intel AI

使用深度学习、RAY和ANALYTICS ZOO进行自动化时间序列分析

Intel *Analytics Zoo* Team

ANALYTICS ZOO

Mar 13, 2020

Agenda

- **Background**
  - Introduction of *Analytics Zoo*
  - Background about *Time Series Analysis*
  - Background about *AutoML* and *Ray*
- **Time Series Analysis using AutoML and Ray on Analytics Zoo**
- **Use Case Sharing**

Background

What is Analytics Zoo

BigDL

Distributed, High-Performance

Deep Learning Framework

for Apache Spark

https://github.com/intel-analytics/bigdl

ANALYTICS ZOO

Unified Analytics + AI Platform

Distributed TensorFlow, Keras, PyTorch and BigDL on Apache Spark

https://github.com/intel-analytics/analytics-zoo

Accelerating Data Analytics + AI Solutions At Scale

Mar 13, 2020

Unified Big Data Analytics and AI Platform

Seamless Scaling from Laptop to Production

- Easily prototype the integrated data analytics & AI solution
- "Zero" code change from laptop to distributed cluster
- Directly access production data (Hadoop/Hive/HBase) without data copy
- Seamlessly deployed on production big data clusters

Analytics Zoo

Unified Big Data Analytics and AI Platform

Models & Algorithms	Recommendation	Time Series	Computer Vision	NLP
ML Workflow	AutoML for Time Series		Automatic Cluster Serving	
Integrated Analytics & AI Pipelines	Distributed TensorFlow & PyTorch on Spark		RayOnSpark	
	Spark Dataframes & ML Pipelines for DL		Model Serving	

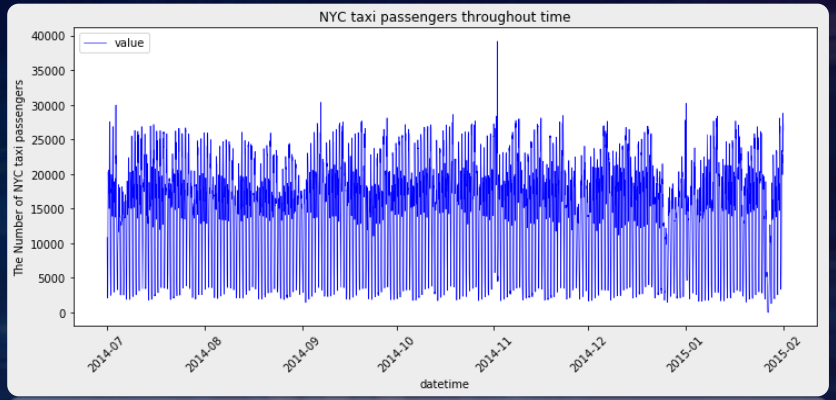
Library & Framework	Distributions (Cloudera/Databricks/....)	Distributed Analytics (Spark/Flink/Ray/...)	DL Frameworks (TF/PyTorch/...)	Python Libraries (Numpy/Pandas/...)

https://github.com/intel-analytics/analytics-zoo

Time Series Analysis

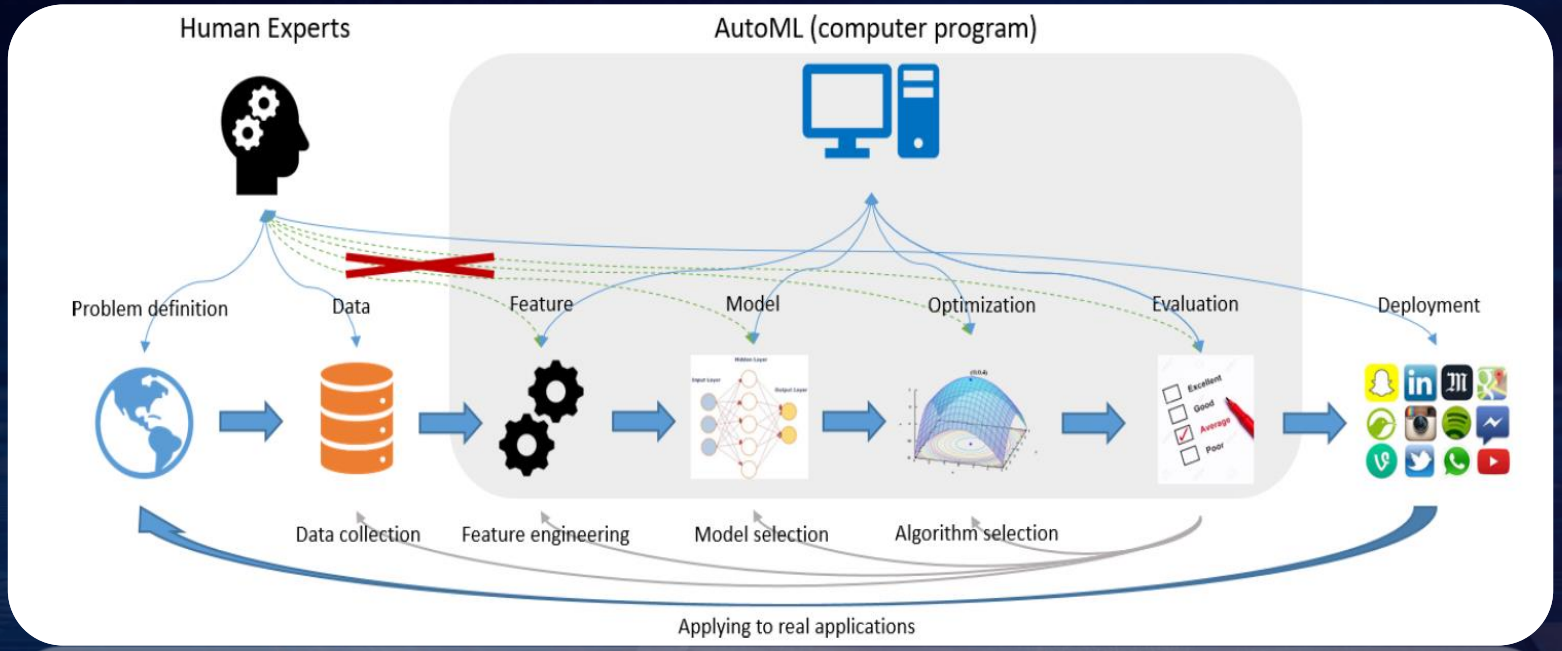
- **Time Series data**
  - A series of data that is observed sequentially in time.
  - Numerical & unstructured
  - Stock prices, sales volume, CPU/IO monitoring data, etc.

