10 May 2021 Brad's Report

May 9, 2021

1 Overview

- There are 333 results in the search for "Machine Learning" in *Accident Analysis and Prevention*, the journal that Dr. Sun recommended.
- I downloaded most of them over four days. On the UL Library website, I can download 100 per day. Since I had them sorted by relevance, towards the end, some of the articles were so old (1970's) that UL's subscription doesn't cover them.
- I downloaded the BibTeX citations. If you set the UL Library website to view 100 results at a time, you can download all 100 citations as one .bib file. There is no daily limit.
- I skimmed 28 of the articles as of Sunday morning, and am continuing to read.
  - I kept all of my notes in the .bib file.
  - I added an *institution* field for the universities, so I can look for active research hubs.
  - If the article had *suggestions for future research*, I added an *addendum* field.
  - Most articles used a local (city, state, province) database, but if they used some popular database, like SHRP2, and it wasn't listed in the citation already (in the keywords or abstract), I put it in the keywords, so I can see which databases are popular.
  - I read the abstracts and created an *annotation* field.
  - About a third of the articles had nothing to do with machine learning, although the article might have been interesting for other reasons. I put "Not ML" somewhere in the annotation.
  - About half of the articles I flagged for future review by putting "Interesting" somewhere in the annotation.
- I treated the .bib file as a database file, and did what we do with datasets.
  - Who are the prolific authors?
  - What are the most common keywords?
  - Which universities are the research hubs?
  - Which algorithms, metrics, and databases are most mentioned in the keywords and abstracts?
- I also glanced at *Transportation Research, Part C: Emerging Technologies* and ran the same analysis on that dataset. There are 500 results for a search for "Machine Learning" there.

- While there may be differences in how the journals handle keywords, based on the common keywords, I suspect that *Transportation Research, Part C: Emerging Technologies* will be more useful than *Accident Analysis and Prevention*
  - AAP (333 Entries):

'Machine learning': 32, 'Road safety': 14, 'Safety': 13, 'Traffic safety': 9, 'Deep learning': 9, 'Automated driving': 7, 'Crash severity': 7, 'Data mining': 7, 'Support vector machine': 7, 'Connected vehicles': 6, 'Driving simulator': 6, 'Fatigue': 6, 'Driver behavior': 6,

- TRC (500 Entries):

'Machine learning': 44, 'Deep learning': 31, 'None': 18, 'Big data': 15, 'Data mining': 14, 'Clustering': 13, 'Traffic flow': 11, 'Reinforcement learning': 10, 'Prediction': 10, 'Traffic forecasting': 9, 'Autonomous vehicles': 9, 'Social media': 8, 'Car-following': 8, 'Air traffic management': 7, 'Classification': 7, 'GPS': 7, 'Intelligent transportation systems': 7, 'Traffic prediction': 7, 'Neural network': 7, 'Calibration': 7,

2 Sample Augmented BibTeX File

@article{IJAZ2021106094,

title = {A comparative study of machine learning classifiers for injury severity prediction of crashes involving three-wheeled motorized rickshaw},

journal = {Accident Analysis & Prevention},

volume = {154},

pages = {106094},

year = {2021},

issn = {0001-4575},

doi = {https://doi.org/10.1016/j.aap.2021.106094},

url = {https://www.sciencedirect.com/science/article/pii/S0001457521001251},

author = {Muhammad Ijaz and Liu lan and Muhammad Zahid and Arshad Jamal},

keywords = {Three-wheeled motorized rickshaws, Crash severity, Machine learning (ML), Rawalpindi},

