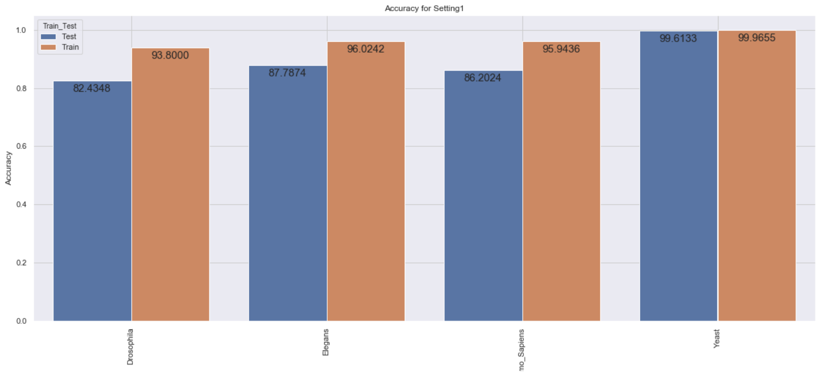
Approaches and Results

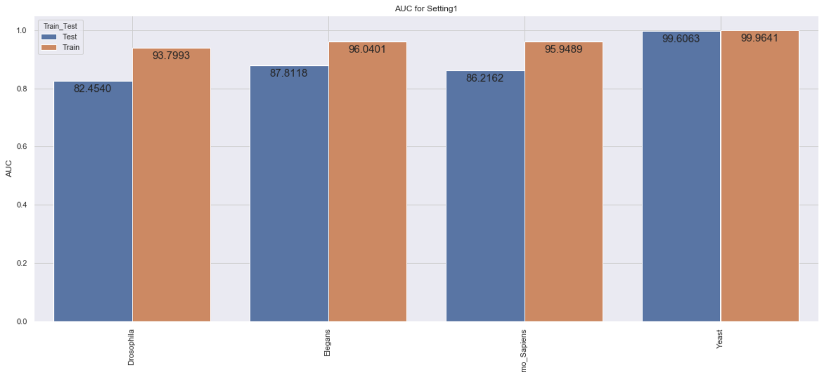
- **Classification with Numerical Measures**
    - **Second Approach:** Train ML model by combining all the *32 MathFeature* mappings into one vector of length 3217
        - **Deep Forest Architecture**
            - Multi-grain scanning and Cascade Forest approach
            - Developed for lncRNA-miRNA interaction prediction, but repurposed to use numerical measures for classification

