ORACLE®

Using Deep Learning and Graph Analysis against Cyberattacks

ITOUG TechDay 2018

Hans Viehmann
Product Manager EMEA
ORACLE Corporation
February 1, 2018

@SpatialHannes

ORACLE

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

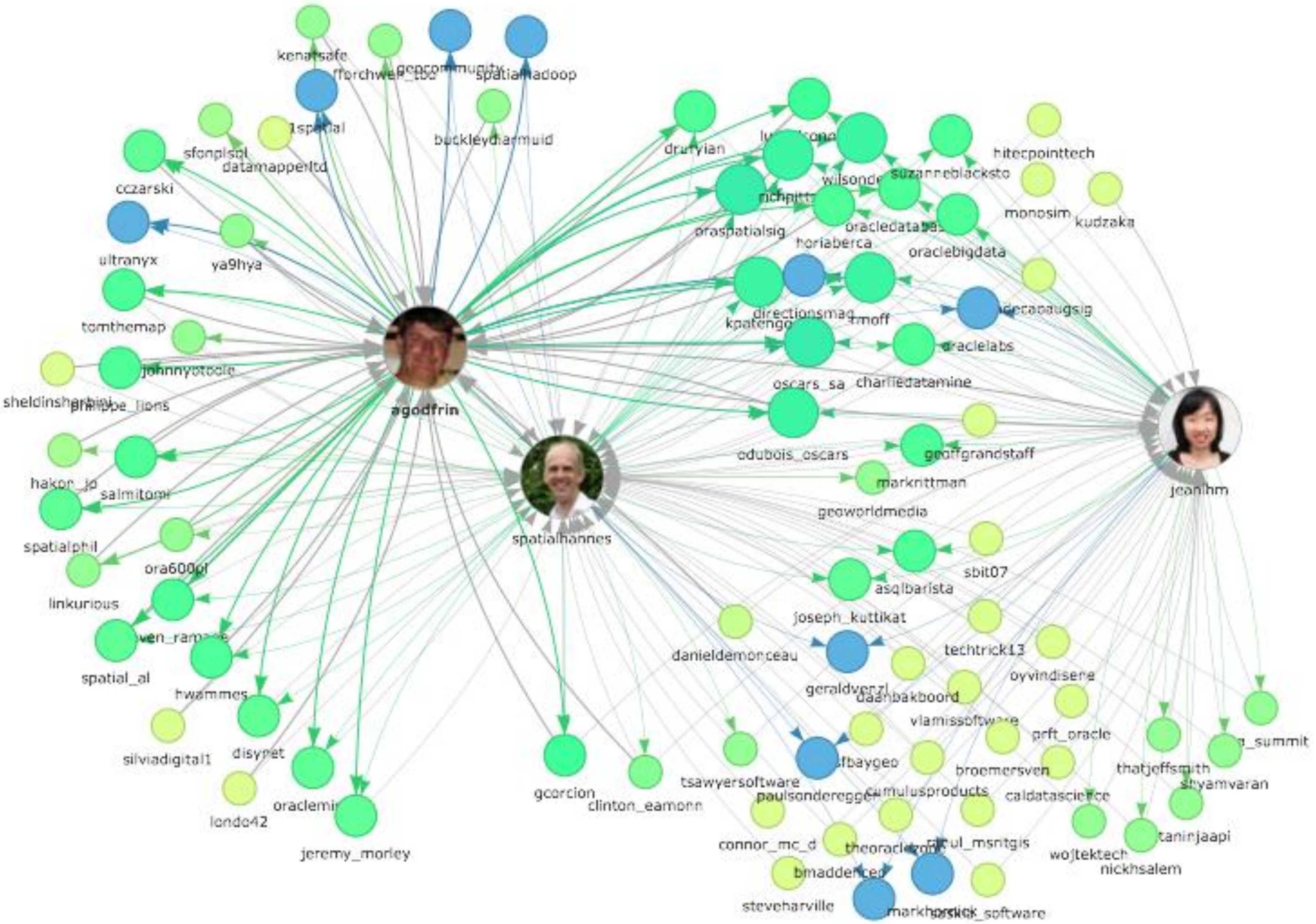
1. Introduction to graph analysis
2. Using Oracle‘s graph technologies to work with graphs
3. Combining graph analysis and machine learning
4. Using machine learning for network intrusion detection
5. Wrap-up

ORACLE

Following, no follow back

Follower, no follow back

Follow each other

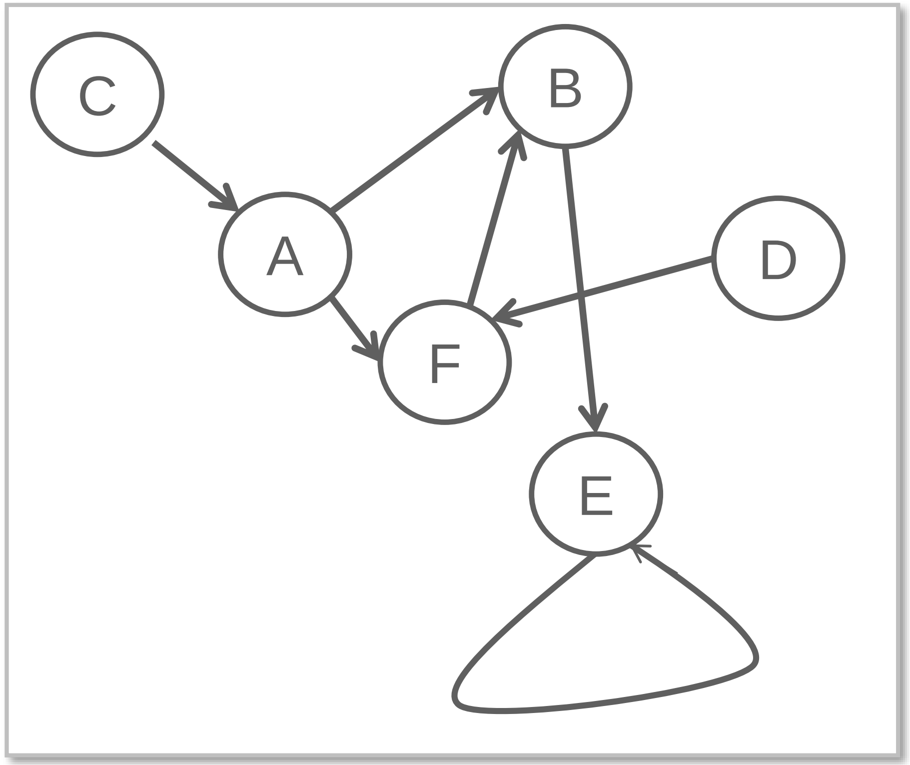


https://twitter.jeffprod.com

ORACLE

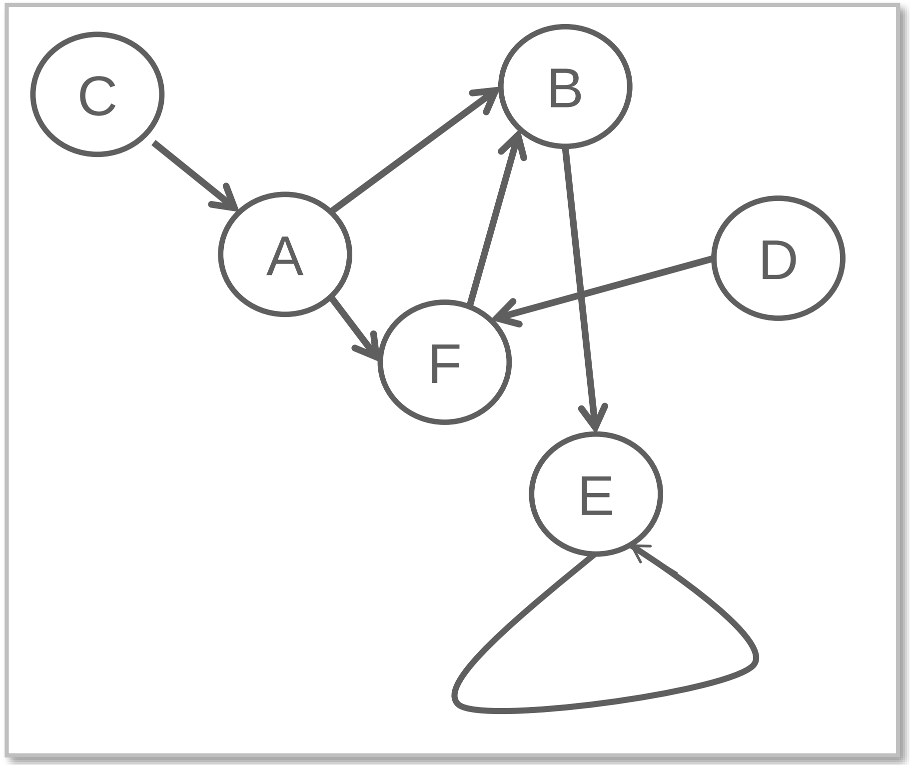
Graph Data Model

- What is a graph?
  - Data model representing entities as vertices and relationships as edges
  - Optionally including attributes
  - Also known as „linked data“
- What are typical graphs?
  - Social Networks
    - LinkedIn, facebook, Google+, ...
  - IP Networks, physical networks, ...
  - Knowledge Graphs
    - Apple SIRI, Google Knowledge Graph, ...



Graph Data Model

- Why are graphs popular?
  - Easy data modeling
    - „whiteboard friendly“
  - Flexible data model
    - No predefined schema, easily extensible
    - Particularly useful for sparse data
  - Insight from graphical representation
    - Intuitive visualization
  - **Enabling new kinds of analysis**
    - Overcoming some limitations in relational technology
    - Basis for Machine Learning (Neural Networks)



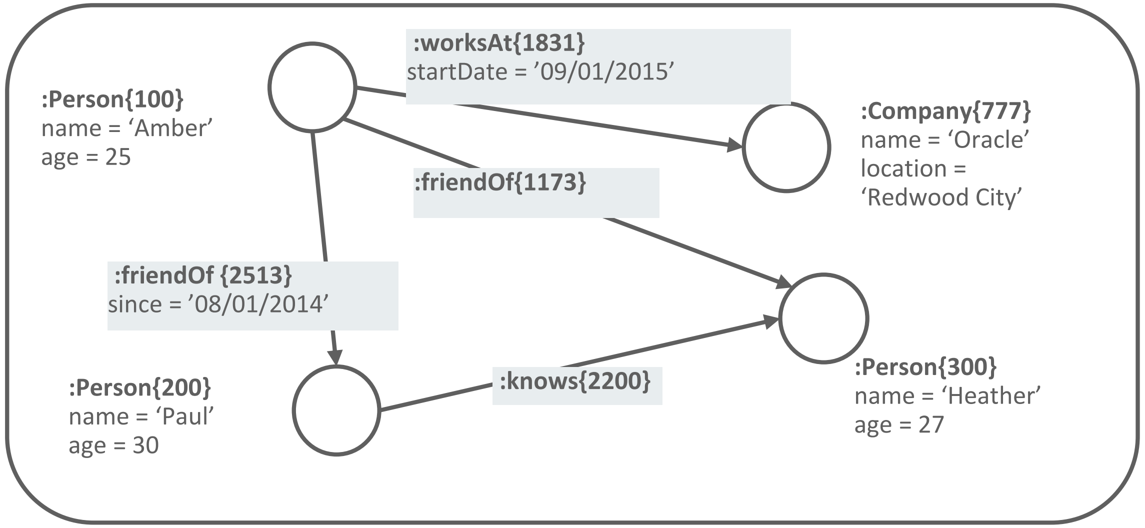
Categories of Graph Analysis

Computational Graph Analytics

- Compute values on vertices and edges
- Traversing graph or iterating over graph (usually repeatedly)
- Procedural logic
- Examples:
  - Shortest Path, PageRank, Weakly Connected Components, Centrality, ...

