Setup Resilient ML Research Platform

NOTE: This instruction is to demo MLaaS. Adjustments are required for production deployment. Please use with your judgments.

Setup OS:

Download CentOS 7.0 from http://www.centos.org/download/ or any Linux flavor
Burn DVD if needed. Install Gnome Desktop and LVM (optional)
The scripts below are for CentOS only. Please adjust accordingly if not using CentOS.

Example clusters for demo:
Local 3 node cluster:

xx1.your.com (Hadoop/ Spark master, HDFS name node, Django web server)
xx2.your.com (HDFS 2nd name/data node, Spark worker)
xx3.your.com (HDFS data node, Spark worker)

Note: There is no security setup for Hadoop and Spark clusters in this instruction. If security is required, suggest to separate Django web server from Hadoop and Spark master and only allow Django web to access Hadoop/Spark clusters.

Instruction below is for **ALL** nodes unless specified for specific node.

Setup Prerequisites:

```
# update yum
yum repolist
yum update
```

Add user "hadoop" and install openssh*; this account is used to run Hadoop and Spark.

```
# add user
useradd hadoop
passwd <your pwd>
yum install openssh openssh-clients
```

Generate public key pair for each node and copy public key to other nodes

```
# impersonate hadoop
su hadoop
cd ~
# generate pub key
ssh-keygen -t rsa
# copy key to the file authorized_keys
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
# copy key to other nodes: example for xx1
ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@xx2.your.com
ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@xx3.your.com
# make sure permission is correct
chmod 0600 ~/.ssh/authorized_keys
# verify ssh, should be able to connect without a pwd
ssh localhost
ssh hadoop@xx1.your.com
ssh hadoop@xx2.your.com
ssh hadoop@xx3.your.com
```

<u>Setup Hadoop on master node:</u> example hostname “xx1”
Download Hadoop from http://www.apache.org/dyn/closer.cgi/hadoop/common/ on xx1 node only
Assume Hadoop will be run as user “hadoop”.

```
# un-tar at download folder
tar xzf hadoop-2.n.m.tar.gz
# move to home folder
cp to /home/hadoop/
# make sure files own by hadoop
chown hadoop.hadoop /home/hadoop/* -R
# create folder for data or name nodes based on hdfs-site.xml ===
mkdir /home/hadoop/hadoopdata
mkdir /home/hadoop/hadoopdata/hdfs
mkdir /home/hadoop/hadoopdata/hdfs/datanode
mkdir /home/hadoop/hadoopdata/hdfs/namenode
```

Configure Hadoop .xml files on xx1 node only

```
gedit hadoop-2.n.m/etc/hadoop/core-site.xml    # hdfs master xx1
<configuration>
    <property>
        <name>fs.default.name</name>
        <value>hdfs://xx1.your.com:9000/</value>
    </property>
    <property>
        <name>dfs.permissions</name>
        <value>false</value>
    </property>
</configuration>
# define data folders here at ~/hadoopdata
gedit hadoop-2.n.0/etc/hadoop/hdfs-site.xml #1st namenode: xx1, 2nd: xx2
        <configuration>
                <property>
                        <name>dfs.datanode.data.dir</name>
                        <value>file:/home/hadoop/hadoopdata/hdfs/datanode</value>
                        <final>true</final>
                </property>
                <property>
                        <name>dfs.namenode.name.dir</name>
                        <value>file:/home/hadoop/hadoopdata/hdfs/namenode</value>
                        <final>true</final>
                </property>
                <property>
                        <name>dfs.http.address</name>
                        <value>xx1.your.com:50070</value>
                </property>
                <property>
                        <name>dfs.secondary.http.address</name>
                        <value>xx2.your.com:50090</value>
                </property>
                <property>
                        <name>dfs.replication</name>
                        <value>1</value>
                </property>
        </configuration>
# list datanode
gedit hadoop-2.n.m/etc/hadoop/slaves        # add all data nodes here
xx2.your.com
xx3.your.com
```

Copy Hadoop program folder from xx1 to xx2 and xx3 nodes (for Hadoop master xx1 only)

```
scp -r hadoop-2.m.n xx2.your.com:/home/hadoop
scp -r hadoop-2.m.n xx3.your.com:/home/hadoop
```

