

```

} catch (final Exception e) {
    s_logger.error("message: "Failed execute cmd: ", e);
    txn.rollback();
    return new Answer(cmd, false, e.toString());
}

```

# Automated deep-learning based on syntactic context features for log level recommendation

## Background

Log levels are an essential part of software maintenance

- Log level too high -> increase of log management overhead
- Log level too low -> missing information about run-time failures

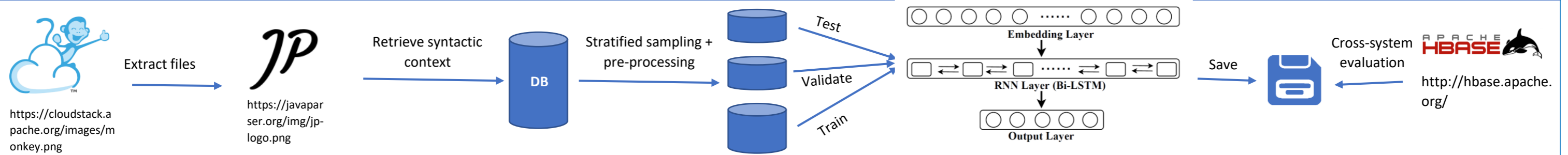
Currently this comes down to developer experience

## Research questions

RQ1: What is the performance of automated deep-learning based on syntactic context features for log level recommendation?

RQ2: Is the approach able to perform cross-system recommendations?

## Methodology



## Results

- Metrics:
- The percentage of correctly recommended log levels (Accuracy)
  - The ability to discriminate between different log levels (AUC)
  - The average distance between the wrong recommendations and the actual log levels (AOD)

	Model trained on CloudStack and evaluated on HBase			Model trained on combined dataset consisting of HBase and CloudStack and evaluated on CloudStack			Model trained on CloudStack and evaluated on HBase		
	Accuracy	AUC	AOD	Accuracy	AUC	AOD	Accuracy	AUC	AOD
Training	94.8	79.8	97.7	76.3	73.0	88.4	26.0	48.9	62.2
Validation	78.1	73.1	88.7	76.6	72.7	88.3	25.9	50.6	62.5
Testing	77.2	72.8	88.5	75.9	72.0	88.0	26.4	48.2	61.9

## Conclusions

- automated deep-learning based on syntactic context features for log level recommendation shows potential, but must be investigated further.
- The model can perform accurate within-system recommendations without specific requirements
  - The model can perform accurate cross-system recommendations without specific requirements
  - The model slightly outperforms a random guessing model for recommendations on systems the model has never seen before
  - The model performs better for larger syntactic context sizes (30-50 nodes)
  - The model performance fluctuates for different containing block types

## Recommendations and future work

- Use custom configurations per feature (basic block type\syntactic context size group)
- Include the basic block types or the syntactic context size groups during stratified sampling
- Enlarge the dataset, especially the sparse features (basic block type\syntactic context size group)
- Optimize the node frequency threshold
- Update the syntactic context extraction such that the order of the nodes remains during conversion to frequency vector
- Capture the type of system around the log statement
- Investigate whether it is better to give the containing block type implicitly or explicitly to the model

## Contact details

Erwin van Dam  
G.B.vanDam@student.tudelft.nl