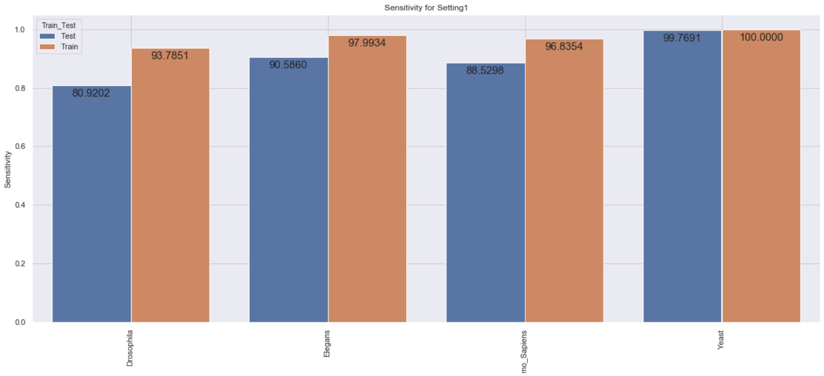


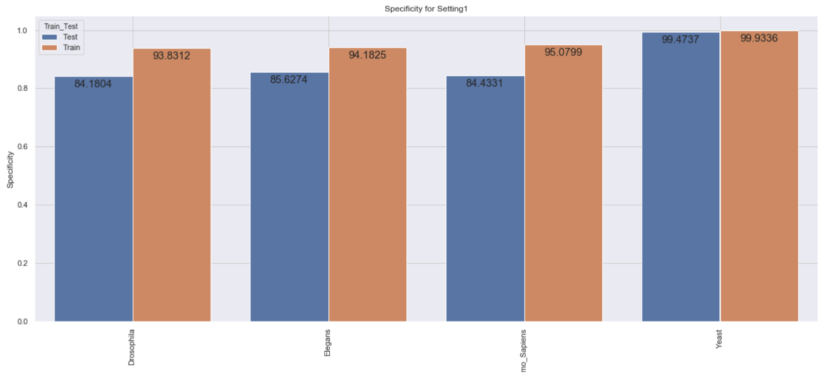
Approaches and Results

- **Classification with Numerical Measures**
  - Second Approach: Deep Forest Architecture