Create a symbolic link for $HADOOP_HOME for all nodes

```
# create soft link for Hadoop folder
ln -s ~/hadoop-2.m.n ~/hadoop_latest
```

Create a folder for PID and modify `hadoop-env.sh`

```
# create pid folder
sudo mkdir /var/hadoop
sudo chown hadoop.hadoop /var/hadoop
# modify a variable in hadoop-env.sh
export HADOOP_PID_DIR=/var/hadoop
```

Set Env for Hadoop program and modify `.bashrc` for all nodes

```
#set Env variables for user hadoop's ~/.bashrc on all nodes
#  please edit JAVA home path accordingly
gedit ~/.bashrc
        and add:
export HADOOP_USER_NAME=hadoop
export HADOOP_HOME=/home/hadoop/hadoop_latest
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_HOME=$HADOOP_HOME
export JAVA_HOME=/usr/java/default
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin:$JAVA_HOME/bin :
$HADOOP_HOME/spark_latest/bin
export _JAVA_OPTIONS="-Xmx2g"
export THEANO_FLAGS=mode=FAST_RUN,floatX=float32
# for darknet; path to KML libs etc
source /opt/intel/compilers_and_libraries/linux/bin/compilervars.sh intel64
# source it
source ~/.bashrc
```

Format Hadoop Name Node (for Hadoop master xx1 only)

```
# format Namenode
hdfs namenode -format
```

Open ports for all nodes or disable firewalld if possible

```
# make sure service firewalld stop, disable firewalld if possible
systemctl disable firewalld
systemctl stop firewalld
systemctl status firewalld

        # FYI for production, specified all ports...
        firewall-cmd --state
        # need to find all hadoop ports...
        firewall-cmd --permanent --zone=public --add-port=50070/tcp
        firewall-cmd --permanent --zone=public --add-port=50075/tcp
        etc…
        firewall-cmd --reload
```

Start Hadoop master and verify (for Hadoop master xx1 only)

```
# start Hadoop ====
$HADOOP_HOME/sbin/start-dfs.sh

# verify namenode
http://xx1.your.com:50070/dfshealth.html#tab-overview
# verify cluster
http://xx1.your.com:8088/cluster
# verify 2nd namenode
http://xx2.your.com:50090/status.jsp
# verify datanode
http://xx1.your.com:50075/dataNodeHome.jsp

       # FYI: stop Hadoop  ====
       sbin/stop-dfs.sh
```

Setup Spark on master node “xx1”:

Download and unzip “spark-n.n.n-bin-hadoop2.n.tgz” from https://spark.apache.org/downloads.html
Spark will be run as user “hadoop”

```
# copy unzip Spark program folder to $HADOOP_HOME
# change owner to hadoop for Spark program folder
cd /home/hadoop/spark-n.n.n-bin-hadoopm.m

# modify config files:
gedit conf/spark-env.sh
       # edit below based on your hardware specs (optional)
SPARK_MASTER_IP=xx1.your.com
SPARK_WORKER_CORES=4
SPARK_WORKER_MEMORY=32g
_JAVA_OPTIONS="-Xmx2g"
SPARK_PID_DIR=/var/hadoop

# modify file, conf/slaves for master only
gedit conf/slaves
       # add
xx2.your.com
xx3.your.com

# modify file, conf/log4.properties, to
log4j.rootCategory=ERROR, console

# modify file, conf/spark-defaults.conf, to
spark.driver.maxResultSize  16g
spark.rpc.message.maxSize 1024

# Copy Spark programs to other nodes
scp -r spark-n.n.n-bin-hadoopm.m xx2.your.com:/home/hadoop
scp -r spark-n.n.n-bin-hadoopm.m xx3.your.com:/home/hadoop

# may need to open ports if firewall not disabled

# create spark_latest link
ln -s ~/spark-n.n.n-bin-hadoopm.m ~/spark_latest
```

Start Spark master at xx1

```
# Start Spark master on xx2
/home/hadoop/spark_latest/sbin/start-all.sh
# Verify by
http://xx1.your.com:8080
       # FYI Stop Spark
       sbin/stop-all.sh
```

