Oct 31th, 2019

Leveraging AI in IT Service Automation Modeling: from Classical AI Through Deep Learning to Combination Models

Qing Wang[1], Laura Shwartz[1], Genady Ya. Grabarnik[2], Michael Nidd[3], Jinho Hwang[1]

1.IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
2.Dept. Math & Computer Science, St. John's University, Queens
3.IBM Research Zurich, 8803 Rueschlikon, Switzerland

Agenda

- Introduction into Service Management
- System Overview
- Problem Definition & Methodology
- Experiments
- Conclusion

Service Management

- Service Management Workflow

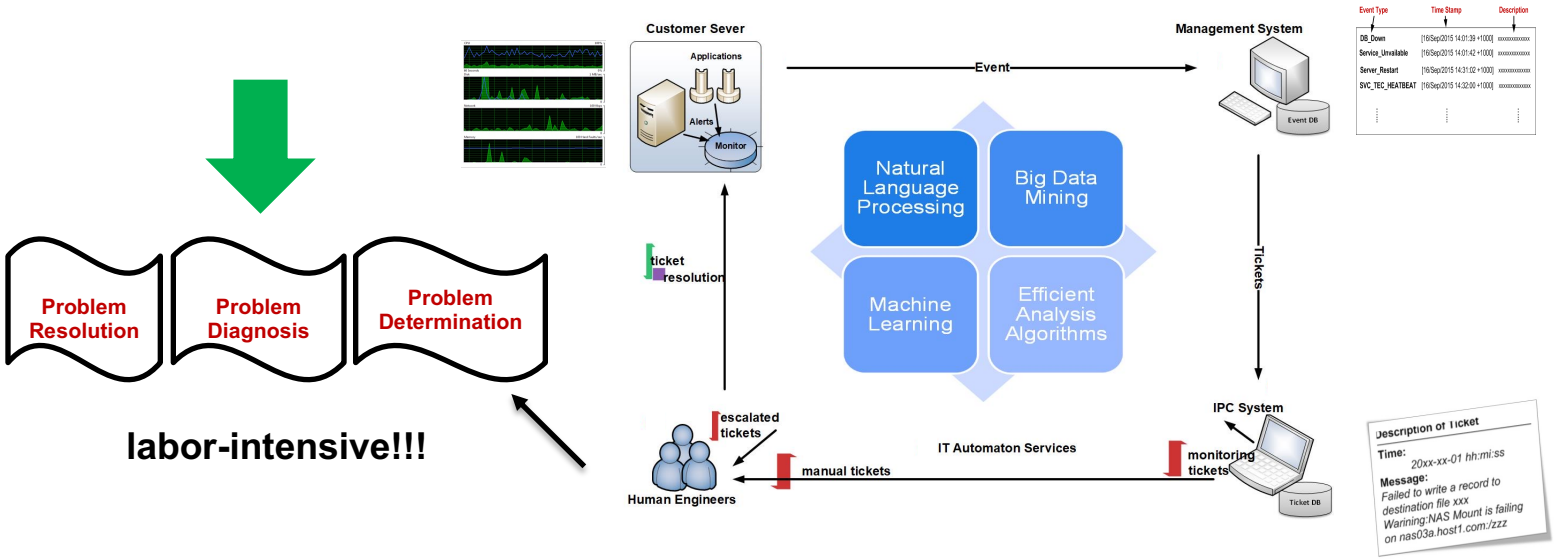


Figure 3: A Typical Workflow of Service Management

System Overview

- System Architecture

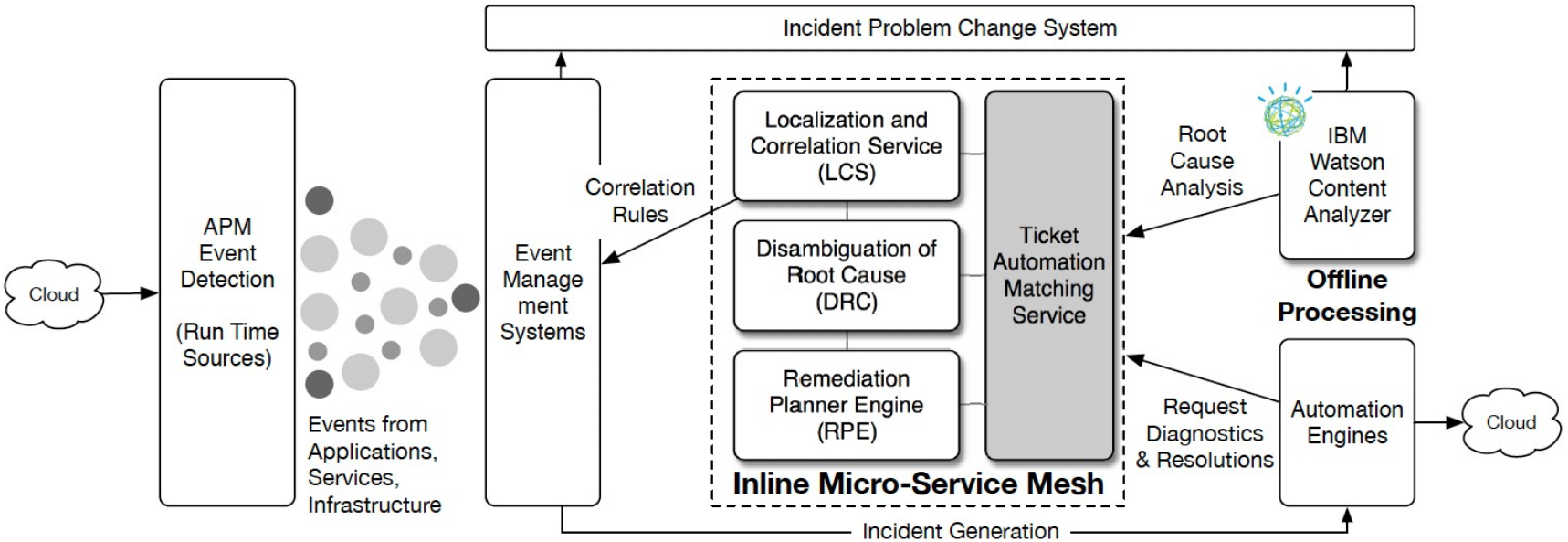


Figure 4: Cognitive Event Automation System Architecture

Challenges with monitoring data

- Data produced by different monitors (and systems) is highly variable
- New monitors are continuously added

ALERT_KEY	XXX_logalrt_x072_aix		AUTOMATION	Disk Path Checker	