Graph Pattern Matching

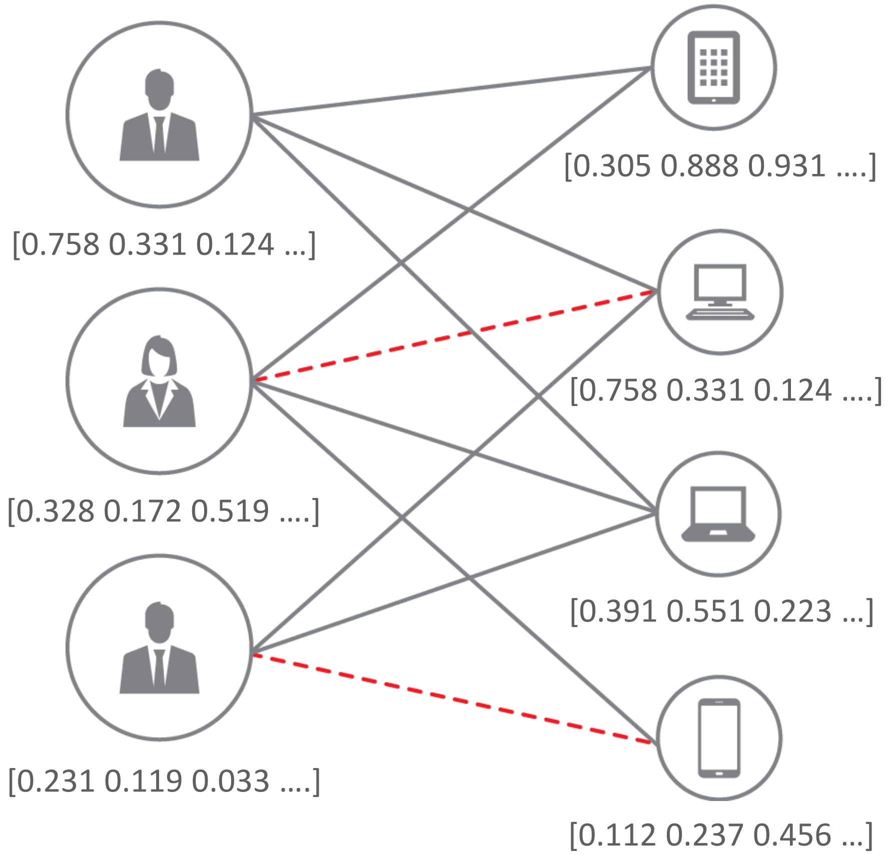
- Based on description of pattern
- Find all matching sub-graphs



ORACLE

Detecting similarities – Recommentation Engines

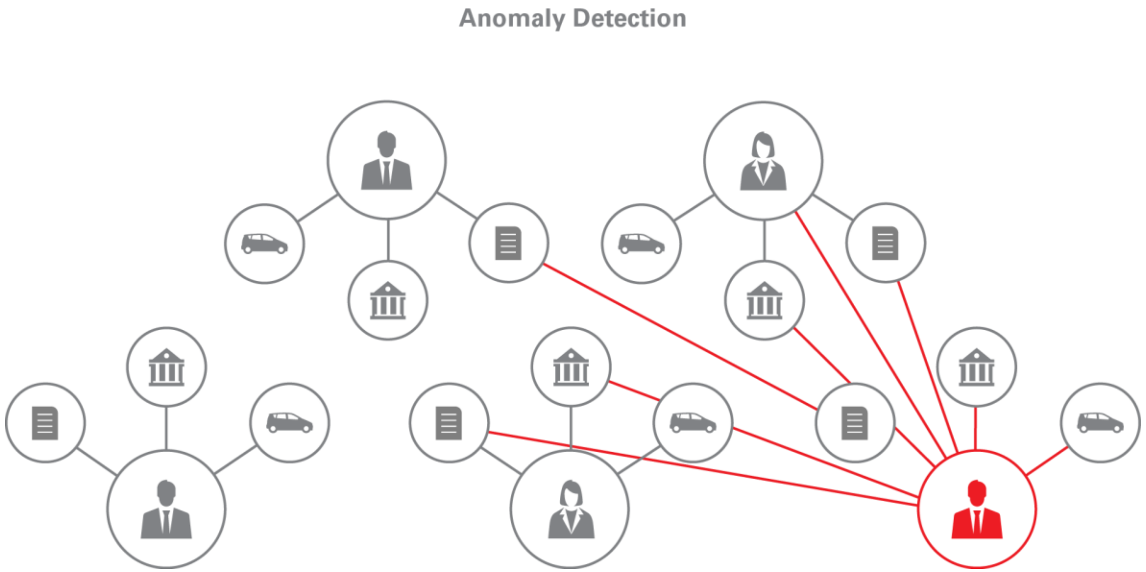
- Identifying users with similar behaviour or buying pattern
- Viewing customer-item relations as large (sparse) matrix
  – Customers as one dimension, items as other
- Matrix cells filled with rating/rank
  – Represent as graph, not as matrix
- Collaborative Filtering [1] algorithm solves taste signature of customers, items
  – Resulting vectors are like DNA
- Inner product of vectors reflects quality of match



[1] https://en.wikipedia.org/wiki/Collaborative_filtering

Detecting Outliers – Graph Analysis and Anomaly Detection

- Requirement:
  - Identify entities from a large dataset that look different than others, especially in their relationships
- Approaches:
  - Define an anomaly pattern, find all instances of the pattern in the graph
  - Given nodes in the same category, find nodes that stand out (eg. low Pagerank value)



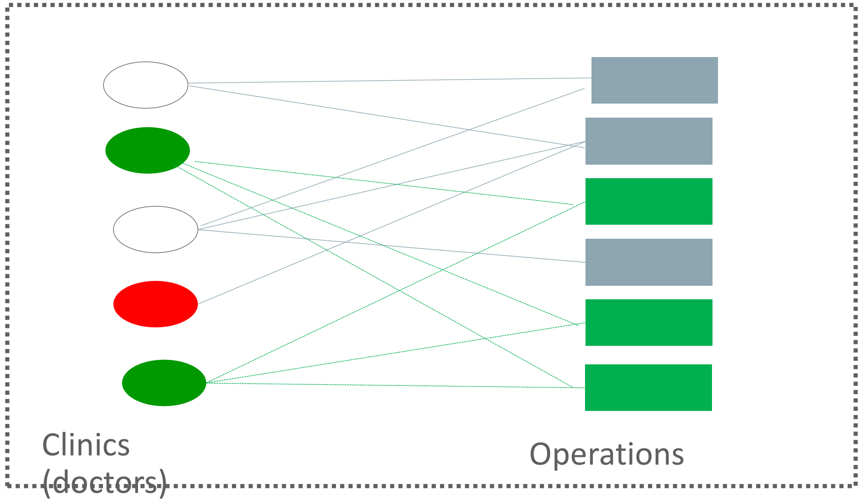
Use case: Fraud Detection in Healthcare

- Example for potential fraud detection
  - Public domain dataset
  - Medical providers and their operations
- Question
  - Are there any medical providers that are suspicious

  ➔ medical providers that perform different operations than their fellows

  (e.g. eye doctors doing plastic surgery)

- Approach
  - Create graph between doctors and operations
  - Apply personalized pagerank (a.k.a equivalent to random walking)
  - Identify doctors that are *far* from their fellows



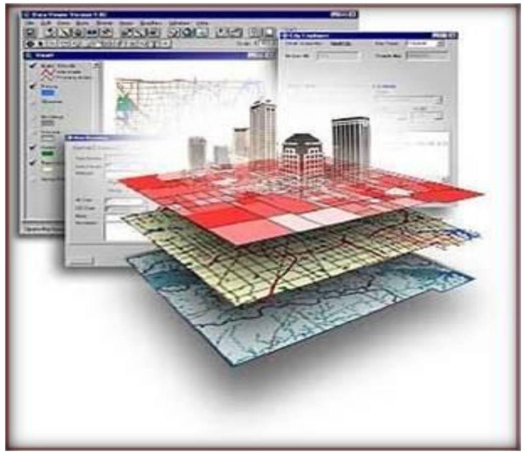
Agenda

1. Introduction to graph analysis
2. Using Oracle‘s graph technologies to work with graphs
3. Combining graph analysis and machine learning
4. Using machine learning for network intrusion detection
5. Wrap-up

ORACLE

Introducing: Oracle Big Data Spatial and Graph

Spatial Analysis:



- Location Data Enrichment
- Proximity and containment analysis, Clustering
- Spatial data preparation (Vector, Raster)
- Interactive visualization

Property Graph Analysis:

- Graph Database
- In-memory Analysis Engine
- Scalable Network Analysis Algorithms
- Developer APIs

In-memory Analytics Engine – Product Packaging

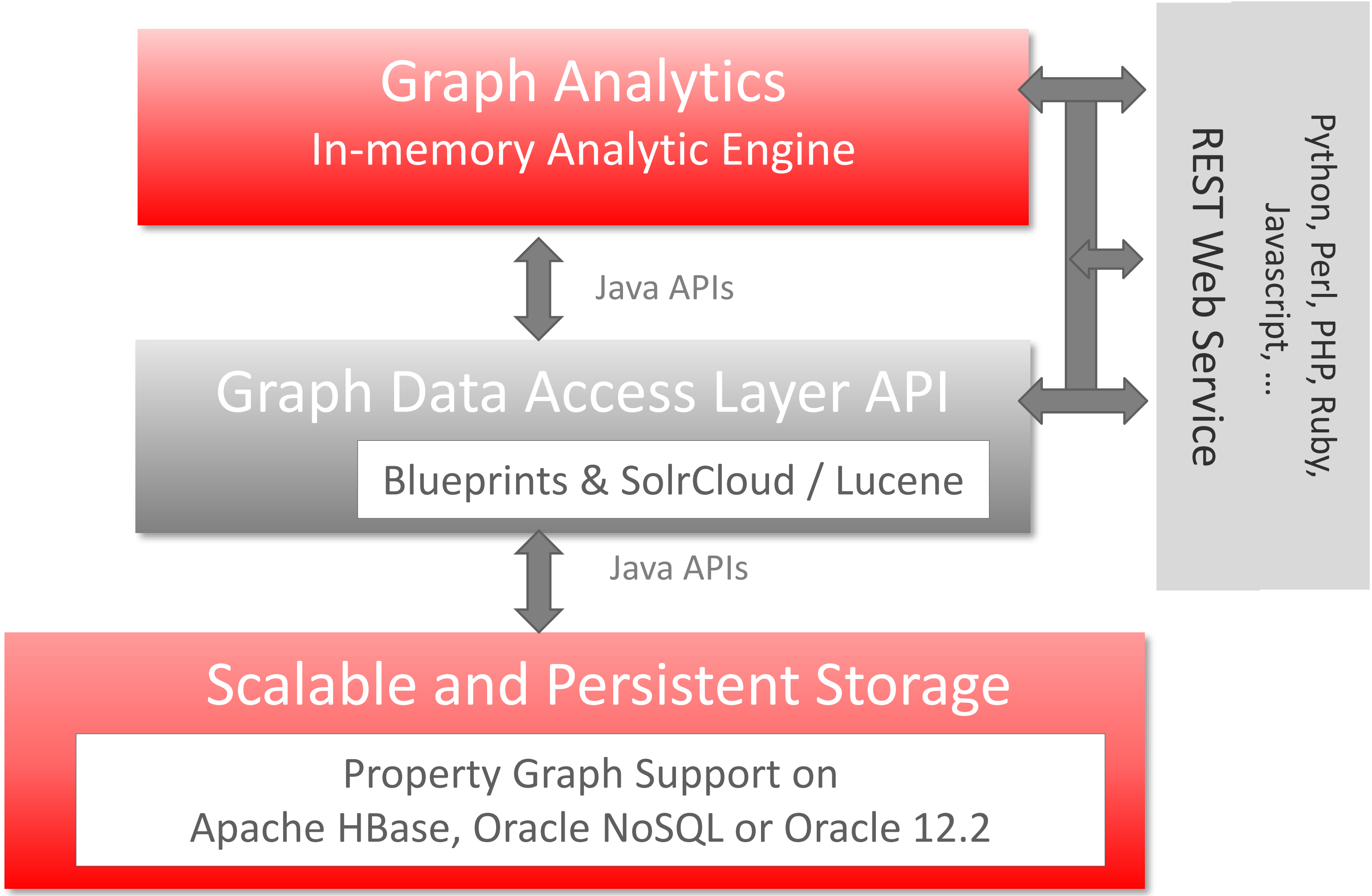
Oracle Big Data Spatial and Graph

- Available for Big Data platform
  - Hadoop, HBase, Oracle NoSQL
- Supported both on BDA and commodity hardware
  - CDH and Hortonworks
- Database connectivity through Big Data Connectors or Big Data SQL
- Included in Big Data Cloud Service

Oracle Spatial and Graph (DB option)

- Available with Oracle 12.2 (EE)
- Using tables for graph persistence
- In-database graph analytics
  - Sparsification, shortest path, page rank, triangle counting, WCC, sub graph generation…
- SQL queries possible
  - Integration with Spatial, Text, Label Security, RDF Views, etc.