Install MongoDB:

http://docs.mongodb.org/manual/tutorial/install-mongodb-on-red-hat/

<u>Install software packages for machine learning framework</u>:
Enable yum, pip and wget to work behind proxy

```
# To allow YUM to work with proxy server,
gedit /etc/yum.conf
       # add proxy to /etc/yum.conf:
proxy=http://yourproxy.your.com:1234/

# Also proxy for wget, edit http_proxy=
gedit /etc/wgetrc

# setup EPEL repository ============\
yum install epel-release
yum repolist

# allow pip work behind proxy
export http_proxy=http://yourproxy.your.com:1234
export https_proxy=http://yourproxy.your.com:1234
```

Required packages:

```
# make sure python is 2.7+. by default CentOS 7 has 2.7.5
python -V
# needed package
yum -y install python-setuptools python-setuptools-devel
# install pip
yum -y install python-pip
# packages for python ====
yum install python-argparse
yum -y install gcc gcc-c++ python-devel
yum install blas blas-devel lapack lapack-devel

pip install ujson
pip install numpy --upgrade
pip install scipy --upgrade
pip install distribute
pip install --upgrade setuptools
pip install python-dateutil
pip install pytz
pip install tornado
pip install pyparsing
pip install scikit-learn
yum install libpng-devel
yum install freetype-devel
pip install matplotlib --upgrade

pip install opencv-python
pip install Pillow
yum install python-imaging -y
yum install opencv -y
yum install opencv-devel -y
yum install cmake  -y
yum install gcc gcc-c++ -y
yum install gtk2-devel -y
yum install libdc1394-devel -y
yum install libv4l-devel -y
yum install ffmpeg-devel -y
yum install gstreamer-plugins-base-devel -y
yum install libpng-devel libjpeg-turbo-devel jasper-devel openexr-devel -y

pip install sympy seaborn pyzmq pyxdg pycrypto psutil pickleshare pexpect joblib ipaddr
ecdsa
# jupyter?
pip install nbconvert nbformat jupyter jupyter-client jupyter-console jupyter-core
jsonschema ipython
# may need this version. error from pandas/gtk-2.0? TBD
pip install pandas==0.17.1
yum install graphviz graphviz-devel graphviz-graphs graphviz-python
pip install requests isodate pydot
pip install pymongo
pip install py4j
pip install importlib
yum install boost  boost-devel openssl-devel
```

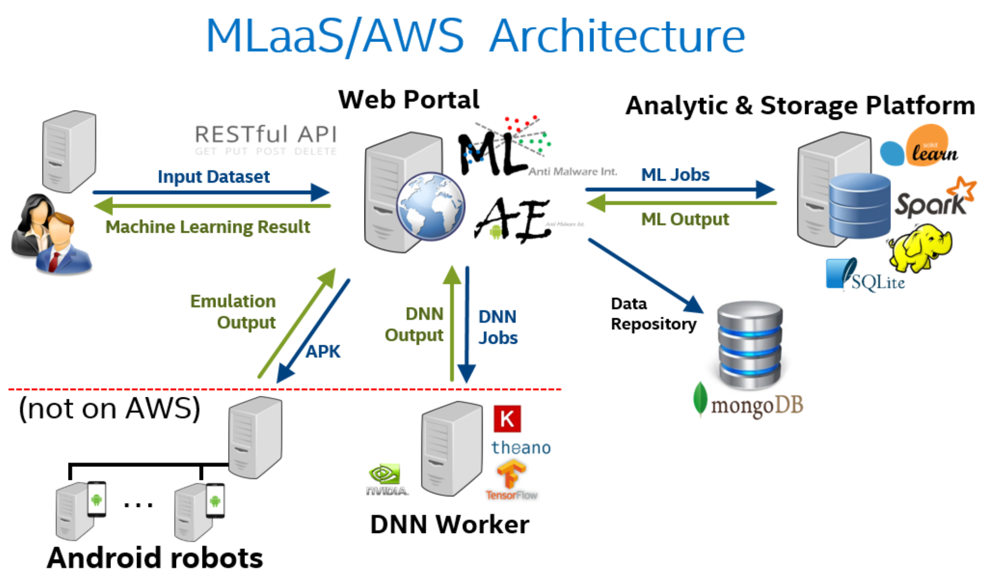
Install Django web framework and database SQLite3 on xx1 only:

Install SQLite before Django and also create user “django” which will be the account to run the web application.