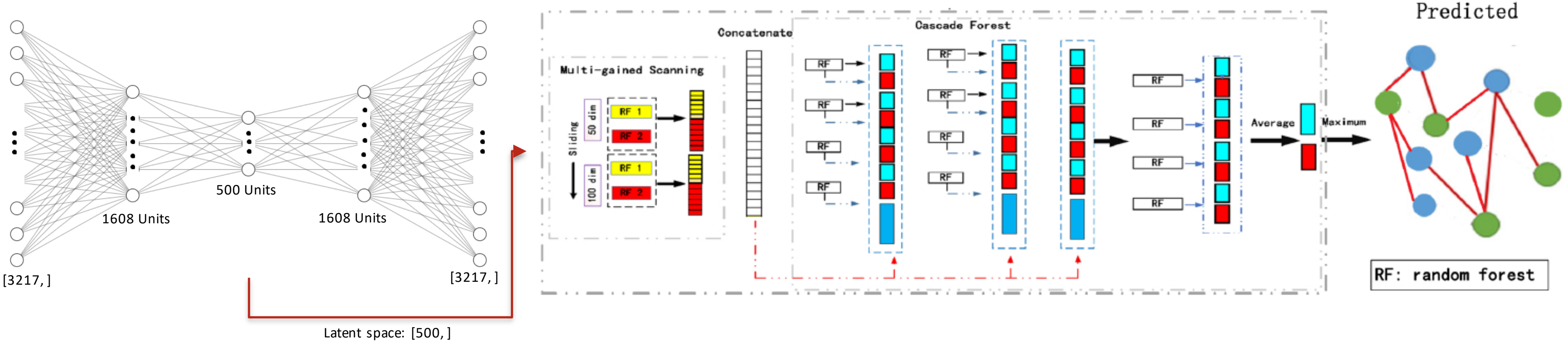






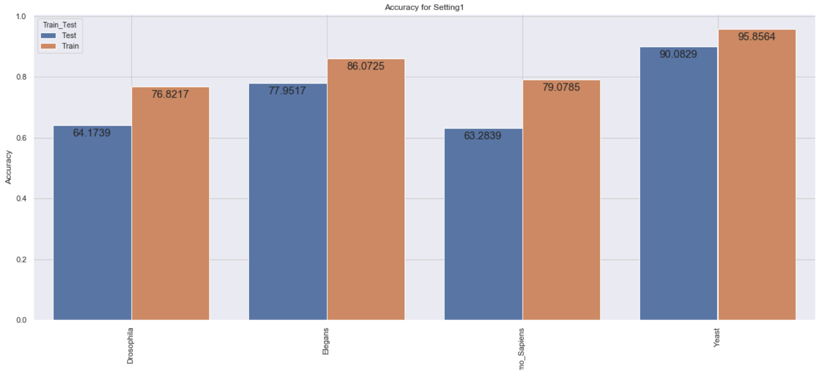
Approaches and Results

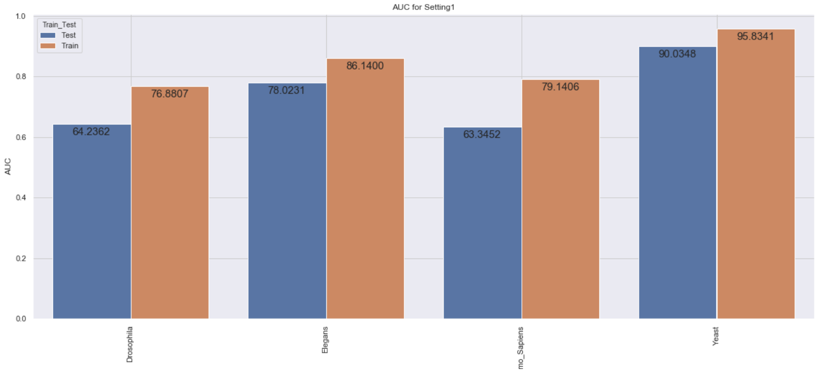
- **AutoEncoders**
  - **Cosine Similarity based AE with Deep Forest architecture**
    - Using the MathFeature based numerical measures
    - Fully connected AutoEncoder without activation, using Cosine Similarity as loss function
    - Using the Second-approach data – combining all 32 MathFeature mappings

