Machine Learning Based Verification Planning Methodology Using Design and Verification Data

Hanna Jang, Seonghee Yim
Various and complicate

https://m.post.naver.com/viewer/postView.naver?volumeNo=8860727&memberNo=1834
Communicate with data
Same but different : Design and Verification

https://www.samebutdifferentmath.com/early-numeracy
How can I find faster?

SOC Specification

IP Development

Design revision

Veriﬁcation

SOC Development
Focus on difference
Plan between Design and Verification

Design specification
- CMU
- POWER
- GSA
- MIF
- CORE
- DPU
- BUS
- SECURITY

Verification specification
- vPlan1
- vPlan2
- vPlan3
- vPlan4
- vPlan5
- vPlan6

* vPlan : Verification Plan
Data orchestration: Using ML

https://blog.stackadapt.com/data-orchestration/
Data for analysis

Design component

Design Changes

Design Stage

Verification component

Verification Test result
Data pre-processing

• Hash mapping and cleaning

• Vectorize

<table>
<thead>
<tr>
<th>aarch</th>
<th>abox</th>
<th>acc</th>
<th>access</th>
<th>addr</th>
<th>aes</th>
<th>...</th>
<th>alive</th>
<th>ap</th>
<th>apg</th>
<th>apm</th>
<th>armsys</th>
<th>aspm</th>
<th>Run_time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.213673</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0.357441</td>
<td>0.370710</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>229.79</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.23861</td>
<td>0</td>
<td>427.73</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0.187469</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.571612</td>
<td>0</td>
<td>89.44</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.571612</td>
<td>0</td>
<td>16.81</td>
</tr>
</tbody>
</table>
Clustering result
Verification test classification model

AI Algorithm: Deep learning
epoch: 500
Learning rate: 0.001
Batch: 512
Activation function: Relu
Optimizer: AdamW
Accuracy: 31/35
Experimental result

<table>
<thead>
<tr>
<th>Project</th>
<th>SOC Complexity</th>
<th>DV Silicon Bug KPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.52</td>
<td>11.5</td>
</tr>
<tr>
<td>B</td>
<td>0.88</td>
<td>5.7</td>
</tr>
<tr>
<td>C</td>
<td>1.00</td>
<td>n/a</td>
</tr>
<tr>
<td>D</td>
<td>0.67</td>
<td>11.9</td>
</tr>
<tr>
<td>E</td>
<td>0.65</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Conclusion

• Goal
  • Shortening TAT and improving completeness on verification

• Idea
  • The failure pattern generated at each design stage is similar for each project

• Result
  • A verification plan can be established by defining a class using machine learning and failure can be detected early

• additional research
  • Preserve more data
  • Maximize machine learning model performance
Thank you