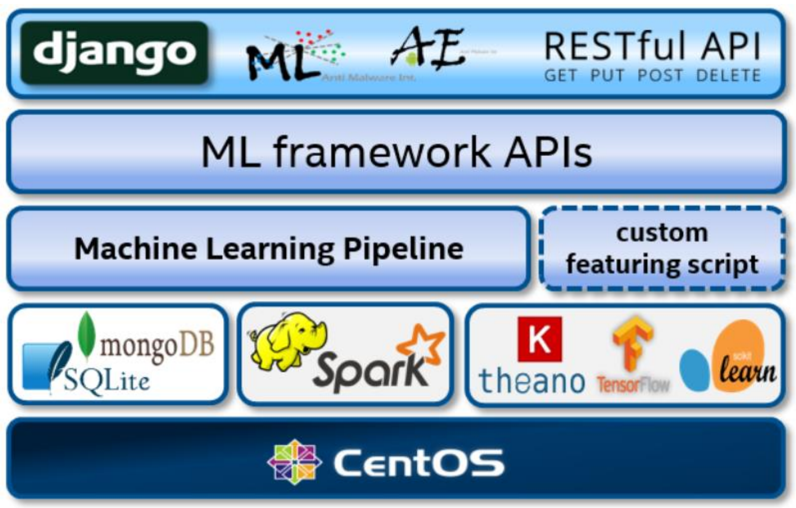
```
# sqlite3 & django
yum install -y zlib-devel openssl-devel sqlite-devel bzip2-devel
pip install Django

# cd to folder /home/django
# django needs to be in “hadoop” group for HDFS access
sudo usermod -a -G  hadoop django
# allow grp to access hadoop folder
chmod 750 /home/hadoop/hadoop-2.x.x # to allow access by group
```

Start a web project and application. Copy web files over

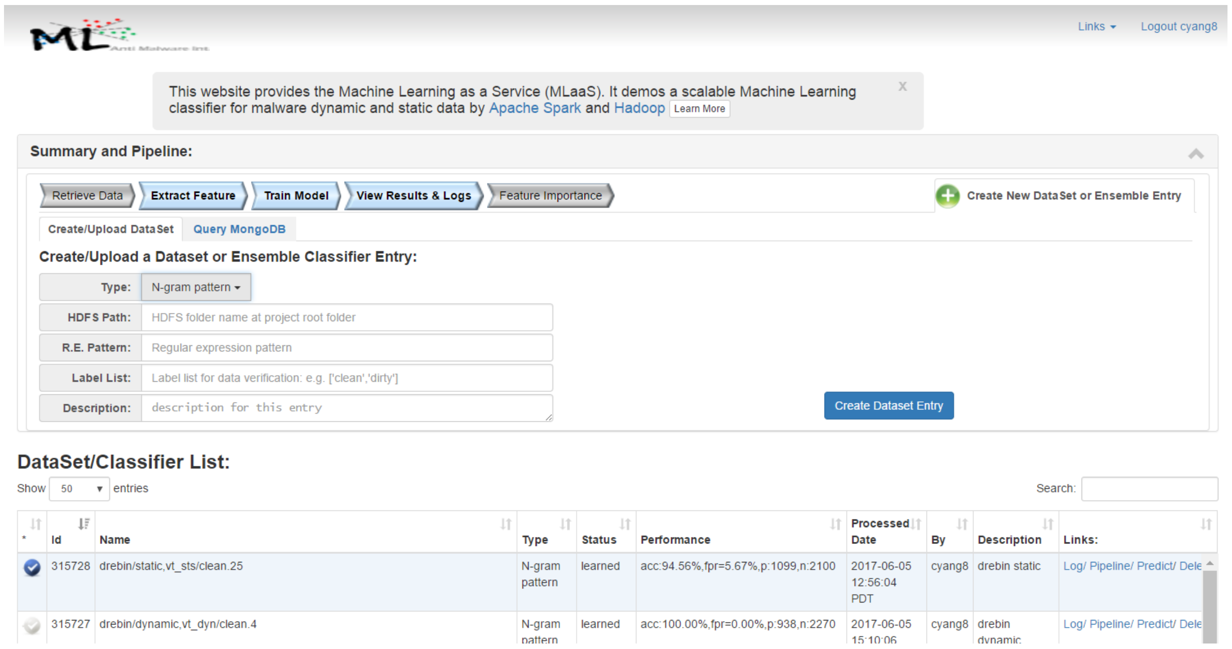
```
# start a web project “myml”
django-admin startproject myml
cd myml

# start an application  “atdml”
python manage.py startapp atdml

# Copy files from Github to folder “myml”
Copy code from a source control site, overwrite existing files

# add entries to user django’s ~/.bashrc; please edit JAVA home path accordingly
export HADOOP_USER_NAME=hadoop
export JAVA_HOME=/usr/java/default

# init the website
python manage.py migrate

# Modify settings.py, atdml/settings.py and app.config for your server names etc.
# search by ”?” to find items needed to be edited

# init the app
python manage.py makemigrations atdml
python manage.py migrate

# create web root admin account
python manage.py createsuperuser

# make sure folders under “media” exists:
upload, tmpdata, result, log etc.

# start web server
python manage.py runserver 0.0.0.0:8000

# add 3 groups from http://xx1.your.com:8000/admin by web root login
1-reader
3-writer
5-developer

# add web users and sign a group
make sure web root has a group assigned

# verify homepage at
http://xx1.your.com:8000/
```