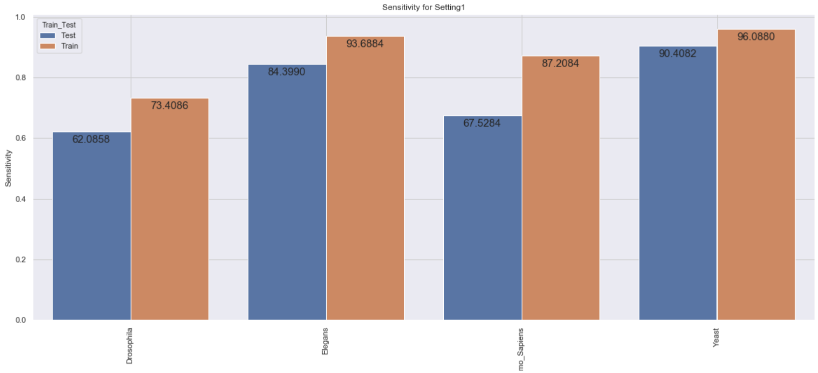


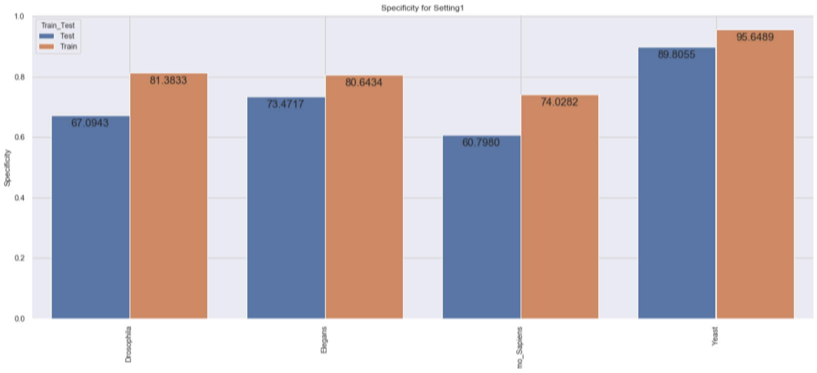
Approaches and Results

- **AutoEncoders**
  - Cosine Similarity based AE with Deep Forest architecture



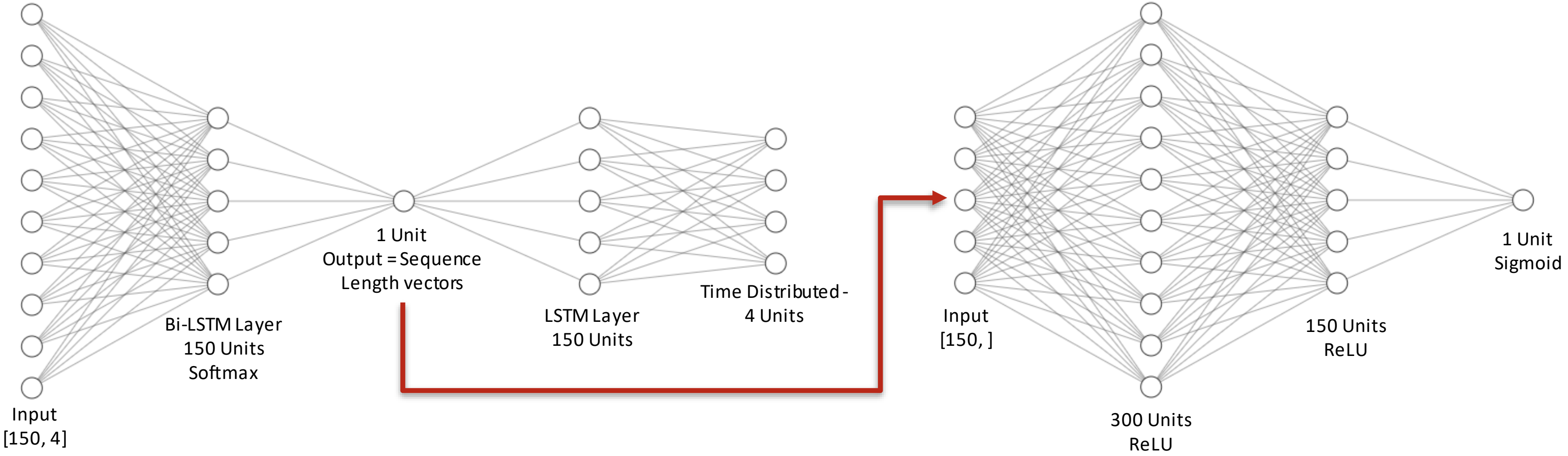






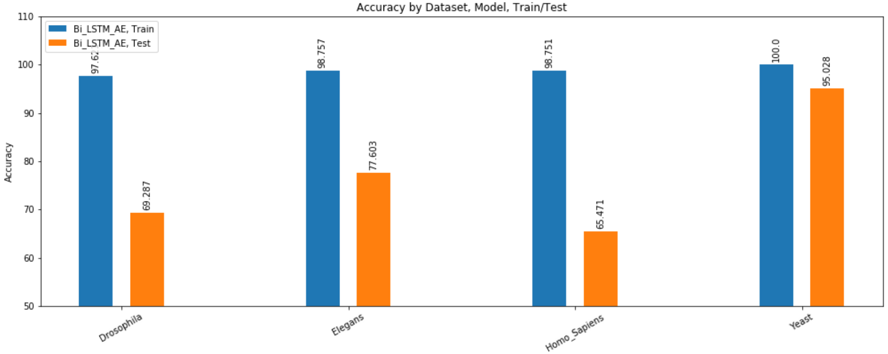
Approaches and Results

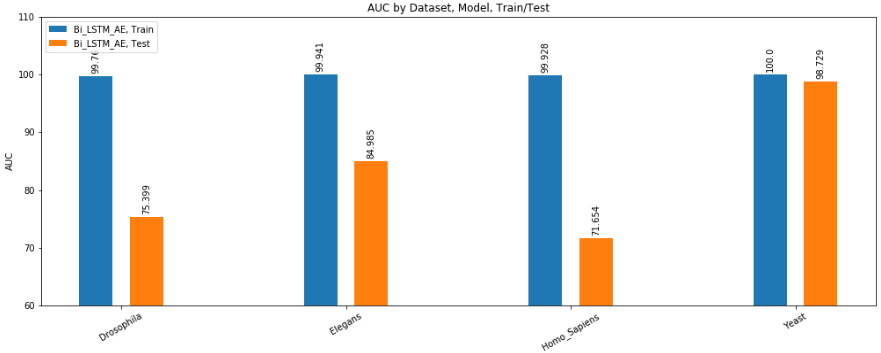
- **AutoEncoders**
  - **Bi-LSTM AE with FCNN**
    - Bi-directional LSTM based AutoEncoder, using Categorical Cross Entropy loss
    - Then, Fully Connected Classification Network using latent representation, using Binary Cross Entropy loss