Oracle Big Data Graph Architecture



Creating a Graph

- From a relational model
  - Rows in tables usually become vertices
  - Columns become properties on vertices
  - Relationships become edges
  - Join tables in n:m relations are transformed into relationships, columns become properties on edges
- Interactively through API or graphical tool
  - Adding vertices, edges, properties to a given graph
- From graph exchange formats
  - GraphML, GraphSON, GML (Graph Modeling Language)

VIDEO_SALES_ORDERS

SALES_ID	CUST_NAME
10	SMITH
20	JONES
30	TURNER
40	ADAMS

SALES_ORDER_LINE_ITEMS

SALES_ID	LINE_ID	PROD_ID
10	1	1000
10	2	3000
20	1	4000
20	2	3000
20	3	2000
30	1	1000
30	2	1000
40	1	4000

VIDEO_PRODUCTS

PROD_ID	PROD_DESC
1000	TOY STORY
2000	TRUE LIES
3000	POPCORN
4000	STARGATE

ORACLE

Creating a Graph from Network Traffic

- Capture network traffic (source/target IP address and port, protocol, state, duration, ...)
- Model each IP address as vertex
- Model each record (from source IP to destination IP) as an edge
- Attributes can become properties on the edge
- Utilities available to convert CSV to graph
  - OraclePropertyGraphUtilsBase.convertCSV2OPV
  - OraclePropertyGraphUtilsBase.convertCSV2OPE

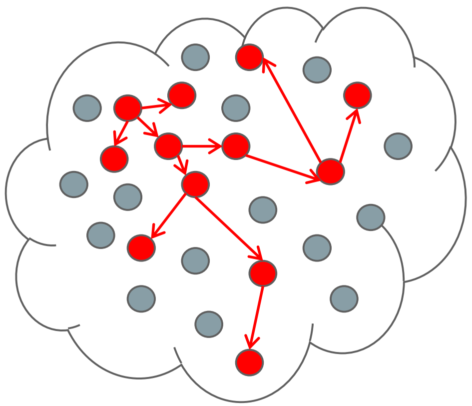
```
59.166.0.1,62377,149.171.126.4,53,udp,CON,0.001044,130,162,31,29,0,0,dns,498084.2813,620689.625,2,
192.168.241.243,259,192.168.241.243,49320,icmp,URH,0,1780,0,64,0,0,0,-,196.4095,0,5,0,0,0,0,0,356,
192.168.241.243,49320,192.168.241.243,0xc0a8,icmp,URH,0,1780,0,64,0,0,0,-,196.4095,0,5,0,0,0,0,0,3
59.166.0.6,38993,149.171.126.0,53,udp,CON,0.00106,132,164,31,29,0,0,dns,498113.1875,618867.875,2,2
59.166.0.9,59720,149.171.126.8,53,udp,CON,0.00107,132,164,31,29,0,0,dns,493457.9375,613084.125,2,2
59.166.0.4,21489,149.171.126.7,53,udp,CON,0.001144,130,162,31,29,0,0,dns,454545.4688,566433.5625,2
```

Agenda

1. Introduction to graph analysis
2. Using Oracle‘s graph technologies to work with graphs
3. Combining graph analysis and machine learning
4. Using machine learning for network intrusion detection
5. Wrap-up

Combining Graph Analytics and Machine Learning

Graph Analytics

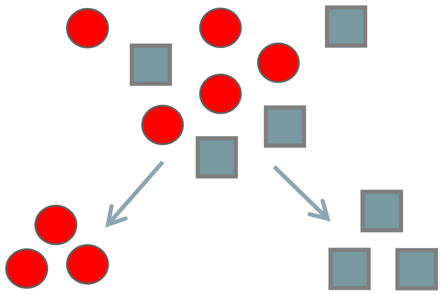


- Compute graph metric(s)
- Explore graph or compute new metrics using ML result

Add to structured data

Add to graph

Machine Learning



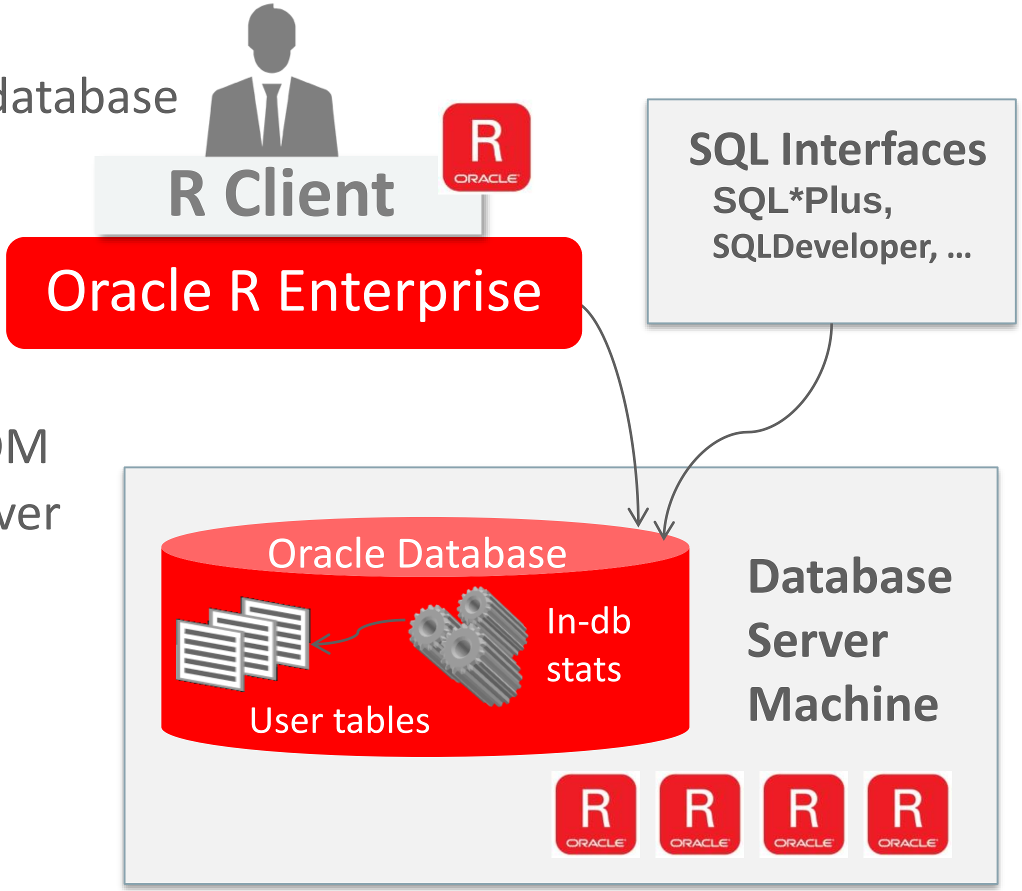
- Build predictive model using graph metric
- Build model(s) and score or classify data

ORACLE

Using Oracle R Enterprise for Machine Learning

Use Oracle Database as a high performance compute environment

- Transparency layer
  - Leverage proxy objects (ore.frames) - data remains in the database
  - Overload R functions that translate functionality to SQL
  - Use standard R syntax to manipulate database data
- Parallel, distributed ML algorithms
  - Scalability and performance
  - Exposes in-database machine learning algorithms from ODM
  - Additional R-based algorithms executing and database server
- Embedded R execution
  - Store and invoke R scripts in Oracle Database
  - Data-parallel, task-parallel, and non-parallel execution
  - Invoke R scripts at Oracle Database server from R or SQL
  - Use open source CRAN packages

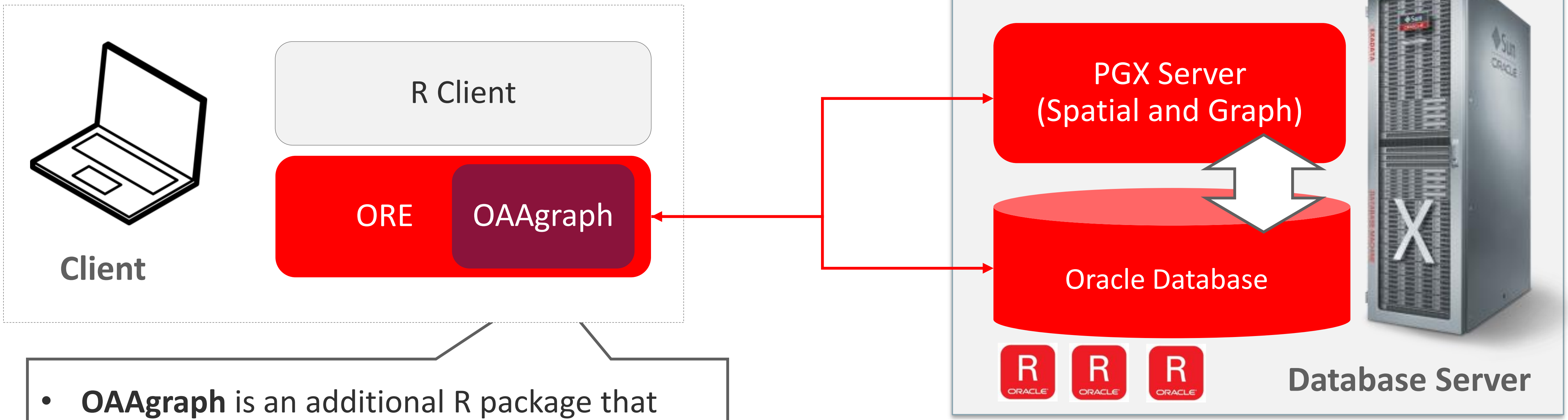


One option: OAAgraph integration with R

- OAAgraph integrates in-memory engine into ORE and ORAAH
- Adds powerful graph analytics and querying capabilities to existing analytical and machine learning portfolio of ORE and ORAAH
- Built-in algorithms of PGX available as R functions
- PGQL pattern matching
- Concept of “cursor” allows browsing of in-memory analytical results using R data structures (R data frame), allows further client-side processing in R
- Exporting data back to Database / Spark allows persistence of results and further processing using existing ORE and ORAAH analytical functions

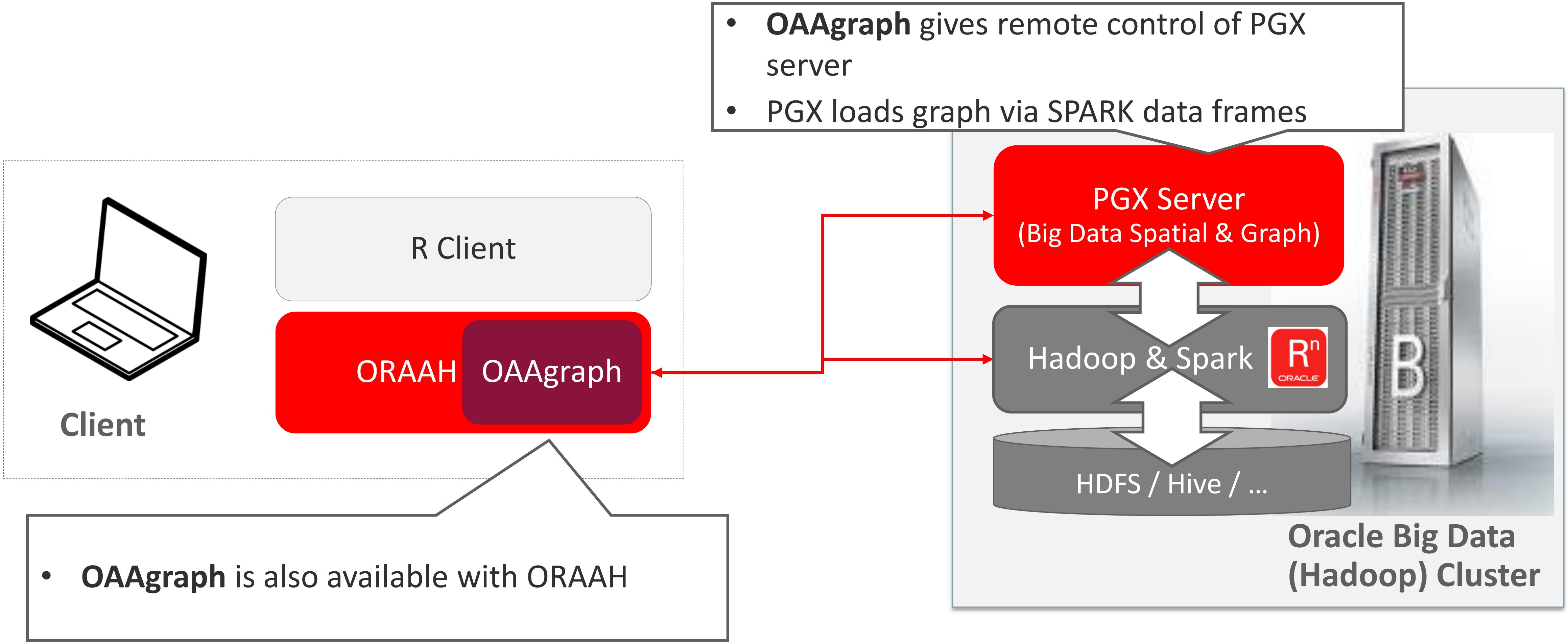
OAAgraph Architecture

- **OAAgraph** gives remote control of PGX server
- PGX loads graph from database (ore.frames)