abstract = {Motorcycles and motorcyclists have a variety of attributes that have been found to be a potential contributor to the high liability of vulnerable road users (VRUs). Vulnerable Road Users (VRUs) that include pedestrians, bicyclists, cycle-rickshaw occupants, and motorcyclists constitute by far the highest share of road traffic accidents in developing countries. Motorized three-wheeled Rickshaws (3W-MR) is a popular public transport mode in almost all Pakistani cities and is used primarily for short trips to carry passengers and small-scale goods movement. Despite being an important mode of public transport in the developing world, little work has been done to understand the factors affecting the injury severity of three-wheeled motorized vehicles. Crash injury severity prediction is a promising research target in traffic safety. Traditional statistical models have underlying assumptions and predefined associations, which can yield misleading results if flouted. Machine learning(ML) is an emerging non-parametric method that can effectively capture the non-linear effects of both continuous and discrete variables without prior assumptions and achieve better prediction accuracy. This research analyzed injury severity of three-wheeled motorized rickshaws (3W-MR) using various machine learning-based identification algorithms, i.e., Decision jungle (DJ), Random Forest (RF), and Decision Tree (DT). Three years of crash data (from 2017 to 2019) was collected from Provincial Emergency Response Service RESCUE 1122 for Rawalpindi city, Pakistan. A total of 2,743 3W-MR crashes were reported during the study period that resulted in 258 fatalities. The predictive performance of proposed ML models was assessed using several evaluation metrics such as overall accuracy, macro-average precision, macro-average recall, and geometric means of individual class accuracies. Results revealed that DJ with an overall accuracy of 83.7 % outperformed the DT and RF-based on a stratified 10-fold cross-validation approach. Finally, Spearman correlation analysis showed that factors such as the lighting condition, crashes involving young drivers (aged 20–30 years), facilities with high-speed limits (over 60 mph), weekday, off-peak, and shiny weather conditions were more likely to worsen injury severity of 3W-MR crashes. The outcomes of this study could provide necessary and essential guidance to road safety agencies, particularly in the study area, for proactive implementation of appropriate countermeasures to curb road safety issues pertaining to three-wheeled motorized vehicles.},

institution = {Southwest Jiaotong U},

annotation = {Nothing new.},

addendum = {Future studies could seek more advanced techniques such as ensemble and deep learning on other detailed datasets to explore factors contributing to this VRUs group.}

}

[1]:
```
%%latex
\tableofcontents
```

Contents

3 Setup

3.1 Import Libraries

[2]:

```
import bibtexparser
import pandas as pd
import numpy as np
```

3.2 Parse .bib Files, Choose Journal, and Create Pandas Dataframe

[3]:

```
with open('Accident_Analysis_and_Prevention.bib') as bibtex_file:
    bib_database = bibtexparser.load(bibtex_file)
AAP = pd.DataFrame(bib_database.entries)

with open('Transportation_Research_Part_C.bib') as bibtex_file:
    bib_database = bibtexparser.load(bibtex_file)
TRC = pd.DataFrame(bib_database.entries)

P = AAP
```

4 Fields in .bib File

[4]:

```
for row in P:
    print (row)
```

```
abstract
keywords
author
url
doi
issn
```

```
year
pages
volume
journal
title
ENTRYTYPE
ID
addendum
annotation
institution
note
number
```

5 *Accident Analysis and Prevention*

5.1 Keywords

5.1.1 Sort Keywords by Frequency

[5]:

```
P['keywords'] = P['keywords'].fillna('None')
A = [ x.split(', ') for x in P['keywords'].tolist() ]
B = [item for sublist in A for item in sublist]
C = {x:B.count(x) for x in B}
D = dict(sorted(C.items(), key=lambda item: item[1], reverse=True))
for item in D:
    if D[item] > 6:
        print (D[item], item)
```

```
39 None
32 Machine learning
14 Road safety
13 Safety
9 Traffic safety
9 Deep learning
7 Automated driving
7 Crash severity
7 Data mining
7 Support vector machine
```