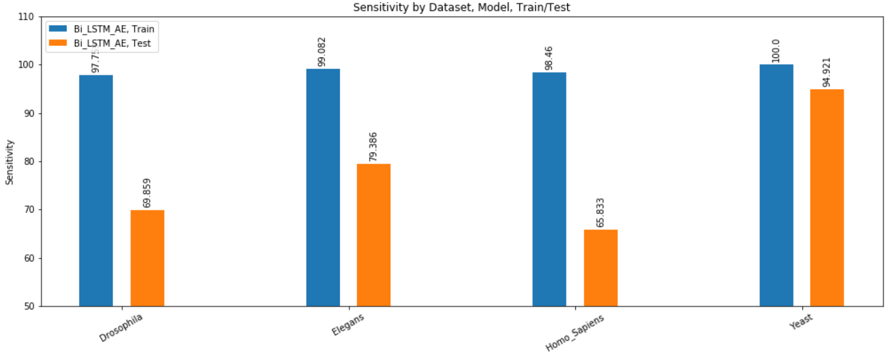


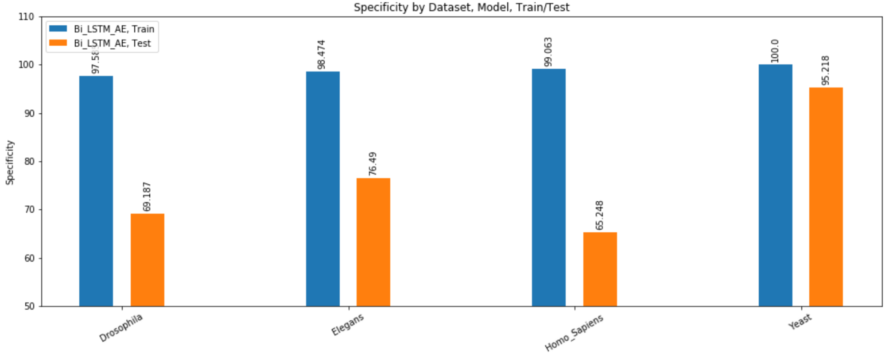
Approaches and Results

- **AutoEncoders**
  - Bi-LSTM AE with FCNN


Accuracy by Dataset, Model, Train/Test


AUC by Dataset, Model, Train/Test


Sensitivity by Dataset, Model, Train/Test


Specificity by Dataset, Model, Train/Test

Conclusion

- **Conclusion of results**

  - It is possible to model short DNA sequences
    - For classification between the sequences being Nucleosomal or Linker
    - With a certain accuracy

  - The **Modified Convolutional LSTM DLNN – CORENup model** is still the **State-of-the-art** based on the results of our experiments.

  - Certain techniques show *potential*:
    - **Combining all MathFeature numerical measures** together, and then perform classification using:
      - **Random Forests**
      - **Deep Forest Architecture**
      - Random Forests have marginally better training performance than Deep Forest, while very similar testing performance

  - Most of the results looks to have **over-fitting issue**
    - Even with the application of **Early Stopping** criterion

Future Work

- **There are many more techniques to explore. Some are:**
    - **GloVe**
        - Unsupervised learning algorithm to obtain Global Vectors for Word representation
    - **ULMFiT**
        - Universal Language Model Fine-tuning for Text Classification
        - An Inductive Transfer Learning method that can be applied to any task in NLP

- **Modifications to presented techniques can be further experimented with:**
    - Deep Forest Architecture directly with Sequences
    - Addition of Bi-LSTM to the CNN – RNN – FCNN (*SOTA*) hybrid architectures
    - Further analysis of the Regularization for the approaches done in the experiment

References

- **Github links**
  - LINKS
- **Papers referred**
  - Papers