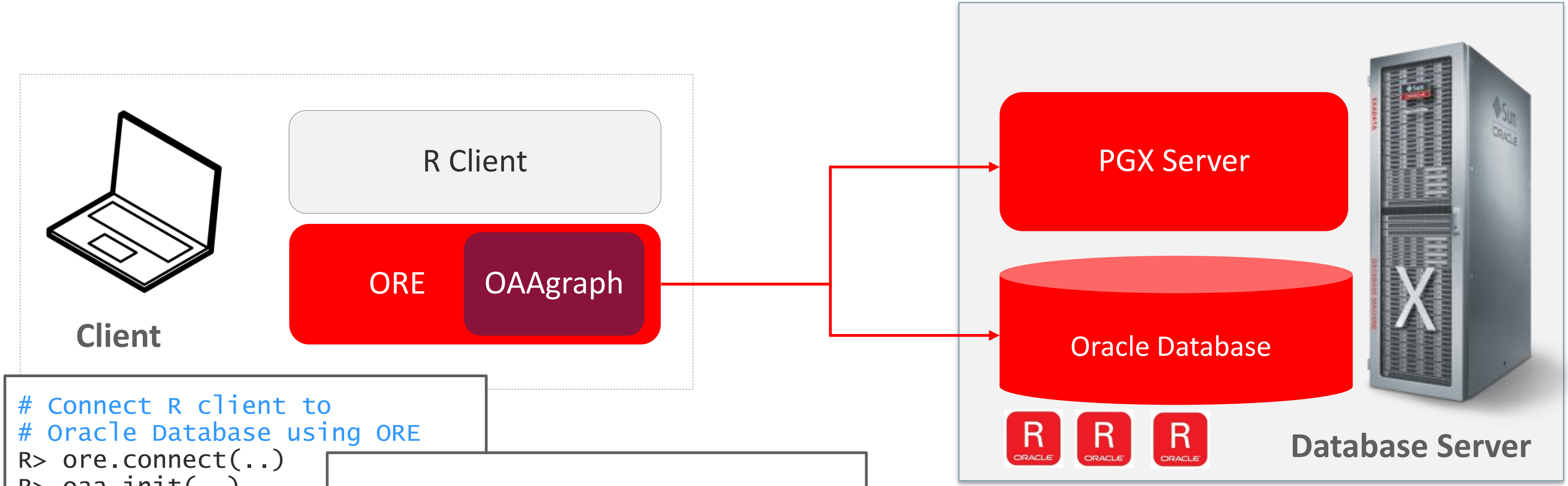
- **OAAgraph** is an additional R package that comes with ORE

OAAgraph Architecture



Execution Overview (ORE)

- Initialization and Connection

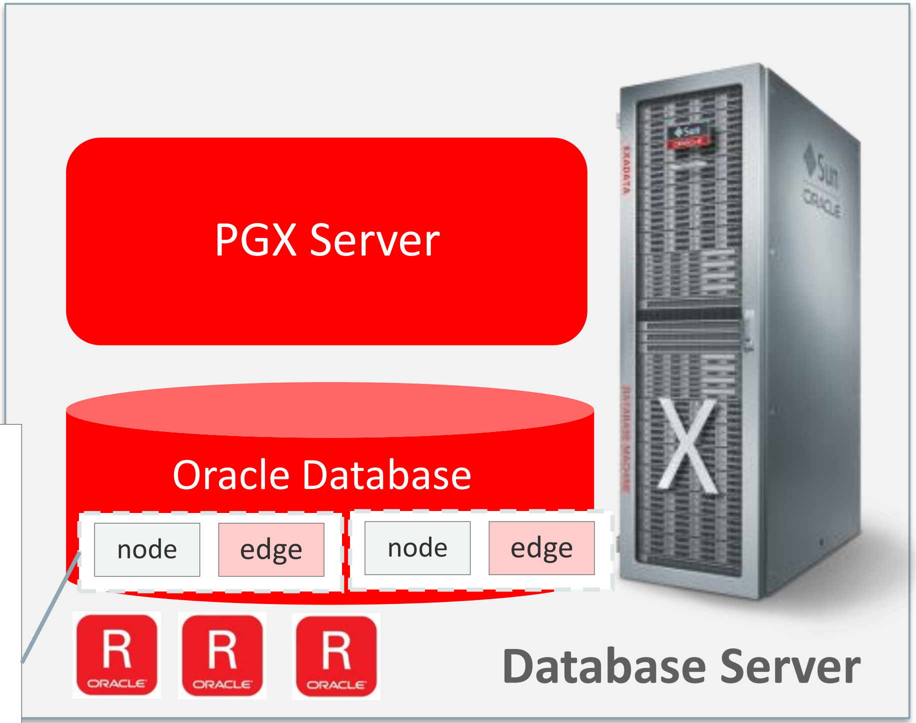



```
# Connect R client to
# Oracle Database using ORE
R> ore.connect(..)
R> oaa.init(..)
```

```
# Connect to PGX server
# using OAAgraph
R> oaa.graphConnect(...)
```

Execution Overview (ORE)

- Data Source
  - Graph data is represented as two tables
    - Nodes and Edges
  - Multiple graphs can be stored in database
    - Using user-specified, unique table names



Node Table

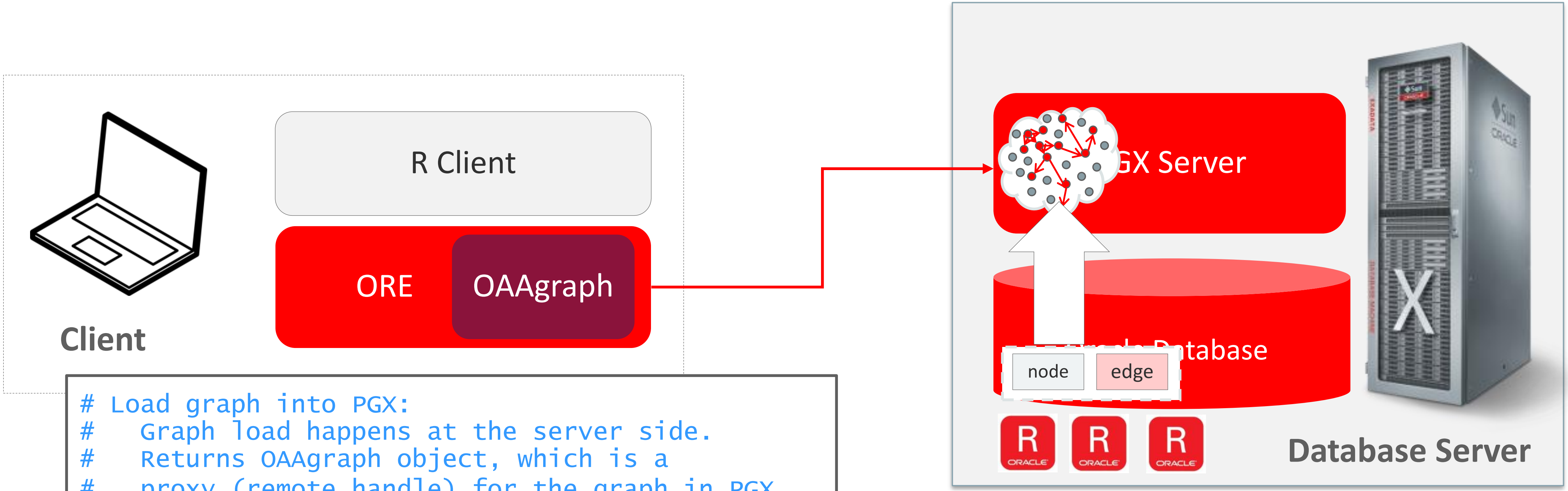
Node ID	Node Prop 1 (name)	Node Prop 2 (age)	…
1238	John	39	…
1299	Paul	41	…
4818	…	…	…

Edge Table

From Node	To Node	Edge Prop 1 (relation)	…
1238	1299	Likes	…
1299	4818	FriendOf	…
1299	6637	FriendOf	…

Execution Overview (ORE)

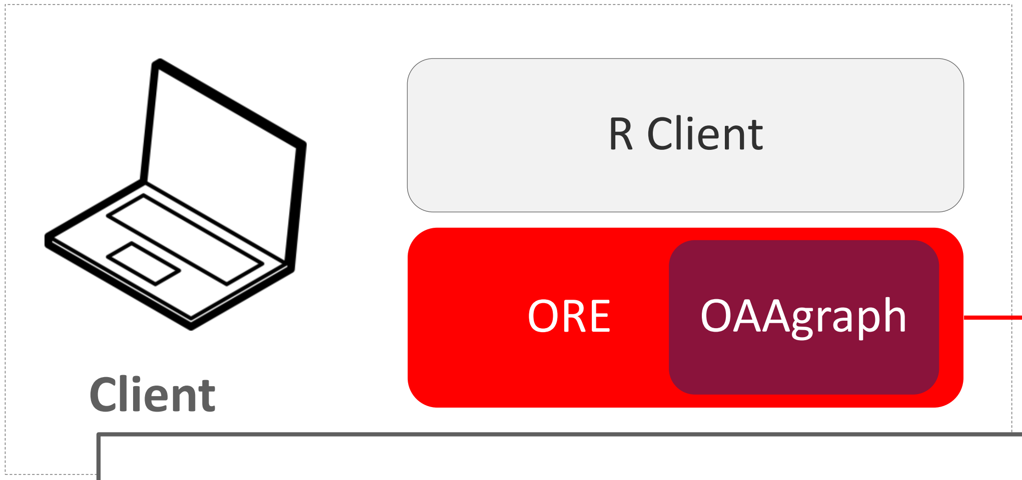
- Loading Graph

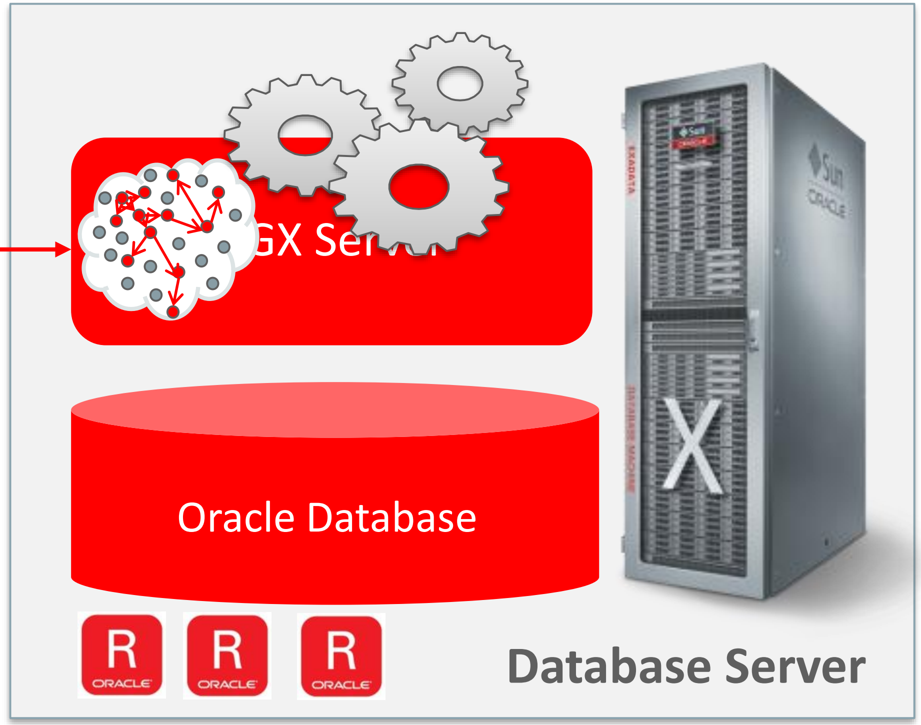



```
# Load graph into PGX:
#   Graph load happens at the server side.
#   Returns OAAgraph object, which is a
#   proxy (remote handle) for the graph in PGX
R> mygraph <-
    oaa.graph (NodeTable, EdgeTable, ...)
```

Execution Overview (ORE)