- **Example of time series analysis**
  - Product demand prediction
  - Network quality analysis
  - Predictive maintenance for high-value equipment



Total volume of taxi passengers in NYC from 2014/07-2015/02 (source : https://github.com/intel-analytics/analytics-zoo/blob/master/apps/anomaly-detection/anomaly-detection-nyc-taxi.ipynb)

AutoML Overview



Taking the Human out of Learning Applications : A Survey on Automated Machine Learning. Yao, Q., Wang, et. al

Ray and Ray On Spark

- **Ray**
  - A distributed framework for emerging AI applications
- **RayOnSpark**
  - Directly run Ray programs on big data cluster
  - Seamlessly integrate ray into spark data processing pipeline

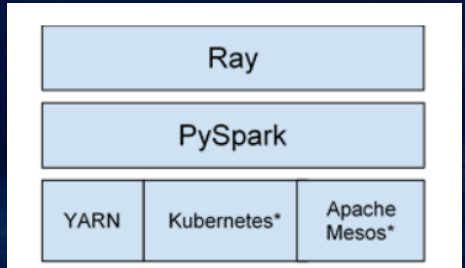


Figure1: Deploy Ray* on Apache Spark*

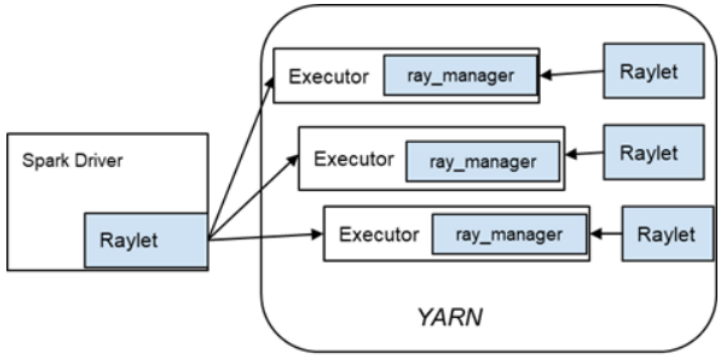


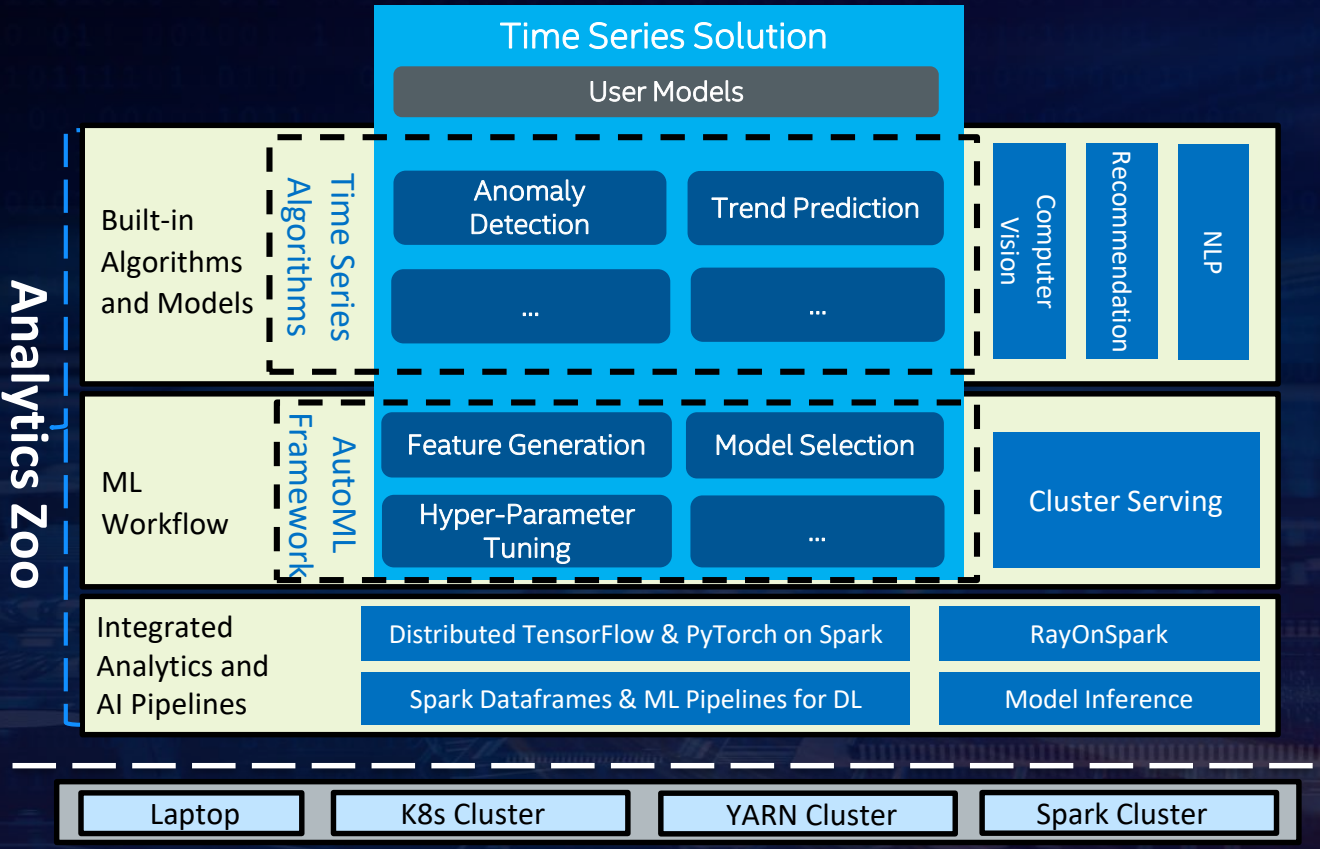
Figure 2: Launch Ray* process on Apache Spark*

https://ray.readthedocs.io/en/latest/
https://analytics-zoo.github.io/master/#ProgrammingGuide/rayonspark/

Time Series Analysis using AutoML and Ray on Analytics Zoo

Mar 13, 2020

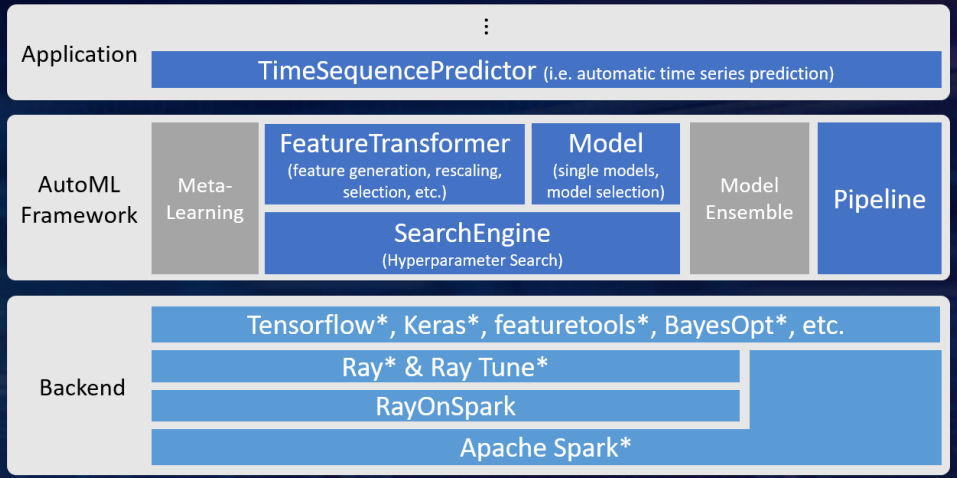
Time Series Solution In Analytics Zoo



- **Time series Applications**
  - Time series forecasting
  - Anomaly detection
  - Time Series Clustering
  - etc
- **AutoML**
- **Seamless scaling**
- **Full-stack Intel SW+HW Optimization w/ Analytics Zoo**