5.2 Algorithms

5.2.1 Create Dictionary of Algorithms

[6]:

```
Algorithms = {
    'ANN:  Artificial Neural Network': ['Artificial Neural Network'],
    'Bagging': ['Bagging'],
    'Bayesian': ['Bayesian Logistics Regression', 'Bayes'],
    'Binomial Regression': ['Binomial Regression'],
    'Convex Hull Algorithm': ['Convex Hull'],
    'CNN:  Convolutional Neural Network': ['Convolutional Neural Network',␣
 ↪'CNN'],
    'CIF: Cumulative Incidence Function': ['Cumulative Incidence Function'],
    'Decision Jungle': ['Decision Jungle'],
    'Deep Learning': ['Deep Learning', 'deep-learning'],
    'Dimensionality Reduction': ['Dimensionality Reduction'],
    'Dynamic Bayesian Network': ['Dynamic Bayesian'],
    'Ensemble': ['Ensemble'],
    'Ensemble Tree': ['Ensemble Tree'],
    'Feature Extraction': ['Feature Extraction'],
    'Fuzzy Logic': ['Fuzzy Logic'],
    'Genetic Algorithm': ['Genetic Algorithm', 'Genetic Programming'],
    'Hierarchical': ['Hierarchical'],
    'IGA: Intelligent Genetic Algorithm': ['Intelligent Genetic Algorithm'],
    'Logistic Regression': ['Logistic Regression'],
    'LSTM: Long Short-Term Memory': ['Long Short-term Memory'],
    'Marginal Effect Analysis': ['Marginal Effect Analysis'],
    'MDU: Maximum Dissimilarity Undersampling': ['maximum dissimilarity␣
 ↪undersampling'],
    'Mixed Methods': ['Mixed Methods'],
    'Neural Network': ['Neural Network'],
    'Random Forest':['Random Forest'],
    'RSF: Random Survival Forest': ['Random Survival Forest'],
    'Self-Organizing Maps': ['Self-Organizing Maps', 'Self Organizing Maps'],
    'Shapley': ['Shapley'],
    'Statistical Learning': ['Statistical learning'],
    'SMO: Synthetic Minority Oversampling': ['synthetic minority oversampling'],
    't-SNE': ['t-SNE'],
    'VIMP: Variable Importance': ['Variable Importance'],
    'XGBoost':['XGBoost', 'XGB'],

}
```

5.2.2 Find Mentions of Algorithms in Abstracts or Keywords

[7]:

```
for alg in Algorithms:
    P[alg] = P['abstract'].str.contains('|'.join(Algorithms[alg]), case=False)␣
 ↪| P['keywords'].str.contains('|'.join(Algorithms[alg]), case=False)
```

5.2.3 Count Mentions of Algorithms in Abstracts or Keywords

[8]:

```
A = P[Algorithms.keys()].sum()
A.sort_values(ascending=False)
```

[8]:
```
Bayesian                                    45
Neural Network                              33
Random Forest                               28
Logistic Regression                         19
Deep Learning                               16
ANN:  Artificial Neural Network              9
XGBoost                                      9
Hierarchical                                 8
CNN:  Convolutional Neural Network           8
LSTM: Long Short-Term Memory                 8
Genetic Algorithm                            6
Ensemble                                     5
Feature Extraction                           5
SMO: Synthetic Minority Oversampling         4
VIMP: Variable Importance                    4
Statistical Learning                         4
Fuzzy Logic                                  3
Dynamic Bayesian Network                     3
Binomial Regression                          2
Bagging                                      2
Shapley                                      2
t-SNE                                        1
Self-Organizing Maps                         1
RSF: Random Survival Forest                  1
Decision Jungle                              1
Convex Hull Algorithm                        1
Mixed Methods                                1
MDU: Maximum Dissimilarity Undersampling     1
CIF: Cumulative Incidence Function           1
IGA: Intelligent Genetic Algorithm           1
Ensemble Tree                                1
Dimensionality Reduction                     1
Marginal Effect Analysis                     1
dtype: int64
```