- Running Graph Algorithm

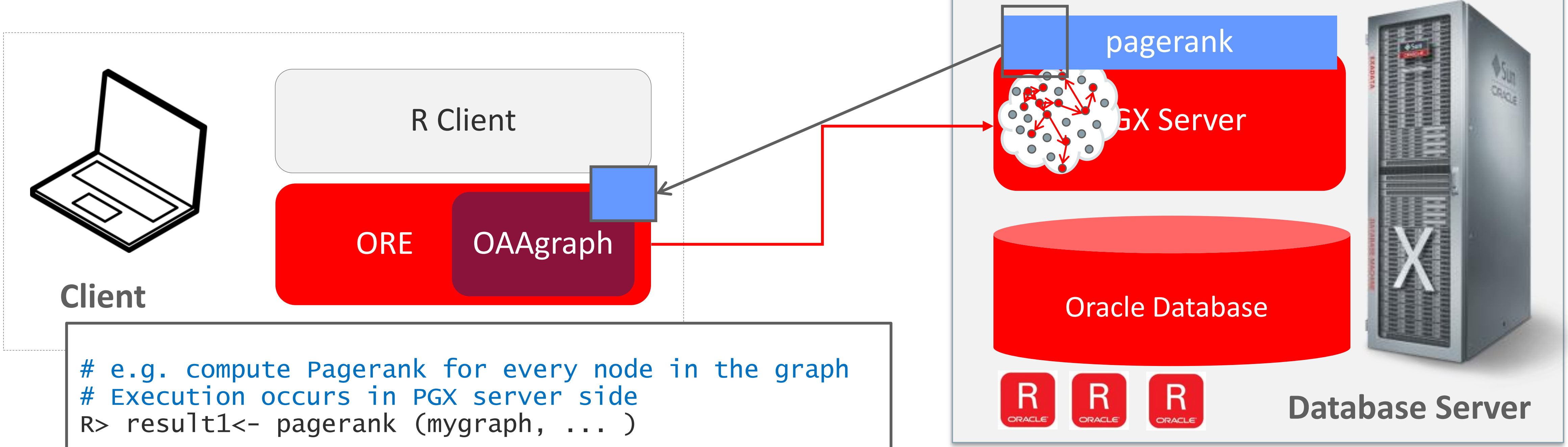





```
# e.g. compute Pagerank for every node in the graph
# Execution occurs in PGX server side
R> result1<- pagerank (mygraph, ... )
```

Execution Overview (ORE)

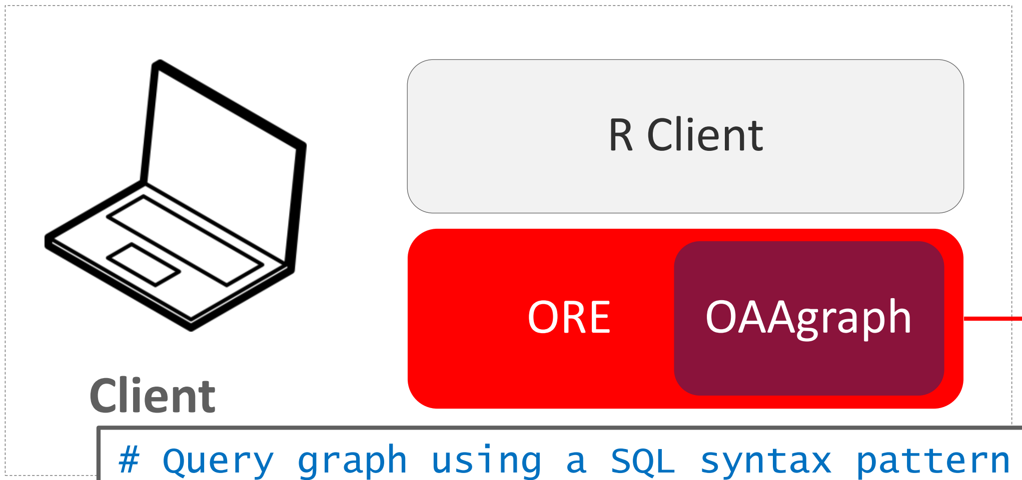
- Iterating remote values with cursor



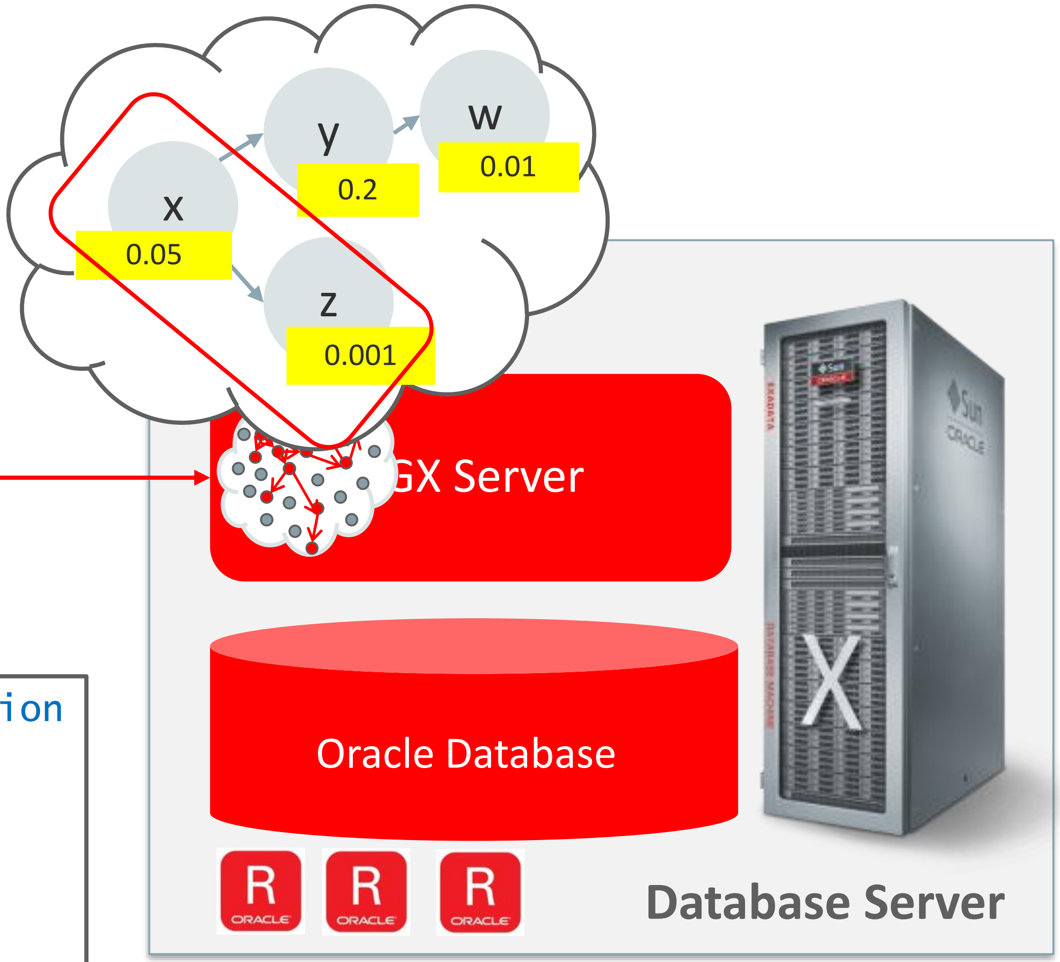

```
# e.g. compute Pagerank for every node in the graph
# Execution occurs in PGX server side
R> result1<- pagerank (mygraph, ... )

# Return value is a “cursor” object
# for the computed result:
#   client can get local data frames by oaa.next()
R> df <- oaa.next(result1, 10)
```

Execution Overview (ORE)

- Querying the graph

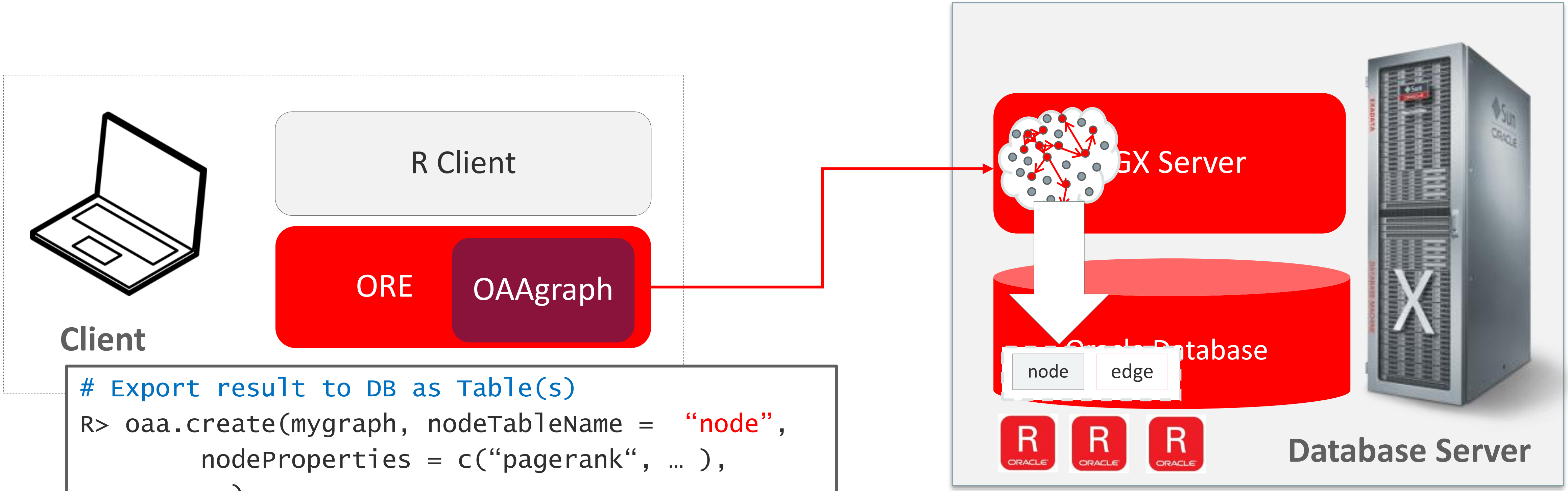



```
# Query graph using a SQL syntax pattern specification
R> q_result <- oaa.cursor(mygraph,
"SELECT n.name, m.name, n.pagerank, m.pagerank
 WHERE (n WITH pagerank < 0.1) -> (m),
       n.pagerank > m.pagerank
 ORDER BY n.pagerank"
)
# Returns a cursor to examine results
R> df <- oaa.next(q_result, 10)
```



Execution Overview (ORE)