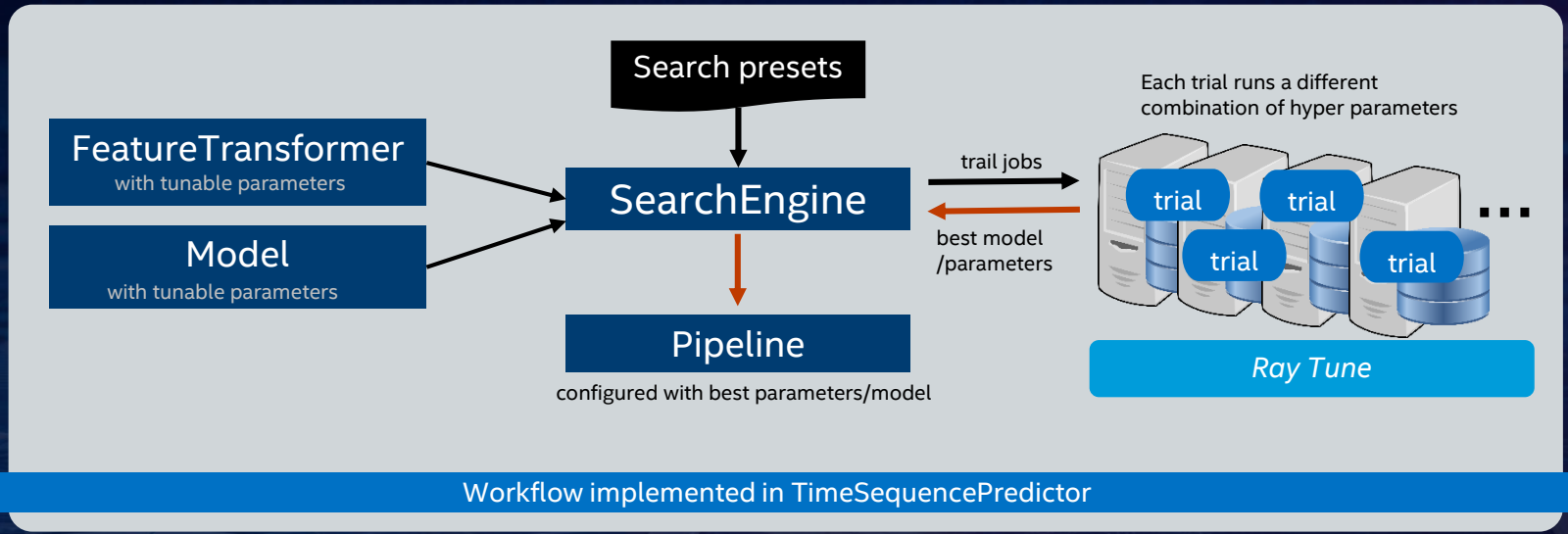
AutoML + Time Series Analysis Framework In *Analytics Zoo*

- **AutoML Framework**
  - FeatureTransformer
  - Model
  - SearchEngine
  - Pipeline
- **Time Series Prediction w/ AutoML**
  - TimeSequencePredictor
  - TimeSequencePipeline



https://medium.com/riselab/scalable-automl-for-time-series-prediction-using-ray-and-analytics-zoo-b79a6fd08139

Typical Workflow of Training w/ AutoML



General API Usage

- Training a **Predictor**
  - **fit** (w/ automl)
  - recipe
  - distributed

```
from zoo.automl.regression.time_sequence_predictor import TimeSequencePredictor
tsp = TimeSequencePredictor( dt_col="datetime",
                             target_col="value",
                             extra_features_col=None,
                             future_seq_len=1)
pipeline = tsp.fit(train_df,
                   metric="mean_squared_error",
                   recipe=RandomRecipe(num_samples=100),
                   distributed=True)
```

- Using a **Pipeline**
  - save/load
  - evaluate/predict
  - fit (incremental)

```
pipeline.save("/tmp/saved_pipeline/my.ppl") #save

from zoo.automl.pipeline.time_sequence import load_ts_pipeline
pipeline = load_ts_pipeline("/tmp/saved_pipeline/my.ppl") #load
rs = pipeline.evaluate(test_df, metric=["r_square"]) # evaluation
result_df = pipeline.predict(test_df) # inference
pipeline.fit(newtrain_df, epoch_num=5) # incremental training
```

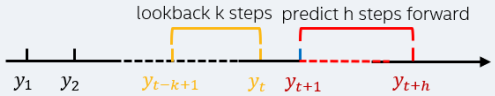
Application: Time Series Forecasting

Time series forecasting
Anomaly Detection
Time series clustering
Characterization
Time series Transformation
Representation Learning
Pattern Discovery

Time Series Application

Definition

- Given all history observations $y_1, \dots, y_t$, Predict values of next **h** steps, $y_{t+1}, \dots, y_{t+h}$
- Usually only lookback **k** steps, $y_{t-k+1}, \dots, y_t$

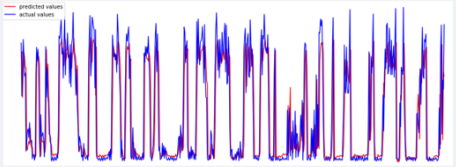


Use Case

- Sales volume/demand prediction.
- As a pre-step for Anomaly Detection
- Commonly used in AIOps: Resource planning, Anomaly Detection, etc.