AGENT	CUSTOMER_CODE	ALERT_GROUP	COMPONENT		OSTYPE
EIF Probe on xxxxxx	XXX	ITM_XXX_LOGFILEMONITOR_LOG_FILE	Computer System		Generic
TICKET_GROUP	PRIORITY	HOSTNAME	IP_ADDRESS		SUBCOMPONENT
I-XXX-XXX-DS	PX	XXX	XXX.XXX.XXX.XXX		Log
TICKET SUMMARY	LogEvent: Thu Apr 4 02:00:01 EDT2019,xxxxxx,pcmpath,disk,fiber adapters missing or failed on server		TICKET DESCRIPTION	4 xxxxxx GENERIC LOG /TMP/xxxxxx.LOG AIX is CRITICAL **	

ALERT_KEY	XXX_erp_xlo2_std		AUTOMATION	Disk Path Checker	
AGENT	CUSTOMER_CODE	ALERT_GROUP	COMPONENT		OSTYPE
EIF Probe on xxxxxx	XXX	ITM_XXX_LOGFILEEVENTS	Operating System		AIX
TICKET_GROUP	PRIORITY	HOSTNAME	IP_ADDRESS		SUBCOMPONENT
NUSN_XXDISTOPS	PX	XXX	XXX.XXX.XXX.XXX		ErrorReport
TICKET SUMMARY	Errpt log entry: xxxxxx 0404051519 P H - PATH HAS FAILED - hdisk21		TICKET DESCRIPTION	xxx.xxx.xxx.IBM.COM AIX ERRORREPORT /VAR/ADM/RAS/ERRLOG.HDISK21 UNIX is CRITICAL **	

Figure 2: Two different monitoring tickets and the same matching automation.