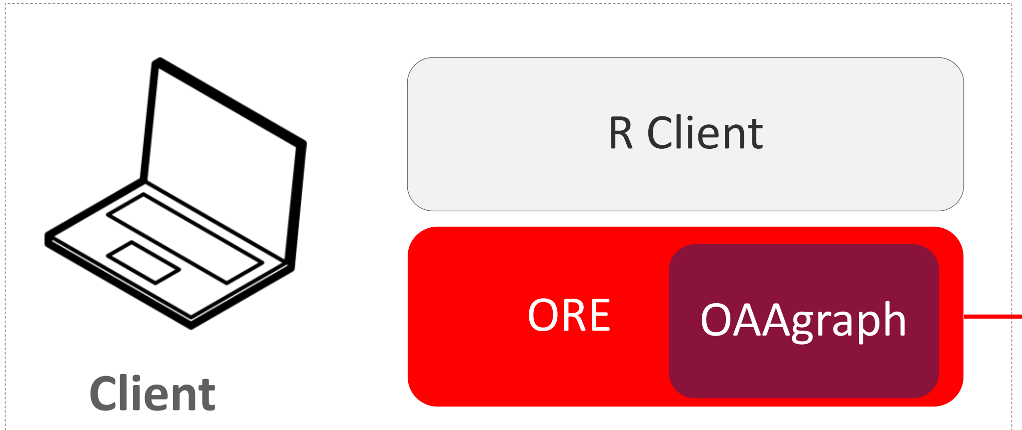
- Exporting the result to DB

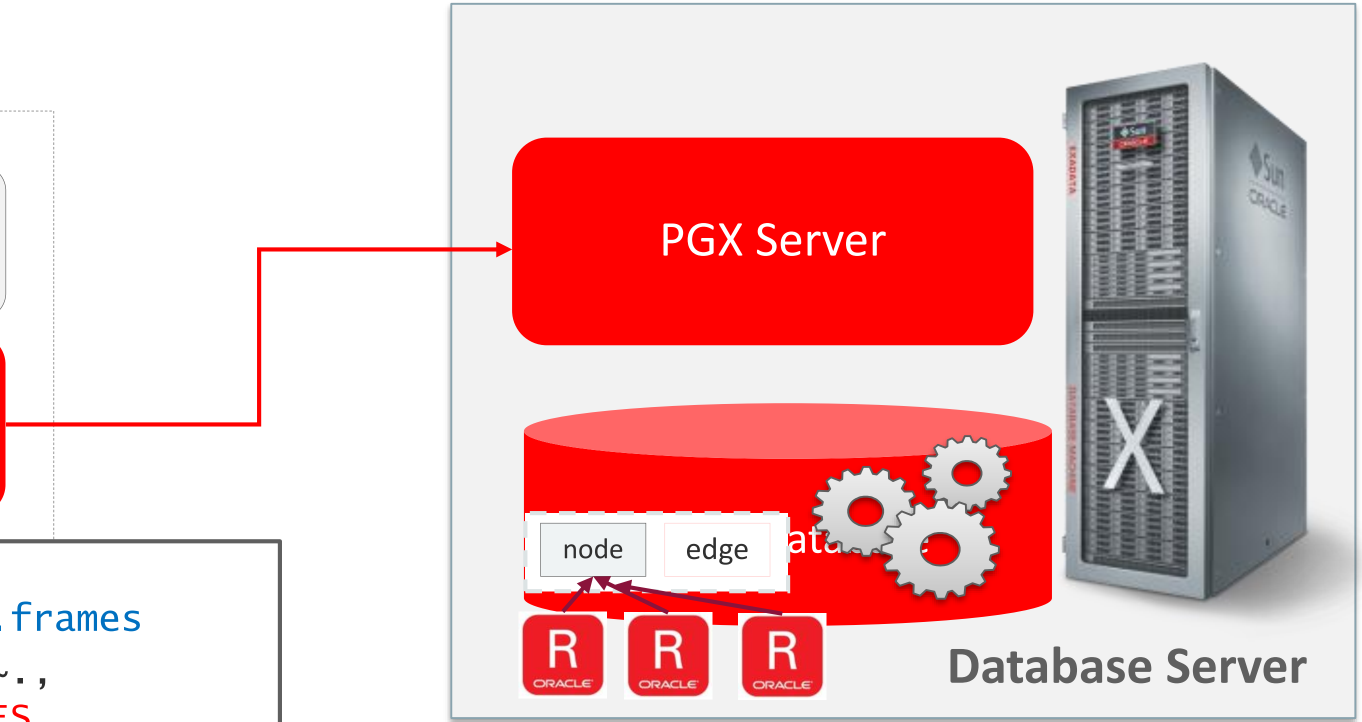



```
# Export result to DB as Table(s)
R> oaa.create(mygraph, nodeTableName =  “node”,
        nodeProperties = c(“pagerank“, … ),
        … )
```

Execution Overview (ORE)

- Continuing analysis with ORE






```
# Use ORE Machine Learning on
# the exported table proxy object ore.frames
R> model <- ore.odmKMmeans(formula = ~.,
                           data = NODES,
                           num.centers = 5,…)
R> scores <- predict(model, NODES, …)
…
```

Agenda

1. Introduction to graph analysis
2. Using Oracle's graph technologies to work with graphs
3. Combining graph analysis and machine learning
4. Using machine learning for network intrusion detection
5. Wrap-up

Use case: Network Intrusion Detection

Using deep learning and graph analysis

- Determining if network activity is legitimate or fraudulent
  - Based on sequence of network activity (as above)
  - Complementary to firewalls, network intrusion prevention system, ...
- Possible approaches
  - „Signature-based“, using labeled dataset of known attacks (supervised learning)
  - Anomaly-based, trying to detect previously unseen attacks
- Most effective systems make use of both
  - Combined with rules engine
- Tested supervised learning in project using DL4J

ORACLE

Supervised learning

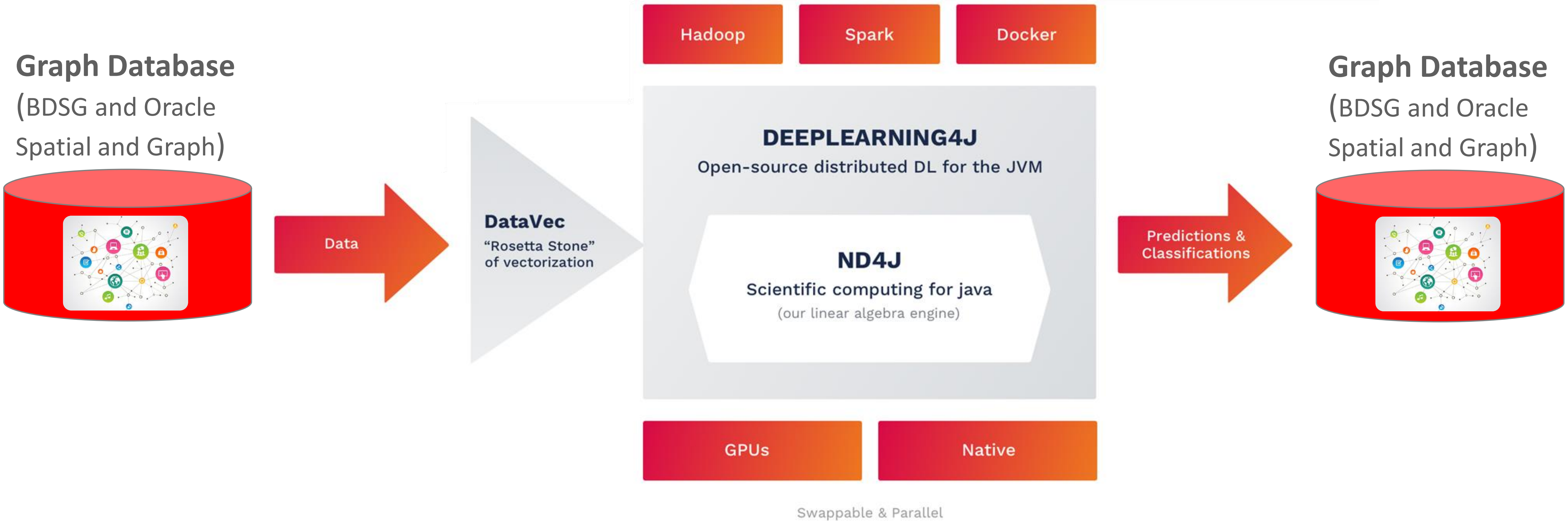
Training dataset

- Labeled Network data set from Univ. of South Wales
  - UNSW-NB15 data set specifically created for **Network Intrusion Detection** systems
  - Generated by IXIA PerfectStorm tool in Cyber Range Lab of Australian Centre for Cyber Security
  - Real modern normal activities plus synthetic contemporary attack behaviours
  - Partitioned into training set (175K records) and testing set (82K records)
  - nine types of attacks – Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms

- Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)."*Military Communications and Information Systems Conference (MilCIS)*, 2015. IEEE, 2015.
- Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set." *Information Security Journal: A Global Perspective* (2016): 1-14.

ORACLE

Prototype with Skymind and DeepLearning4J



Processing Workflow

- Understanding the dataset
  - 49 features in each record – IP addresses, integer, float, timestamp, ...
- Data cleansing
  - Converting Hex to number
- Creating vector as input to DL4J deep learning engine
  - Categorical to One Hot transformation of status strings
- Build Neural Network
  - Train and subsequently test quality using testing set
- Transfer result to graph database
  - Further analysis

Dataset selection | Data Cleansing & preparation | Train Neural Network

- Understand the data
  - Features of

	No.	Name	Type	Description
2	1	srcip	nominal	Source IP address
3	2	sport	integer	Source port number
4	3	dstip	nominal	Destination IP address
5	4	dsport	integer	Destination port number
6	5	proto	nominal	Transaction protocol
7	6	state	nominal	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)
8	7	dur	Float	Record total duration
9	8	sbytes	Integer	Source to destination transaction bytes
10	9	dbytes	Integer	Destination to source transaction bytes
11	10	sttl	Integer	Source to destination time to live value
12	11	dttl	Integer	Destination to source time to live value
13	12	sloss	Integer	Source packets retransmitted or dropped
14	13	dloss	Integer	Destination packets retransmitted or dropped
15	14	service	nominal	http, ftp, smtp, ssh, dns, ftp-data ,irc and (-) if not much used service
16	15	Sload	Float	Source bits per second
17	16	Dload	Float	Destination bits per second
18	17	Spkts	integer	Source to destination packet count
19	18	Dpkts	integer	Destination to source packet count
20	19	swin	integer	Source TCP window advertisement value
21	20	dwin	integer	Destination TCP window advertisement value
22	21	stcpb	integer	Source TCP base sequence number
23	22	dtcpb	integer	Destination TCP base sequence number
24	23	smeansz	integer	Mean of the ?ow packet size transmitted by the src
25	24	dmeansz	integer	Mean of the ?ow packet size transmitted by the dst
26	25	trans_dep	integer	Represents the pipelined depth into the connection of http request/response transaction
27	26	res_bdy_l	integer	Actual uncompressed content size of the data transferred from the server's http service.
28	27	Sjit	Float	Source jitter (mSec)
29	28	Djit	Float	Destination jitter (mSec)
30	29	Stime	Timestam	record start time
31	30	Ltime	Timestam	record last time
32	31	Sintpkt	Float	Source interpacket arrival time (mSec)
33	32	Dintpkt	Float	Destination interpacket arrival time (mSec)
34	33	tcprtt	Float	TCP connection setup round-trip time, the sum of 'synack' and 'ackdat'.
35	34	synack	Float	TCP connection setup time, the time between the SYN and the SYN_ACK packets.
36	35	ackdat	Float	TCP connection setup time, the time between the SYN_ACK and the ACK packets.
37	36	is_sm_ips	Binary	If source (1) and destination (3)IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0
38	37	ct_state_t	Integer	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).
39	38	ct_flw_htt	Integer	No. of flows that has methods such as Get and Post in http service.
40	39	is_ftp_log	Binary	If the ftp session is accessed by user and password then 1 else 0.
41	40	ct_ftp_cm	integer	No of flows that has a command in ftp session.
42	41	ct_srv_src	integer	No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).
43	42	ct_srv_dst	integer	No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).
44	43	ct_dst_ltm	integer	No. of connections of the same destination address (3) in 100 connections according to the last time (26).
45	44	ct_src_ltm	integer	No. of connections of the same source address (1) in 100 connections according to the last time (26).
46	45	ct_src_dpo	integer	No of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26).
47	46	ct_dst_spo	integer	No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26).
48	47	ct_dst_src	integer	No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).
49	48	attack_cat	nominal	The name of each attack category. In this data set , nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms
50	49	Label	binary	0 for normal and 1 for attack records

	A	B	C	D
1	No.	Name	Type	Desc
2	1	srcip	nominal	Sour
3	2	sport	integer	Sour
4	3	dstip	nominal	Desti
5	4	dsport	integer	Desti
6	5	proto	nominal	Trans
7	6	state	nominal	Indic
8	7	dur	Float	Reco
9	8	sbytes	Integer	Sour
10	9	dbytes	Integer	Desti
11	10	sttl	Integer	Sour
12	11	dttl	Integer	Desti
13	12	sloss	Integer	Sour
14	13	dloss	Integer	Desti
15	14	service	nominal	http,
16	15	Sload	Float	Sour
17	16	Dload	Float	Desti
18	17	Spkts	integer	Sour
19	18	Dpkts	integer	Desti

ORACLE

Dataset selection	Data Cleansing & preparation	Train Neural Network model	Generate *Property Graph*	Load Property Graph into BDSG	Graph Visualization

- One round of clean up.
  - Ports should be all integer based, however, there are Hex values
  - Action: convert them back to decimal

```
59.166.0.1,62377,149.171.126.4,53,udp,CON,0.001044,130,162,31,29,0,0,dns,498084.2813,620689.625,2,
192.168.241.243,259,192.168.241.243,49320,icmp,URH,0,1780,0,64,0,0,0,-,196.4095,0,5,0,0,0,0,0,356,
192.168.241.243,49320,192.168.241.243,0xc0a8,icmp,URH,0,1780,0,64,0,0,0,-,196.4095,0,5,0,0,0,0,0,3
59.166.0.6,38993,149.171.126.0,53,udp,CON,0.00106,132,164,31,29,0,0,dns,498113.1875,618867.875,2,2
59.166.0.9,59720,149.171.126.8,53,udp,CON,0.00107,132,164,31,29,0,0,dns,493457.9375,613084.125,2,2
59.166.0.4,21489,149.171.126.7,53,udp,CON,0.001144,130,162,31,29,0,0,dns,454545.4688,566433.5625,2
59.166.0.8,45682,149.171.126.0,53,udp,CON,0.001257,130,162,31,29,0,0,dns,413683.375,515513.125,2,2
59.166.0.8,32958,149.171.126.8,53,udp,CON,0.001124,132,164,31,29,0,0,dns,469750.9063,583629.9375,2
59.166.0.8,55879,149.171.126.3,53,udp,CON,0.001075,146,178,31,29,0,0,dns,543255.8125,662325.5625,2
59.166.0.0,43096,149.171.126.3,53,udp,CON,0.001114,132,164,31,29,0,0,dns,473967.6875,588868.9375,2
59.166.0.2,31439,149.171.126.1,53,udp,CON,0.001088,146,178,31,29,0,0,dns,536764.6875,654411.75,2,2
59.166.0.3,45426,149.171.126.0,53,udp,CON,0.001053,132,164,31,29,0,0,dns,501424.5,622981.9375,2,2,
59.166.0.9,28993,149.171.126.3,53,udp,CON,0.001173,132,164,31,29,0,0,dns,450127.875,559249.8125,2,
```