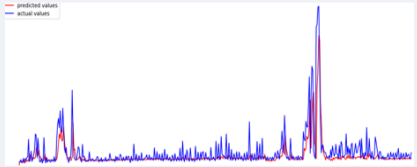
Patterns to Forecast

Varying Length of Periods



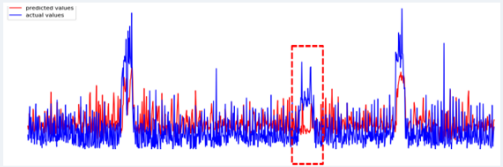
Traditional method does not forecast well when the length of the periods changes over time, while our solution works much better.

Varying Peak Value



Traditional method does not forecast well when peak values changes over time., while our solution works much better.

Various Forms of Waves



Our time series forecasting can tolerate various (normal) forms of waves, while still recognize abnormal forms.

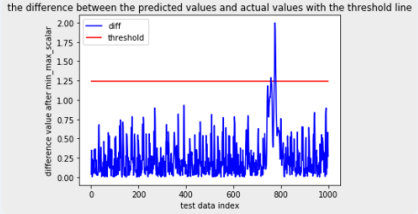
Application: Anomaly Detection

Time Series Application

- Time series forecasting
- **Anomaly Detection**
- Time series clustering
- Characterization
- Time series Transformation
- Representation Learning
- Pattern Discovery

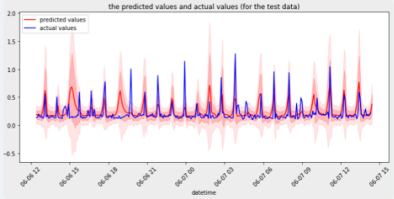
Anomaly Detection after Time Series Forecasting

Threshold w/ X Percentile



Use x Percentile of the diff between predicted and actual values (99 percentile as we use in our anomaly example)

Threshold w/ Uncertainty



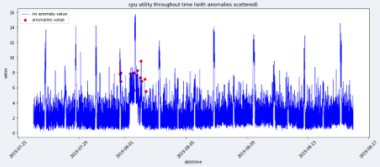
Based on Monte Carlo Dropout (refer to "Deep and Confident Prediction for Time Series at Uber" paper)

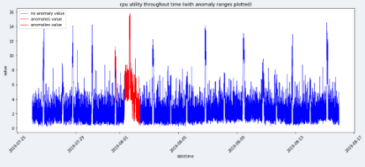
Use Case

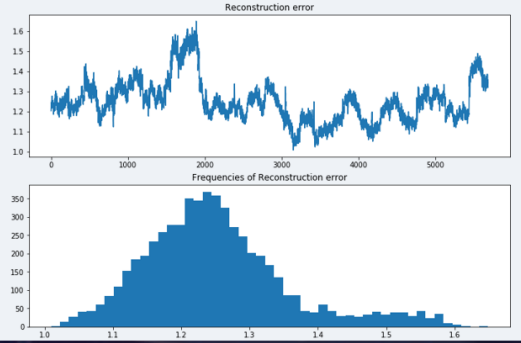
- Alerts generation
- Resource utilization anomaly detection
- ...

Anomaly Detection w/o Time Series Forecasting

Anomaly detection can also be done without forecasting, using an autoencoder and analyze the reconstruction error.