State of the art for automated resolution of the events

- Automated system mostly employ regular expressions which are used as matchers to specific automation

- As monitors changed and added the number of ‘matchers’ is growing.
  - IBM Global services have around 25,000 regular expressions within 4 years
  - Maintaining matchers become difficult or impossible task
- Artificial intelligence techniques are introduced to choose the ‘correct’ automation for a monitoring ticket with the following challenge:
  - Feature selection
  - Deep learning models

How does the matcher service effectively achieve and maintain high accuracy on noisy tickets while automatically adapting to an introduction of a new or changed ticket contents?

Problem Definition & Methodology

- What is the best methodology for solving this challenge

  - Based on the ticket's content (see Figure 2), recommending the best automation can be formulated as a multiclass text classification problem.

  - A set of training data: $D = \{(\boldsymbol{x_t}, y_t)\}, \mathrm{t} = 1,2,\ldots,\mathrm{N}$.
    - $\boldsymbol{x_t} \in R^d$ is the $\boldsymbol{d}$ **– dimension** feature representation for ticket $\boldsymbol{x_t}$.
    - $y_t \in Y = \{1,2,\ldots,K\}$ is the class label for ticket $\boldsymbol{x_t}$.
  - The prediction function is need to learn:
    - $g(x): R^d \mapsto Y$

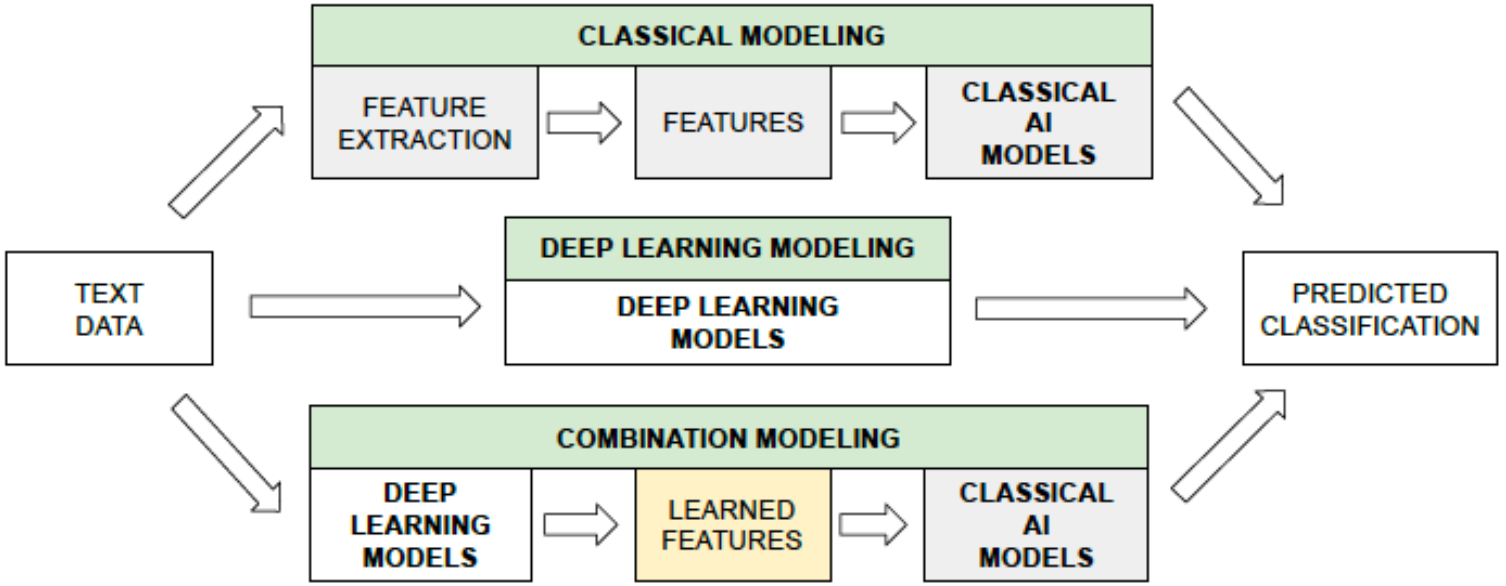
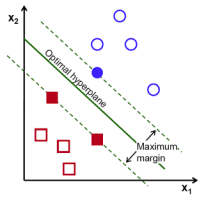


Figure 5: Using modeling for the multiclass text classification: classical AI vs deep learning vs combination.