ORACLE

Dataset selection → **Data Cleansing & preparation** → Train Neural Network model → Generate *Property Graph* → Load Property Graph into BDSG → Graph Visualization

- Understand the data & define transformations

```
.removeColumns("timestamp start", "timestamp end", "source ip", "destination ip", 
      "source TCP base sequence num", "dest TCP base sequence num","attack categor
.filter(new FilterInvalidValues("source port", "destination port")) //Remove example
.transform(new ReplaceEmptyIntegerWithValueTransform("count flow http methods", 0))
.transform(new ReplaceInvalidWithIntegerTransform("count ftp commands", 0)) //Only i
.transform(new ConditionalTransform("is ftp login", 1, 0, "service", Arrays.asList('
.transform(new ReplaceEmptyIntegerWithValueTransform("count flow http methods", 0))
.transform(new StringToCategoricalTransform("service", "-", "dns", "http", "smtp", '
.transform(new MapAllStringsExceptListTransform("transaction protocol", "other", Arr
.transform(new StringToCategoricalTransform("transaction protocol", "unas", "sctp", 
.transform(new MapAllStringsExceptListTransform("state", "other", Arrays.asList("FIN
NT=490469, RST=528, TST=8, ACC=43, REQ=9043, no=7, URH=54})
.transform(new StringToCategoricalTransform("state", "FIN", "CON", "INT", "RST", "RE
.transform(new IntegerToCategoricalTransform("equal ips and ports", Arrays.asList("n
.transform(new IntegerToCategoricalTransform("is ftp login", Arrays.asList("not ftp'
.categoricalToOneHot("is ftp login","equal ips and ports","state","transaction proto
```

Categorical to One Hot transformation

- Service "dns" becomes

0 1 0 0 0 0 0 0 0 0 0 0 0

ORACLE

Dataset selection | Data Cleansing & preparation | Train Neural Network model | Generate *Property Graph* | Load Property Graph into BDSG | Graph Visualization

- Executed transformations with Scala & Apache Spark using Oracle's Big Data stack

```
val stringData = jsc.textFile("/user/oracle/UNSW-complete-all-removedhex.csv");

import org.datavec.spark.transform.AnalyzeSpark;
import org.datavec.spark.transform.SparkTransformExecutor;
import org.datavec.spark.transform.misc.StringToWritablesFunction;

val swf = new StringToWritablesFunction(recordReader);
val parsedInputData = stringData.map(swf)
val processedData = SparkTransformExecutor.execute(parsedInputData, tp);
```

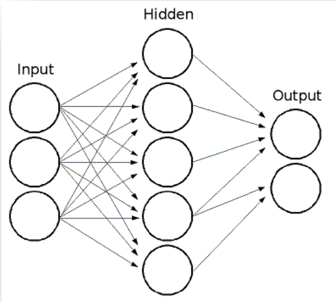
- Save the RDD back to CSV format

ORACLE

Dataset selection | Data Cleansing & preparation | Train Neural Network model | Generate *Property Graph* | Load Property Graph into BDSG | Graph Visualization

- Built a Multi-Layer Perceptron (MLP) Neural Network

```
conf = new NeuralNetConfiguration.Builder()
   .seed(seed)
   .iterations(iIter)
   .activation(Activation.TANH)
   .weightInit(WeightInit.XAVIER)
   .learningRate(learningRate)
   .regularization(true).l2(1e-4)
   .list()
   .layer(0, new DenseLayer.Builder().nIn(numInputs).nOut(iLayer1) .build())
   .layer(1, new DenseLayer.Builder().nIn(iLayer1).nOut(iLayer2) .build())
   .layer(2, new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
       .activation(Activation.SOFTMAX)
       .nIn(iLayer2).nOut(outputNum).build())
   .backprop(true).pretrain(false)
   .build();
```



Dataset selection → Data Cleansing & preparation → Train Neural Network model → Generate *Property Graph* → Load Property Graph into BDSG → Graph Visualization

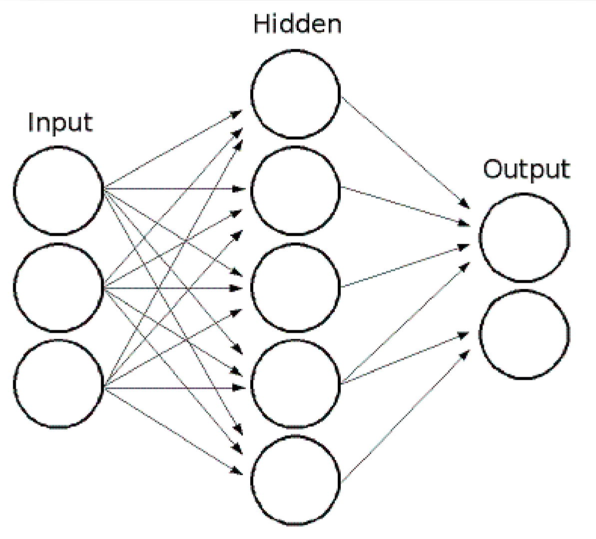
- Tested the quality of MLP NN
  - After 800 iterations of training

    Accuracy: 0.9811

    Precision: 0.9894

    Recall: 0.9286

    F1 Score: 0.958

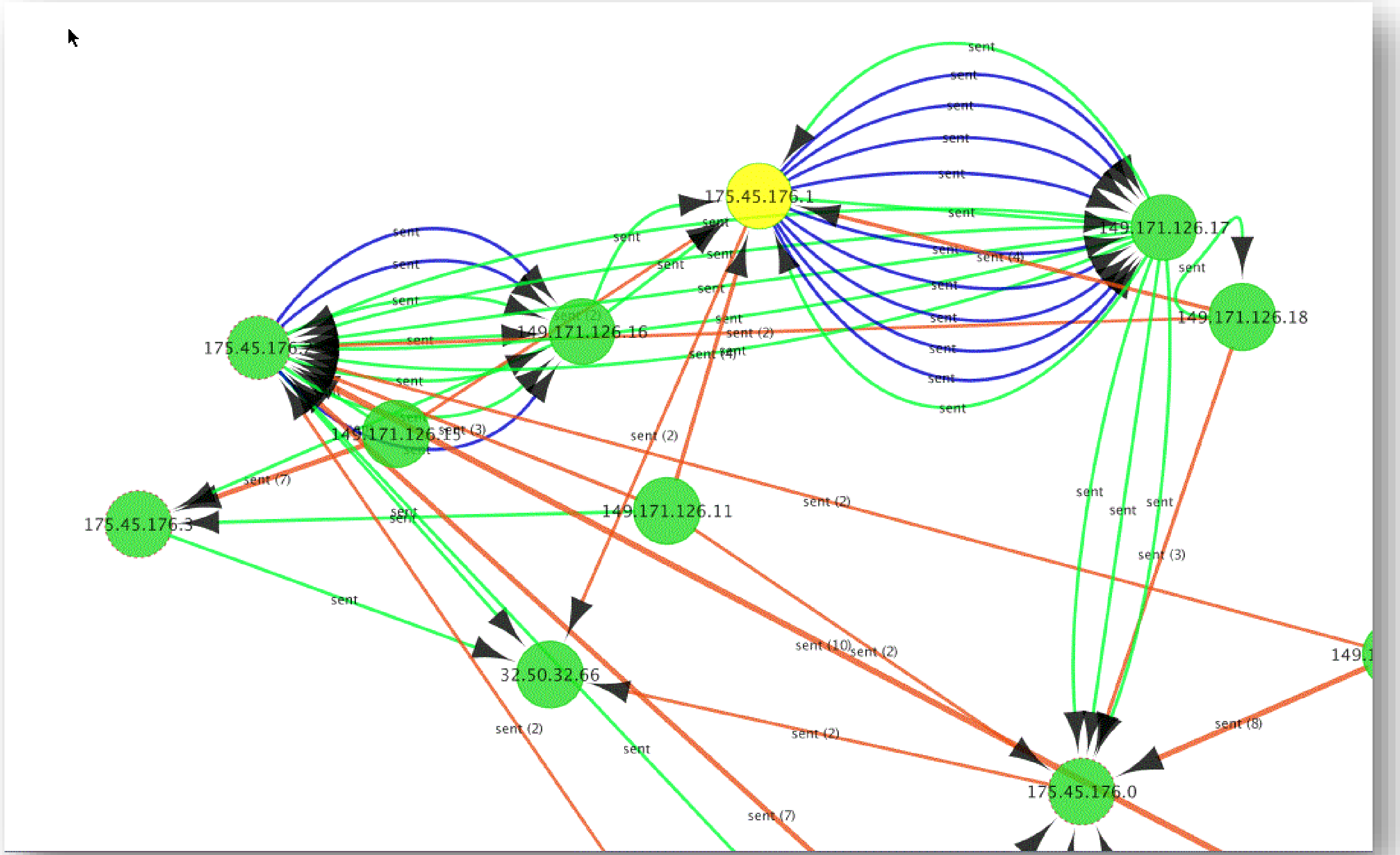


  - Labeled as “non-intrusion” classified as “non-intrusion”: 46 times
  - Labeled as “intrusion” classified as “non-intrusion”: 1 time
  - Labeled as “intrusion” classified as “intrusion”: 6 times ((46+6)/(46+6+1) = 0.9811)
- Long Short-Term Memory (LSTM) NN gave similar F1 result

ORACLE

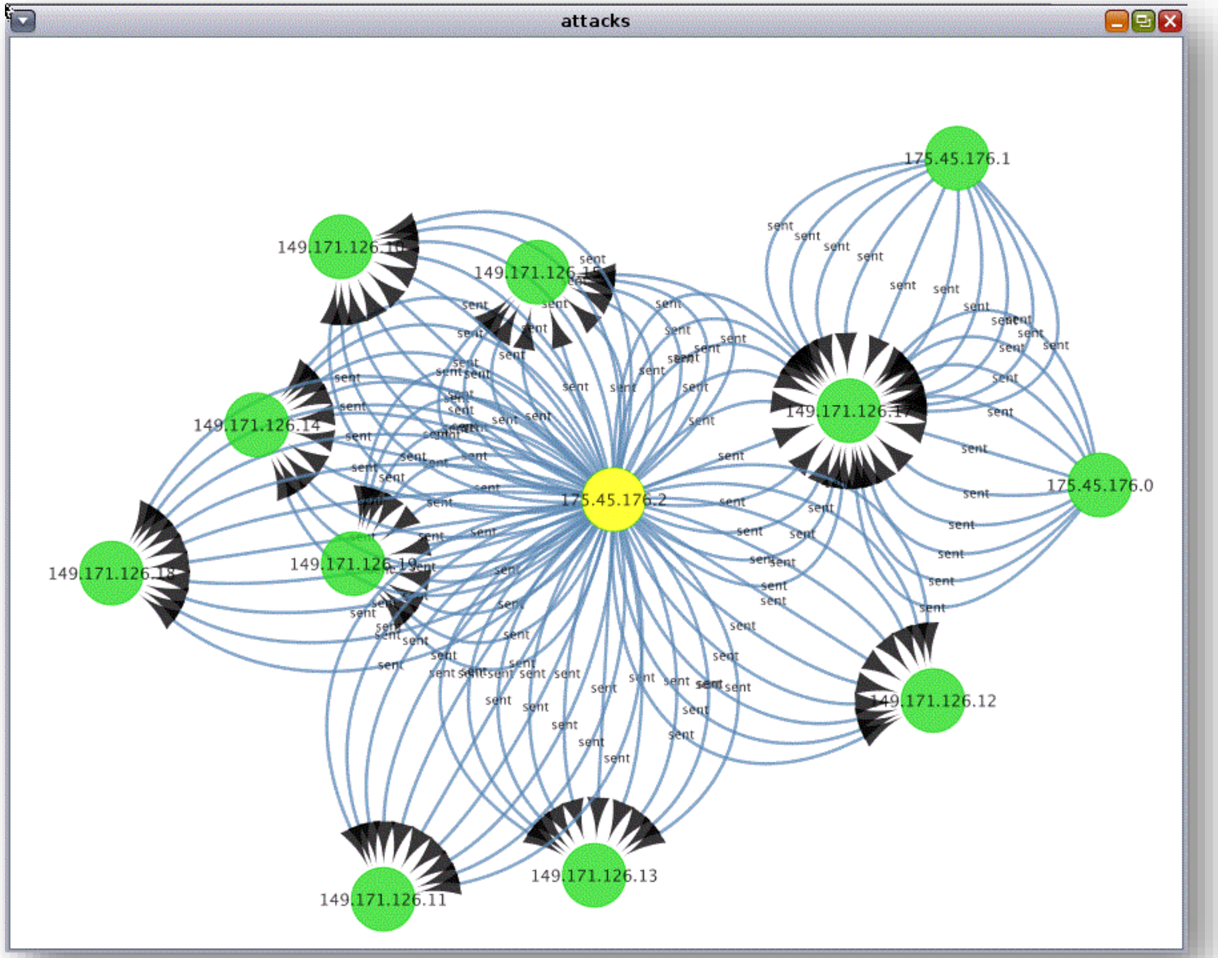
Dataset selection → Data Cleansing & preparation → Train Neural Network model → Generate *Property Graph* → Load Property Graph into BDSG → **Graph Visualization**