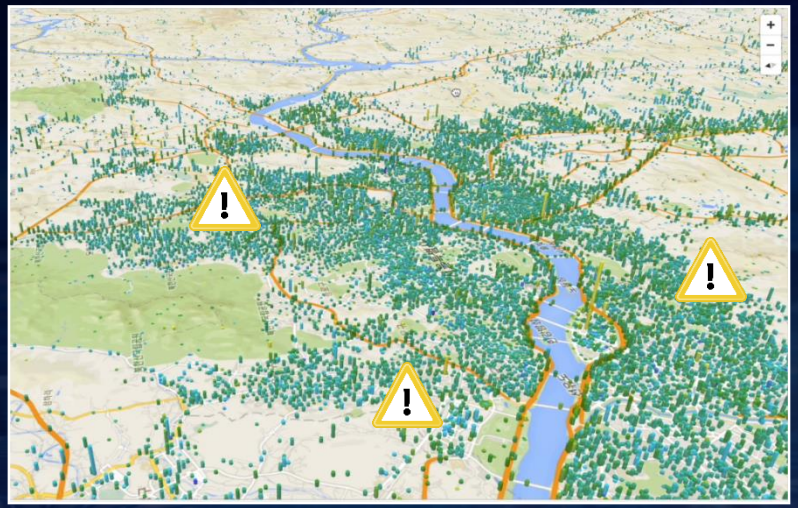


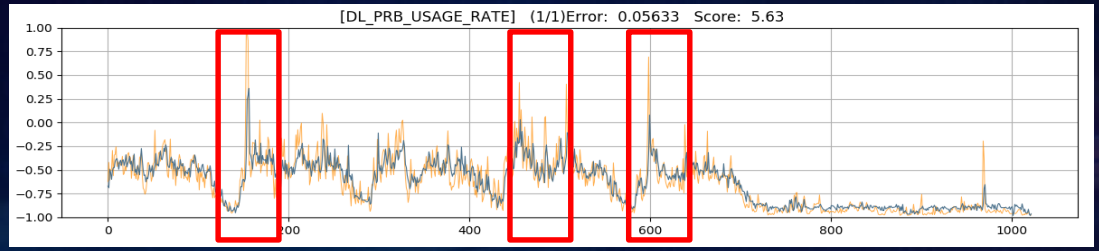


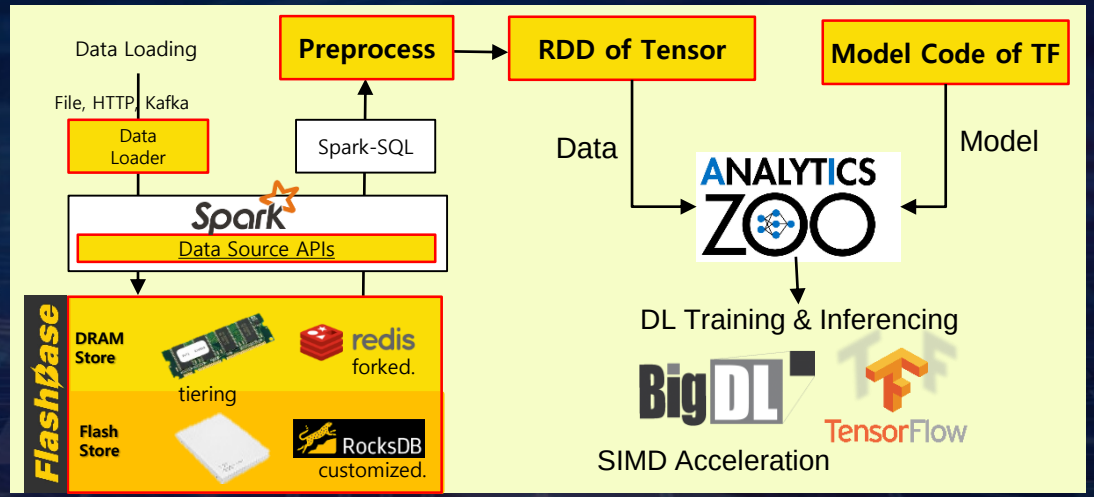
Use case sharing

Mar 13, 2020

Time Series Based Network Quality Prediction in SK Telecom

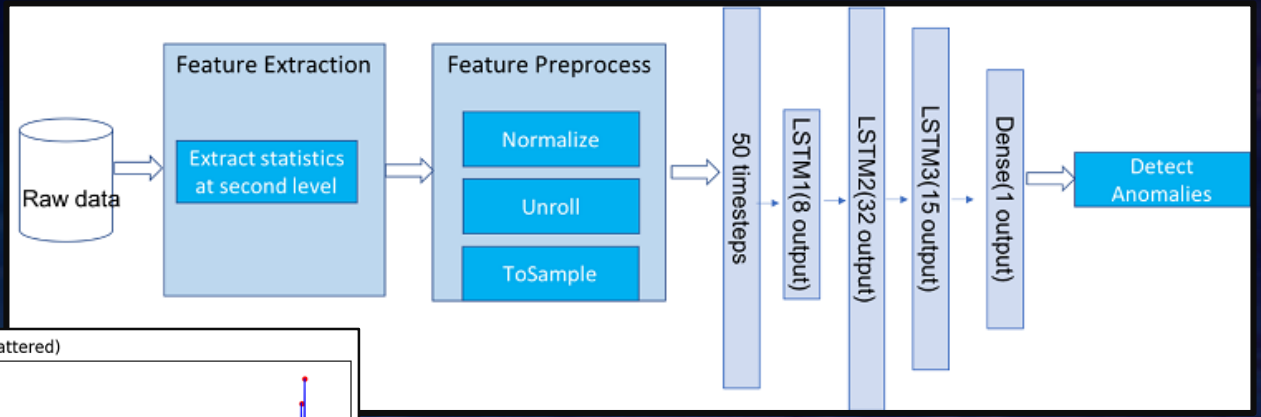


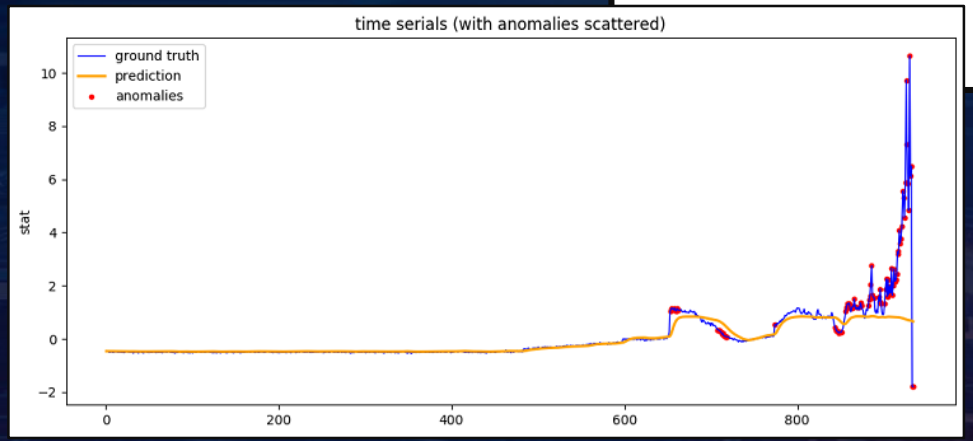




https://databricks.com/session_eu19/apache-spark-ai-use-case-in-telco-network-quality-analysis-and-prediction-with-geospatial-visualization