Classical AI models



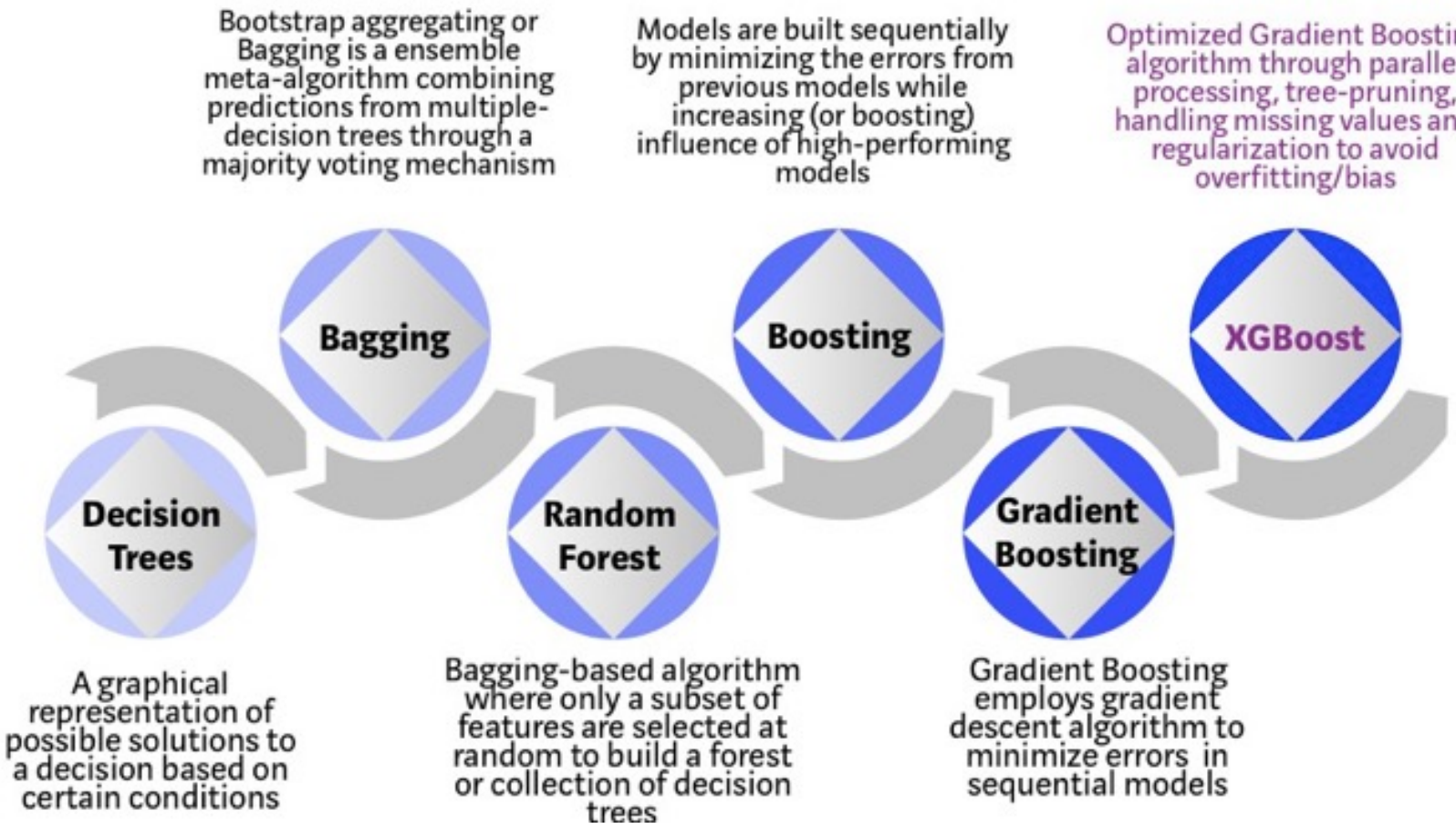
- Classical AI Models:
  - Classical AI models usually work with relatively low-dimension attribute spaces, necessitating well-defined and highly informative attributes as coordinates of feature vectors.
  - We use domain experts' assistance to determine such attributes for the ticket dataset
  - Example: Support Vector Machines:
    - an efficient, theoretically solid and strong baseline for text classification problem
  - Ensemble Methods
    - Train multiple classifiers and apply voting to make final predictions.
    - More accurate than a single classifier
    - Bagging and Boosting
    - Example: Random Forests