5.3 Analysis Tools

5.3.1 Create Dictionary of Analysis Tools

[9]:
```
Analysis_Tools = {
    'Sensitivity': ['Sensitivity'],
    'Area under Curve': ['Area under Curve'],
    'False Alarm Rate': ['False Alarm Rate'],
    'Accuracy': ['accuracy'],
    'Precision': ['macro-average precision'],
    'Recall': ['macro-average recall'],
    'Geometric Mean': ['geometric mean'],
    'Hyperparameters': ['Hyperparameter'],
    'Spearman': ['Spearman'],
    'Aggregated Gain': ['Aggregated Gain'],
    'Time Dependencies': ['Time dependencies'],
    'Temporal': ['Temporal'],
    'Kinematic': ['Kinematic'],
    'Visualization': ['Visualization'],
    'F1 Loss Function': ['F1'],
    'Connected Vehicles': ['Connected Vehicles'],
    'Imbalanced Data': ['Imbalanced Data'],
}
```

5.3.2 Find Mentions of Analysis Tools in Abstracts or Keywords

[10]:

```
for alg in Analysis_Tools:
    P[alg] = P['abstract'].str.contains('|'.join(Analysis_Tools[alg]),␣
 ↪case=False) | P['keywords'].str.contains('|'.join(Analysis_Tools[alg]),␣
 ↪case=False)
```

5.3.3 Count Mentions of Analysis Tools in Abstracts or Keywords

[11]:

```
A = P[Analysis_Tools.keys()].sum()
A.sort_values(ascending=False)
```

[11]:

```
Accuracy              72
Sensitivity           29
Temporal              23
Kinematic             10
Imbalanced Data        9
Connected Vehicles     8
False Alarm Rate       6
Visualization          4
F1 Loss Function       4
Geometric Mean         2
Hyperparameters        2
Aggregated Gain        2
Recall                 1
Area under Curve       1
Time Dependencies      1
Precision              1
Spearman               1
dtype: int64
```

5.4 Datasets

5.4.1 Create Dictionary of Datasets

```
[12]: Datasets = {
          'Second Highway Research Program (Data Set)': ['Second Highway Research␣
       ↪Program', 'SHRP2'],
          'Virginia 100-car Database': ['Virginia', '100-car', '100 car'],
          'NGSIM Trajectory Data': ['NGSIM'],

      }
```

5.4.2 Find Mentions of Dataset in Abstract and Keywords

```
[13]: for x in Datasets:
          P[x] = P['abstract'].str.contains('|'.join(Datasets[x]), case=False) |␣
       ↪P['keywords'].str.contains('|'.join(Datasets[x]), case=False)
```

5.4.3 Count Mentions of Datasets in Abstracts and Keywords

A = P[Datasets.keys()].sum() A.sort_values(ascending=False)

5.5 Authors

5.5.1 Sort Authors by Frequency

```
[14]: P['author'] = P['author'].fillna('None')
      A = [ x.split(' and ') for x in P['author'].tolist() ]
      B = [item for sublist in A for item in sublist]
      C = {x:B.count(x) for x in B}
      D = dict(sorted(C.items(), key=lambda item: item[1], reverse=True))
      for item in D:
          if D[item] > 4:
              print (D[item], item)
```

```
14 Mohamed Abdel-Aty
7 Zhibin Li
6 Junhua Wang
6 Rongjie Yu
6 Pan Liu
5 Asad J. Khattak
5 Ting Fu
5 Mohammed Quddus
5 Jinghui Yuan
5 Mark King
5 Chengcheng Xu
```

5.5.2 Who are these Authors?

5.5.3 Mohamed Abdel-Aty

- U of Central Florida
- Editor in Chief Emeritus of the journal
- PhD from Davis

5.5.4 Zhibin Li

- Southeast University, Nanjing

5.5.5 Junhua Wang

- Tongji U, Shanghai

5.5.6 Rongjie Yu

- Coauthor with Mohamed Abdel-Aty
- Tongji U, Shanghai

5.5.7 Pan Liu

- Southeast University, Nanjing
- Coauthors:
  - Jie Bao (2)
  - Satish V. Ukkusuri
  - Xiao Qin
  - Huaguo Zhou
  - Yanyong Guo
  - Zhibin Li (2)
  - Yao Wu
  - Wei Wang (2)
  - Chengcheng Xu (2)