Unsupervised Time Series Anomaly Detection for Baosight

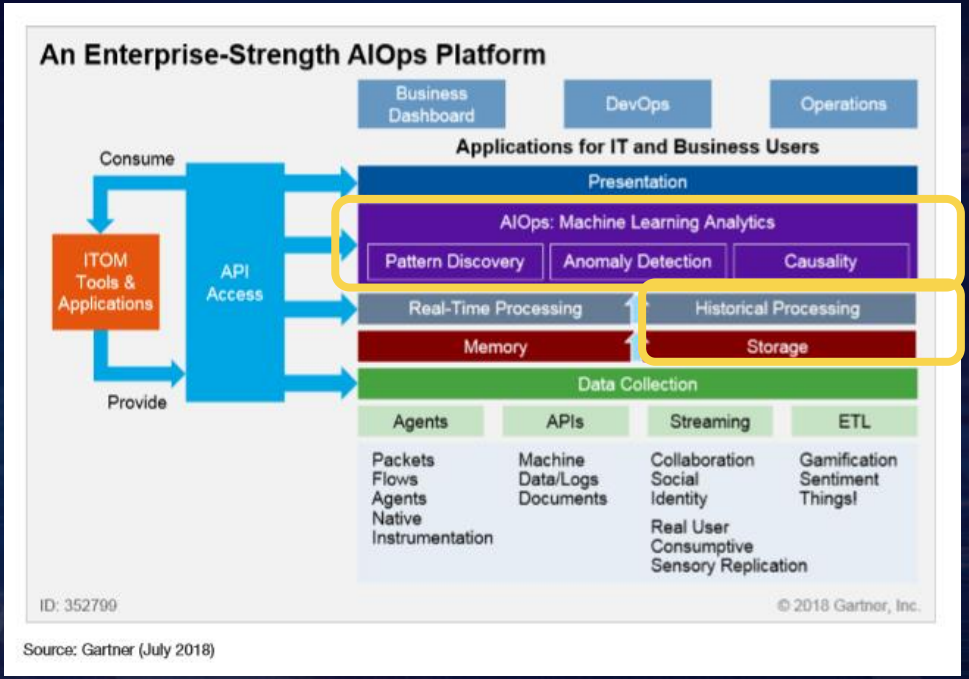




https://software.intel.com/en-us/articles/lstm-based-time-series-anomaly-detection-using-analytics-zoo-for-apache-spark-and-bigdl

Yunda: Anomaly Detection for AIOps

- AIOps
- Monitoring log/metrics analysis for data center operations
- AIOps helps cost saving and MTTR (mean-time-to-repair)



Source: Gartner (July 2018)