      (1) a highly accurate and robust machine learning algorithm.

      (2) capable of modeling large feature spaces

      (3) an ensemble of decision trees

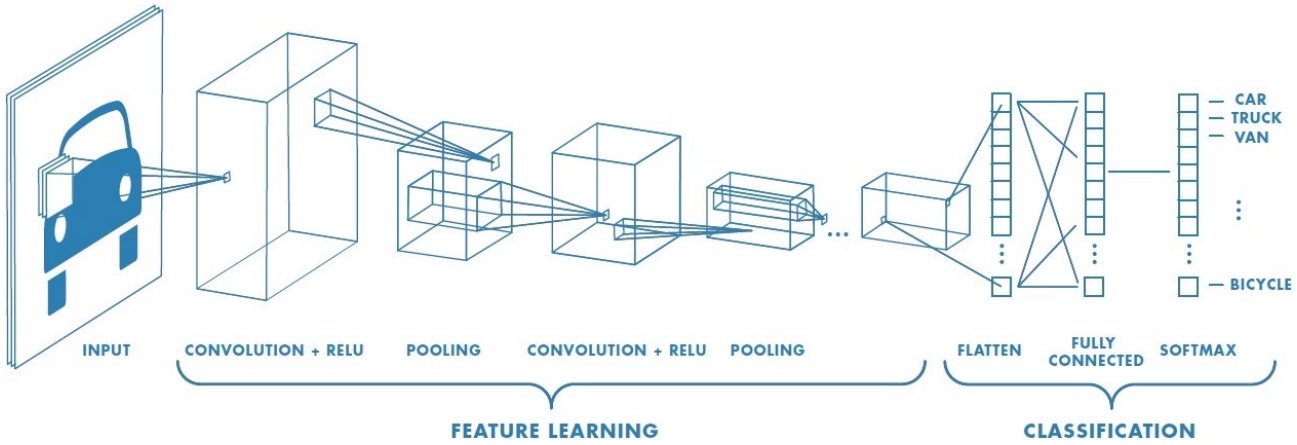
Ensemble methods



- This methods require considerable effort on text preprocessing and feature extraction
- Due to the nature of continuously-changing ticket records automatic feature extractors or selectors are absolutely critical

Deep Learning Methods

- Deep Learning: Convolutional Neural Networks
  - have been shown to be effective in many Computer Vision and NLP tasks.
- Convolution is the first layer to extract features from an input
- ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = max(0,x)$. Why ReLU is important : ReLU's purpose is to introduce non-linearity
- Pooling layers section would reduce the number of parameters
- Fully Connected layer, we flattened our matrix into vector and feed it into a fully connected layer like neural network
- In our case the layers are
    - word embedding layer,
    - fully connected layer and
    - dropout layer
  - The introduction of a dropout layer is a regularization technique that reduces overfitting.