5.5.8 Asad J. Khattak

- U of Tennessee

5.6 Institutions

- Note that the Institutions aren't in the database until I manually add them. ### Sort Institutions by Frequency

[15]:

```
x = 'institution'
P[x] = P[x].fillna('None')
A = [ x.split(', ') for x in P[x].tolist() ]
B = [item for sublist in A for item in sublist]
C = {x:B.count(x) for x in B}
D = dict(sorted(C.items(), key=lambda item: item[1], reverse=True))
D
```

[15]:

```
{'None': 308,
 'Louisiana State U': 3,
 'Tsinghua U': 2,
 'Tongji U': 2,
 'U of Central Florida': 2,
 'Southeast U': 2,
 'Nanjing': 2,
 'Queensland U of Technology': 2,
 'U of Natural Resources and Life Sciences': 2,
 'Vienna': 2,
 'North Dakota State U': 1,
 'Southwest Jiaotong U': 1,
 'Shanghai': 1,
 'Hefei U of Technology': 1,
 'Changsha U of Technology': 1,
 'Nanyang Technological U': 1,
 'Oak Ridge National Laboratory': 1,
 'Virginia Transportation Research Council': 1,
 'Northwestern U': 1,
 'Shahid Bahonar U': 1,
 'Texas A\\&M U': 1,
 'Nanyang U': 1,
 'City University of Hong Kong': 1,
 'Texas A \\& M U': 1,
 'Federal University of Rio Grade do Sul (Brazil)': 1,
 'Federal Rural University of Semi-Arid (Brazil)': 1,
 'Jiangsu U': 1,
 'Deft U': 1}
```

5.7 Interesting Articles

[16]:

```python
P['annotation'] = P['annotation'].fillna('None')
Interesting = P[P['annotation'].str.contains('Interesting', case=False)]
Interesting['title']
```

[16]:

```
12     A data-driven, kinematic feature-based, near r…
71     A deep learning based traffic crash severity p…
84     A Bayesian modeling framework for crash severi…
106    A hierarchical machine learning classification…
124    A multivariate analysis of environmental effec…
142    A contextual and temporal algorithm for driver…
148    A feature learning approach based on XGBoost f…
157    A long short-term memory-based framework for c…
161    A methodology to design heuristics for model s…
188    A Comprehensive Railroad-Highway Grade Crossin…
197    A forward collision avoidance algorithm based …
239    A genetic programming approach to explore the …
```

```
Name: title, dtype: object
```

5.8 Not Machine Learning

[17]:

```
A = P[P['annotation'].str.contains('Not ML', case=False)]
A['title']
```

[17]:

```
34     A Bayesian Tobit quantile regression approach …
57     A crash risk identification method for freeway…
81     A comparative study of state-of-the-art drivin…
84     A Bayesian modeling framework for crash severi…
101    A driver behavior assessment and recommendatio…
116    A crash prediction method based on bivariate e…
130    "It is frustrating to not have control even th…
170    A multivariate-based variable selection framew…
197    A forward collision avoidance algorithm based …
239    A genetic programming approach to explore the …
Name: title, dtype: object
```

6 *Transportation Research Part C: Emerging Technologies*

[18]:

```
P = TRC
```

6.1 Keywords

[19]:

```
P['keywords'] = P['keywords'].fillna('None')
A = [ x.split(', ') for x in P['keywords'].tolist() ]
B = [item for sublist in A for item in sublist]
C = {x:B.count(x) for x in B}
D = dict(sorted(C.items(), key=lambda item: item[1], reverse=True))
for item in D:
    if D[item] > 6:
        print (D[item], item)
```

```
44 Machine learning
31 Deep learning
18 None
15 Big data
14 Data mining
13 Clustering
11 Traffic flow
10 Reinforcement learning
10 Prediction
9 Traffic forecasting
9 Autonomous vehicles
```

```
8 Social media
8 Car-following
7 Air traffic management
7 Classification
7 GPS
7 Intelligent transportation systems
7 Traffic prediction
7 Neural network
7 Calibration
```

[ ]:

```
```