- Network Intrusion Detection Property Graph
  - Blue edges: malicious
  - Other edges: normal traffic
- Many attacks originated from 175.45.176.1 to target 149.171.126.17
- Visualization tool: Cytoscape v3.2.1 + Big Data Spatial and Graph v2.1



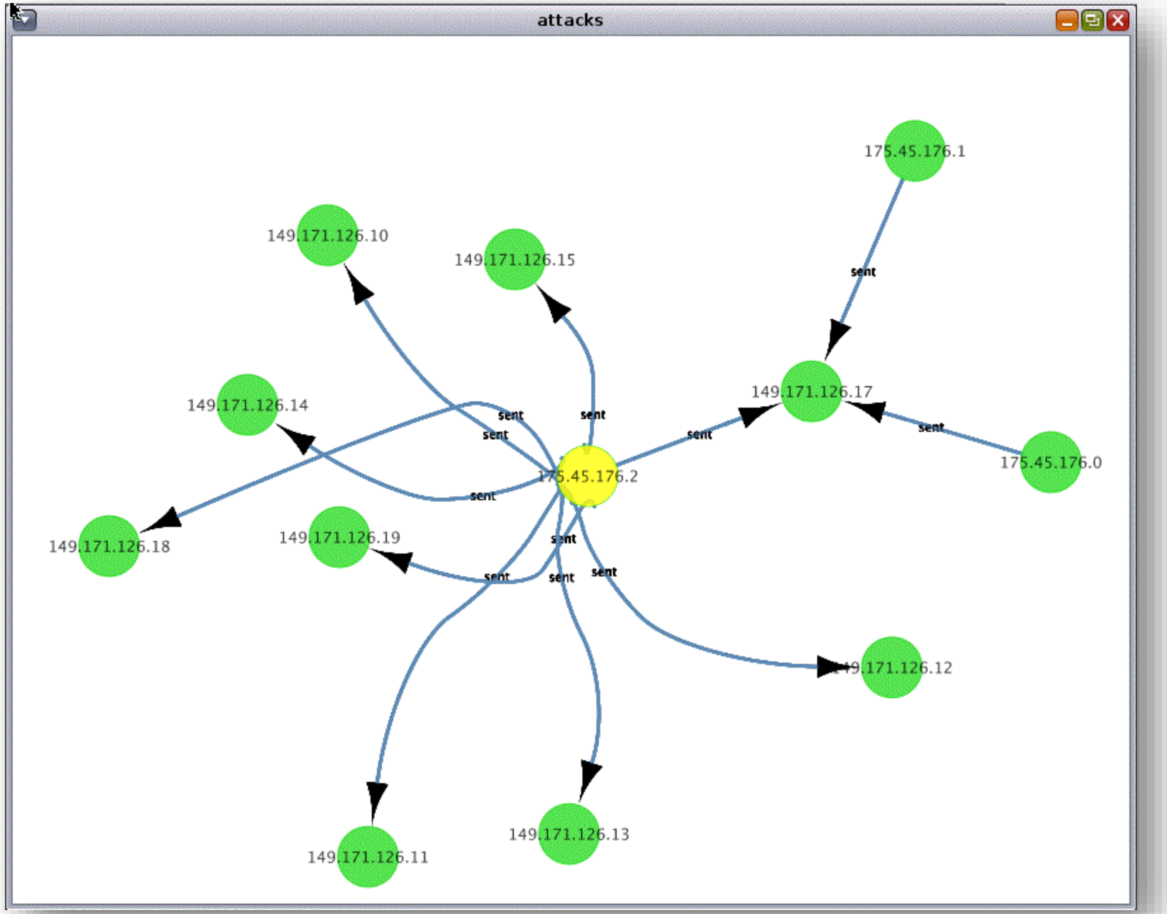
Dataset selection → Data Cleansing & preparation → Train Neural Network model → Generate *Property Graph* → Load Property Graph into BDSG → **Graph Visualization**

- Focused on "Attacks" graph



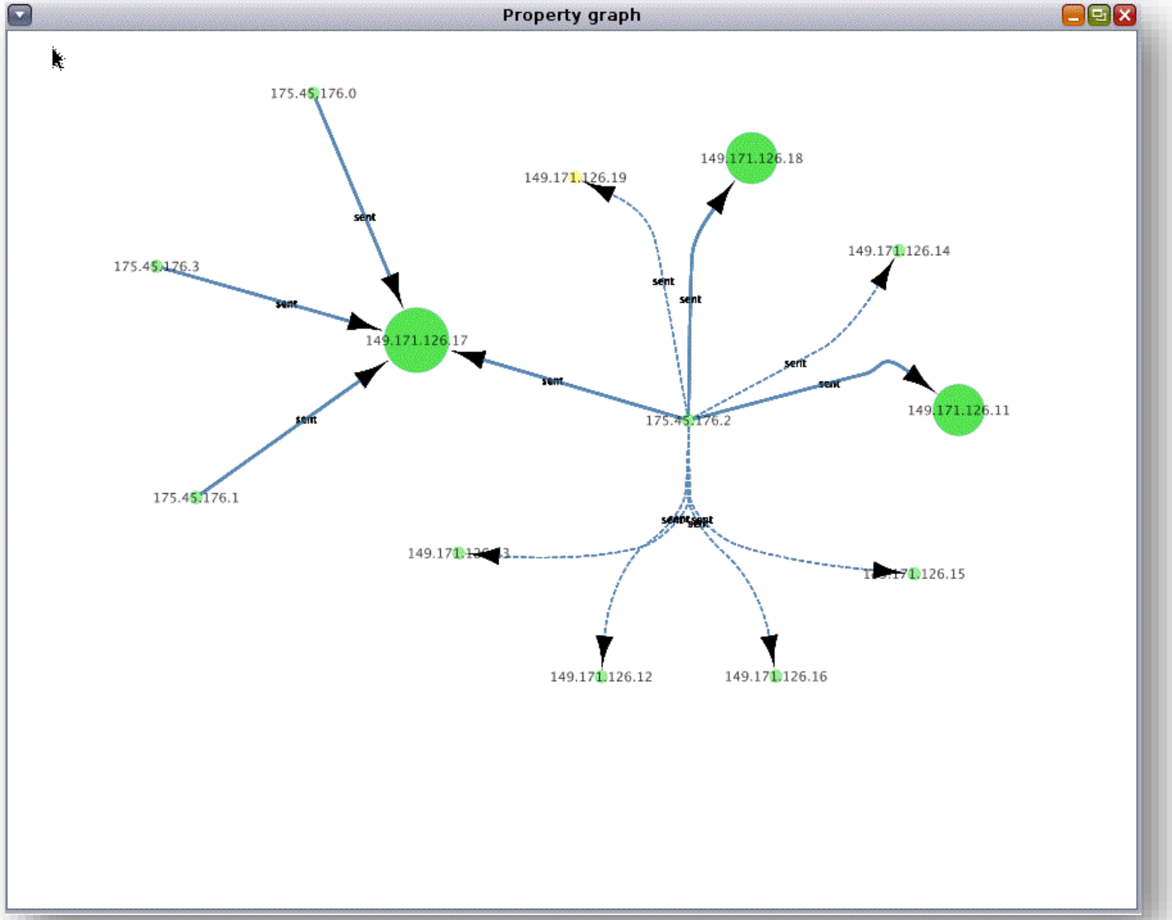
Dataset selection | Data Cleansing & preparation | Train Neural Network model | Generate *Property Graph* | Load Property Graph into BDSG | **Graph Visualization**

- Focused on "Attacks" graph



Dataset selection | Data Cleansing & preparation | Train Neural Network model | Generate *Property Graph* | Load Property Graph into BDSG | Graph Visualization

- Focused on “Attacks” graph
- Applied built-in analytics in BDSG
- Found top-3 IP addresses with **highest Page Rank** value



Wrap-up

Summary

Graph analytics and machine learning

- Graph databases are powerful tools, complementing machine learning technologies
  - Especially strong for analysis of graph topology and multi-hop relationships
- Graph analytics offer new insight which can be used as input to machine learning
  - Especially relationships, dependencies and behavioural patterns
- Oracle Big Data Spatial and Graph and Oracle 12.2 Spatial and Graph offer
  - Comprehensive analytics through various APIs
  - Scaleable, parallel in-memory processing with 40+ graph algorithms pre-built
  - Integration with R, integration with SPARK, integration with relational database
  - Secure and scaleable graph storage on Hadoop using Oracle NoSQL or HBase or Oracle database
- Running both on-premise or in the Cloud

ORACLE

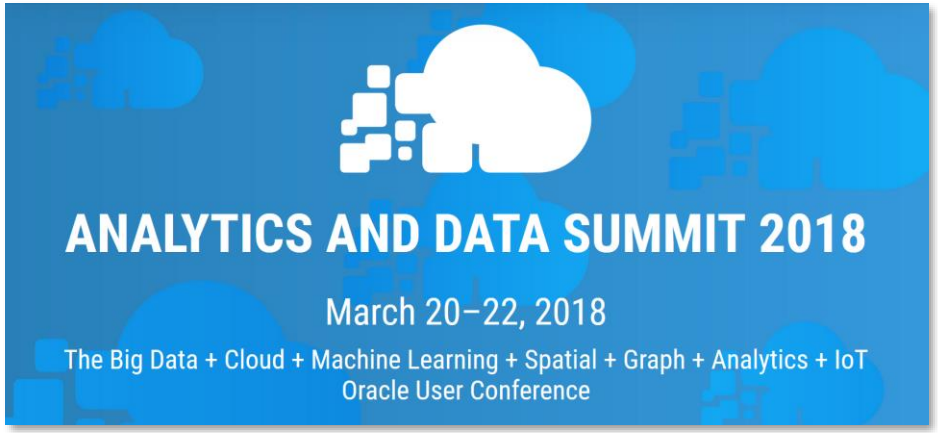
Resources

- Oracle Big Data Spatial and Graph OTN product page: www.oracle.com/technetwork/database/database-technologies/bigdata-spatialandgraph
  - White papers, software downloads, documentation and videos
- Oracle Big Data Lite Virtual Machine - a free sandbox to get started: www.oracle.com/technetwork/database/bigdata-appliance/oracle-bigdatalite-2104726.html
- Hands On Lab included in /opt/oracle/oracle-spatial-graph/
  - Content also available on GITHub under http://github.com/oracle/BigDataLite/
- Blog – examples, tips & tricks: blogs.oracle.com/bigdataspatialgraph
- @OracleBigData, @SpatialHannes, @agodfrin, @JeanIhm
- Oracle Spatial and Graph Group

Interested in project experience, best practices, networking?

Spatial and Graph Summit

- IOUG Business Intelligence, Warehousing and Analytics SIG have established annual BIWA Summit
  - Rebranded as Analytics and Data Summit
  - Planned for March 20 – 22, 2018 at OracleHQ
- Spatial and Graph Summit is separate track
  - Lots of interesting material from previous years available on OTN
- Opportunity for interaction with Spatial PM and Dev‘t team



Q&A

Integrated Cloud

Applications & Platform Services