Combination Models

- Combination Model
  - CNNs is used for learning feature representation:
    - convolution feature filters with varying widths captures several different semantic classes of ngrams by using different activation patterns
    - global maxpooling function induces behavior which separates important ngrams from the rest

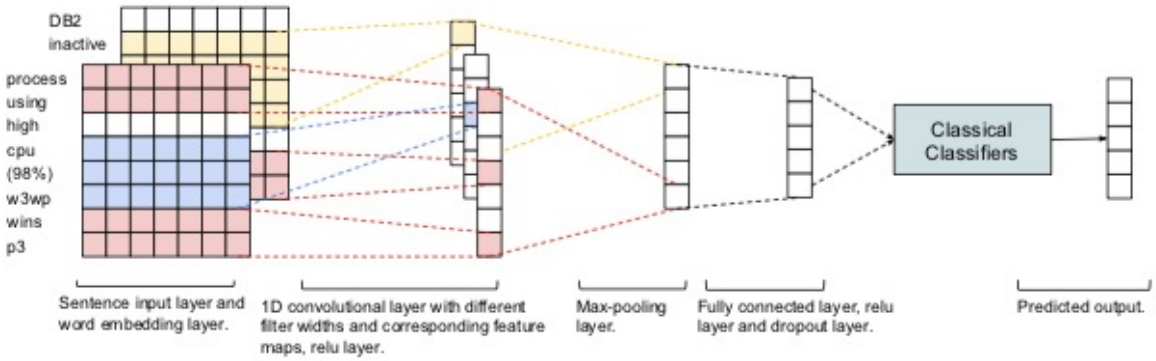


Fig. 4. Architecture of combination models on multiclass text classification tasks.

Data sets

– Experimental incident data is generated by a variety of monitoring systems and stored in the Operational Data Lake.
– It contains |D| = 100, 000 tickets from Jan. 2019 to Apr. 2019.
– There are 114 automations (i.e., 114 classes/labels) in the dataset and a vocabulary V of size |V | = 184, 936
– All incidents used in experiments are automatically resolved

– After some preliminary testing, we designed our primary experiments to
  – Randomly initialize all word vectors with a dimension of 300
  – use ReLU, filter size of 4 × 5 with 64 feature maps each (for CNN only), dropout rate of 0.25, mini-batch size of 128, and epoch number of 20.

Table 1. Dataset summary for classical AI, deep learning and combination modeling.

Models	Training	Validation	Testing	Classes	|CV|
Classical AI models	80,000	–	20,000	144	5
Deep AI models	64,000	16,000	20,000	144	5
Combination models	80,000	–	20,000	144	5

Results

- The accuracy (ACC) and F1-score (F1) are widely applied metrics for evaluating multiclass classifiers.

Table 2. Performance comparison on **Accuracy (ACC(in percent %)), F1-macro (F1(in percent %)), Time Cost (t(in seconds)).**

	$\|\mathcal{D}\| = 4,000$			$\|\mathcal{D}\| = 20,000$			$\|\mathcal{D}\| = 100,000$		
Models	**ACC(%)**	**F1(%)**	**t(s)**	**ACC(%)**	**F1(%)**	**t(s)**	**ACC(%)**	**F1(%)**	**t(s)**
Linear SVM [21]	97.95	88.18	3.60	99.09	92.42	42.81	99.53	93.69	671.97
Decision Tree [22]	97.65	84.71	0.11	98.58	79.96	1.13	98.15	62.74	16.43
KNeighbors [23]	93.75	75.20	0.15	97.39	78.01	3.72	97.80	80.46	99.29
K-Means [24]	< 50.00	–	78.01	< 50.00	–	625.13	< 50.00	–	5960.72
Random Forests [11]	97.65	89.26	1.15	99.05	92.28	13.25	99.29	93.39	251.26
XGBoost [15]	**98.50**	**91.79**	122.06	99.22	89.97	814.90	99.12	79.85	5345.62
MLP [2]	96.37	82.78	2.62	98.85	88.79	18.38	99.23	93.72	251.35
CNN [3]	97.12	81.10	8.65	98.92	88.40	52.87	99.39	93.16	601.11
CNN-SVM [17]	98.77	87.46	145.13	99.48	92.54	403.25	99.79	96.07	3019.69
CNN-Random Forests	98.75	87.92	148.24	99.54	90.01	148.24	**99.80**	**95.90**	1939.16
CNN-XGBoost [25]	93.50	67.41	260.19	97.70	72.15	1804.07	98.75	82.53	14035.91

Conclusion

- Classical AI models perform well when the data size is small; they require handcrafted features
- Deep learning models achieve a better performance when the training data is large enough
- Combination models have the best performance on all dataset sizes and do not require engineered features

Back up