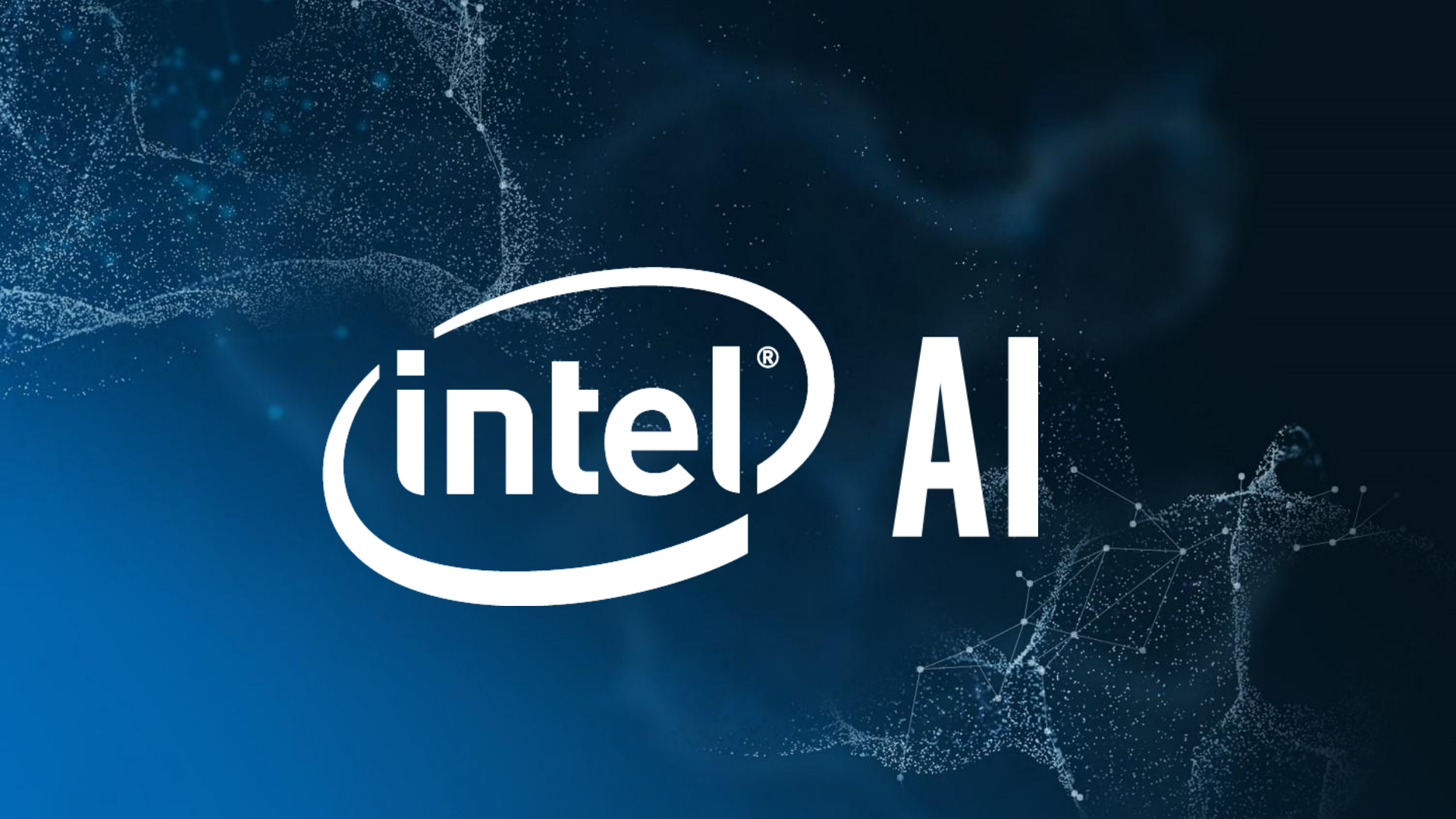
https://www.intel.cn/content/www/cn/zh/analytics/artificial-intelligence/yunda-brings-quality-change-to-the-express-delivery-industry.html

More Information about AutoML + Time Series in Analytics Zoo

- **Scalable AutoML for Time Series Analysis**
  - Source code as a branch of analytics-zoo repo @ https://github.com/intel-analytics/analytics-zoo/tree/automl
  - README @ https://github.com/intel-analytics/analytics-zoo/blob/automl/pyzoo/zoo/automl/README.md
  - Blog https://medium.com/riselab/scalable-automl-for-time-series-prediction-using-ray-and-analytics-zoo-b79a6fd08139
- **Anomaly Detection Reference Examples**
  - Time Series Forecast w/ AutoML https://github.com/intel-analytics/analytics-zoo/blob/automl/apps/automl
  - Anomaly Detection based on Forecast https://github.com/intel-analytics/analytics-zoo/tree/master/apps/anomaly-detection
  - Anomaly Detection based on AutoEncoder https://github.com/intel-analytics/analytics-zoo/tree/master/apps/anomaly-detection-hd
- **Real-world Customer Applications**
  - Baosight's anomaly detection for intelligent equipment management. Details refer to http://software.intel.com/en-us/articles/lstm-based-time-series-anomaly-detection-using-analytics-zoo-for-apache-spark-and-bigdl
  - Yunda anomaly detection for AIOps https://www.intel.cn/content/www/cn/zh/analytics/artificial-intelligence/yunda-brings-quality-change-to-the-express-delivery-industry.html

More Information on Analytics Zoo

- Project website
  - https://github.com/intel-analytics/analytics-zoo
  - https://github.com/intel-analytics/bigdl
- Tutorials
  - CVPR 2018: https://jason-dai.github.io/cvpr2018/
  - AAAI 2019: https://jason-dai.github.io/aaai2019/
- "BigDL: A Distributed Deep Learning Framework for Big Data"
  - *In proceedings of ACM Symposium on Cloud Computing 2019 (SOCC'19)*
- Use cases
  - *Azure, CERN, MasterCard, Office Depot, Tencent, Midea, etc.*
  - https://analytics-zoo.github.io/master/#powered-by/


intel®
AI

LEGAL NOTICES AND DISCLAIMERS

- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations, and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit intel.com/performance.
- Intel does not control or audit the design or implementation of third-party benchmark data or websites referenced in this document. Intel encourages all of its customers to visit the referenced websites or others where similar performance benchmark data are reported and confirm whether the referenced benchmark data are accurate and reflect performance of systems available for purchase.
- Optimization notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.
- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com/benchmarks.
- Intel, the Intel logo, Intel Inside, the Intel Inside logo, Intel Atom, Intel Core, Iris, Movidius, Myriad, Intel Nervana, OpenVINO, Intel Optane, Stratix, and Xeon are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.
- *Other names and brands may be claimed as the property of others.
-

Mar 13, 2020