MLaaS Software Stack



Screenshot for Django web



This website provides the Machine Learning as a Service (MLaaS). It demos a scalable Machine Learning classifier for malware dynamic and static data by Apache Spark and Hadoop

Summary and Pipeline:

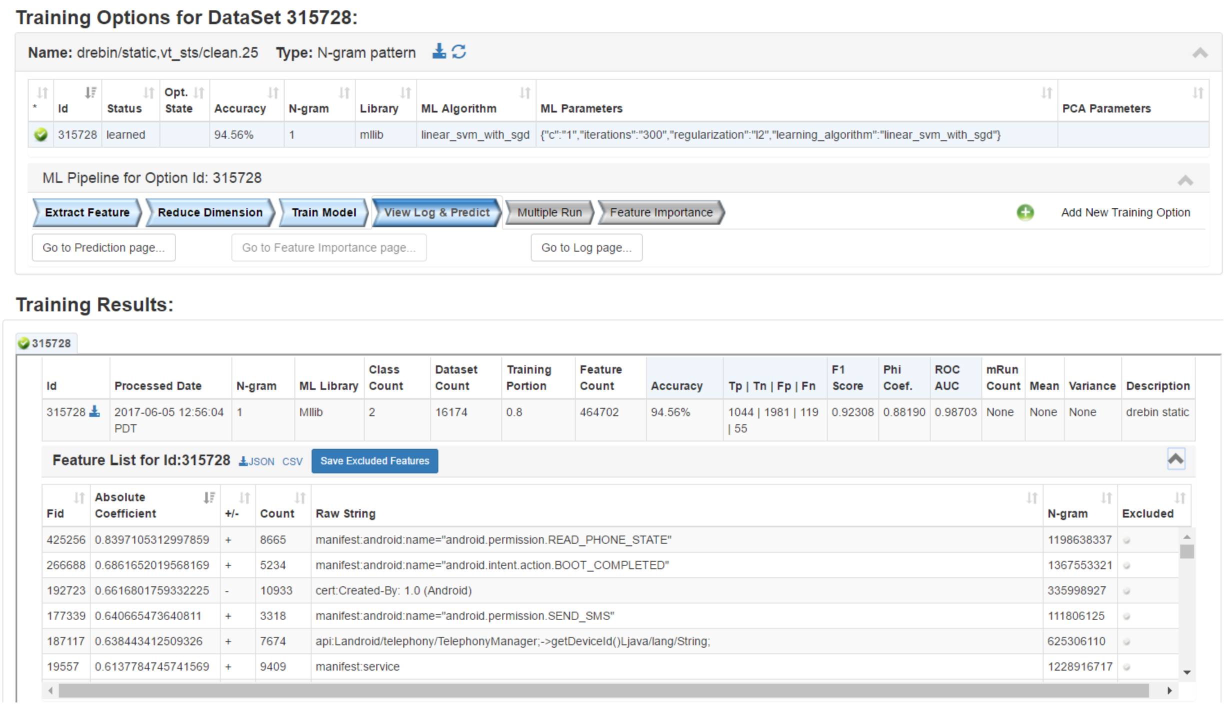
Create/Upload a Dataset or Ensemble Classifier Entry:

Type:	N-gram pattern
HDFS Path:	HDFS folder name at project root folder
R.E. Pattern:	Regular expression pattern
Label List:	Label list for data verification: e.g. ['clean','dirty']
Description:	description for this entry

DataSet/Classifier List:

Show 50 entries   Search:

*	Id	Name	Type	Status	Performance	Processed Date	By	Description	Links:
✓	315728	drebin/static,vt_sts/clean.25	N-gram pattern	learned	acc:94.56%,fpr=5.67%,p:1099,n:2100	2017-06-05 12:56:04 PDT	cyang8	drebin static	Log/ Pipeline/ Predict/ Dele
✓	315727	drebin/dynamic,vt_dyn/clean.4	N-gram pattern	learned	acc:100.00%,fpr=0.00%,p:938,n:2270	2017-06-05 15:10:06	cyang8	drebin dynamic	Log/ Pipeline/ Predict/ Dele



Training Options for DataSet 315728:

Name: drebin/static,vt_sts/clean.25 **Type:** N-gram pattern

*	Id	Status	Opt. State	Accuracy	N-gram	Library	ML Algorithm	ML Parameters	PCA Parameters
✓	315728	learned		94.56%	1	mllib	linear_svm_with_sgd	{"c":"1","iterations":"300","regularization":"l2","learning_algorithm":"linear_svm_with_sgd"}	

ML Pipeline for Option Id: 315728

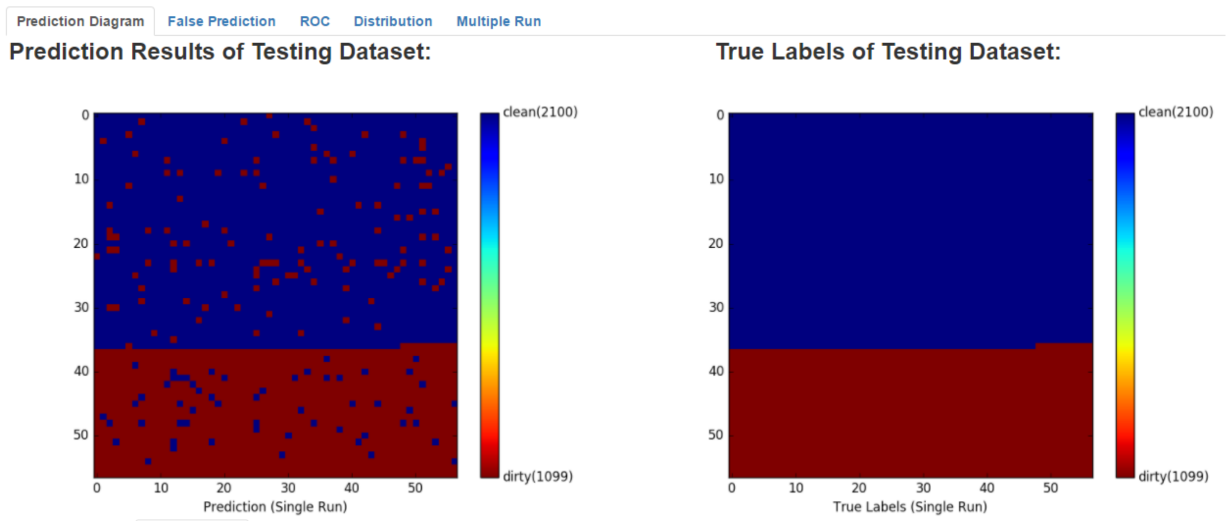
Training Results:

315728

Id	Processed Date	N-gram	ML Library	Class Count	Dataset Count	Training Portion	Feature Count	Accuracy	Tp | Tn | Fp | Fn	F1 Score	Phi Coef.	ROC AUC	mRun Count	Mean	Variance	Description
315728	2017-06-05 12:56:04 PDT	1	Mllib	2	16174	0.8	464702	94.56%	1044 | 1981 | 119 | 55	0.92308	0.88190	0.98703	None	None	None	drebin static

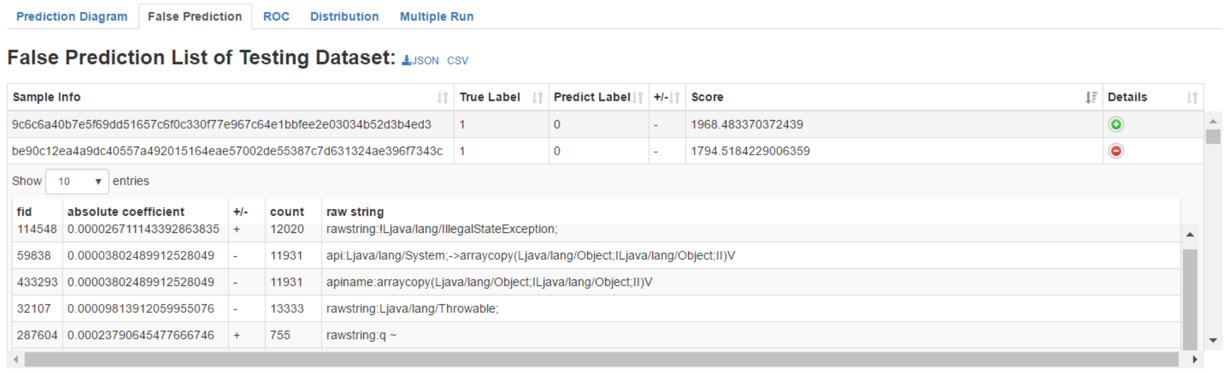
Feature List for Id:315728

Fid	Absolute Coefficient	+/-	Count	Raw String	N-gram	Excluded
425256	0.8397105312997859	+	8665	manifest:android:name="android.permission.READ_PHONE_STATE"	1198638337	○
266688	0.6861652019568169	+	5234	manifest:android:name="android.intent.action.BOOT_COMPLETED"	1367553321	○
192723	0.6616801759332225	-	10933	cert:Created-By: 1.0 (Android)	335998927	○
177339	0.640665473640811	+	3318	manifest:android:name="android.permission.SEND_SMS"	111806125	○
187117	0.638443412509326	+	7674	api:Landroid/telephony/TelephonyManager;->getDeviceId()Ljava/lang/String;	625306110	○
19557	0.6137784745741569	+	9409	manifest:service	1228916717	○



Prediction Results of Testing Dataset:

True Labels of Testing Dataset:

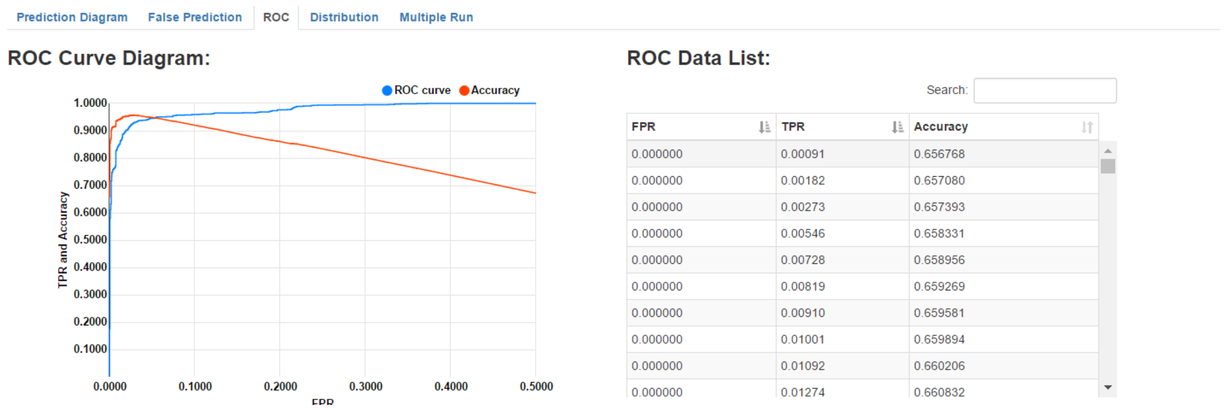


False Prediction List of Testing Dataset: JSON CSV

Sample Info	True Label	Predict Label	+/-	Score	Details
9c6c6a40b7e5f69dd51657c6f0c330f77e967c64e1bbfee2e03034b52d3b4ed3	1	0	-	1968.483370372439	⊕
be90c12ea4a9dc40557a492015164eae57002de55387c7d631324ae396f7343c	1	0	-	1794.5184229006359	⊖

Show 10 entries

fid	absolute coefficient	+/-	count	raw string
114548	0.000026711143392863835	+	12020	rawstring:!Ljava/lang/IllegalStateException;
59838	0.00003802489912528049	-	11931	api:Ljava/lang/System;->arraycopy(Ljava/lang/Object;ILjava/lang/Object;II)V
433293	0.00003802489912528049	-	11931	apiname:arraycopy(Ljava/lang/Object;ILjava/lang/Object;II)V
32107	0.00009813912059955076	-	13333	rawstring:Ljava/lang/Throwable;
287604	0.00023790645477666746	+	755	rawstring:q ~

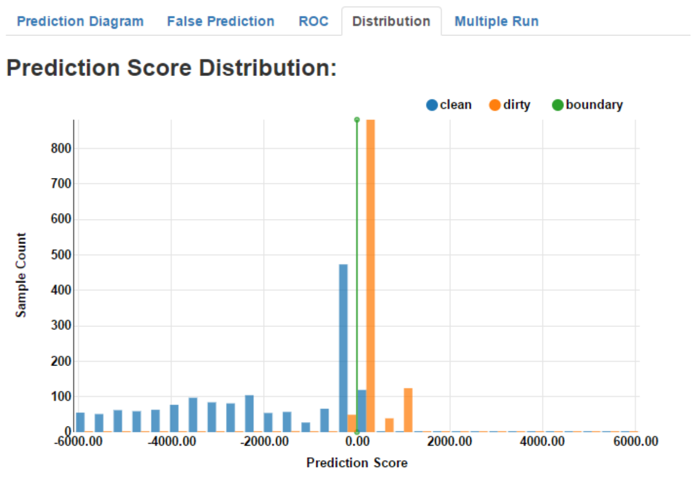


ROC Curve Diagram:

ROC Data List:

Search:

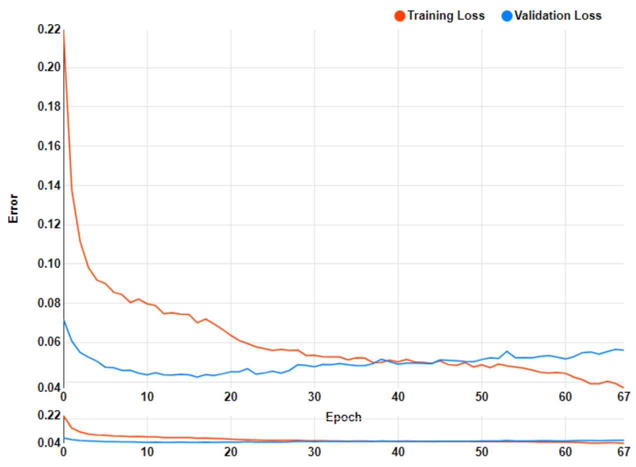
FPR	TPR	Accuracy
0.000000	0.00091	0.656768
0.000000	0.00182	0.657080
0.000000	0.00273	0.657393
0.000000	0.00546	0.658331
0.000000	0.00728	0.658956
0.000000	0.00819	0.659269
0.000000	0.00910	0.659581
0.000000	0.01001	0.659894
0.000000	0.01092	0.660206
0.000000	0.01274	0.660832



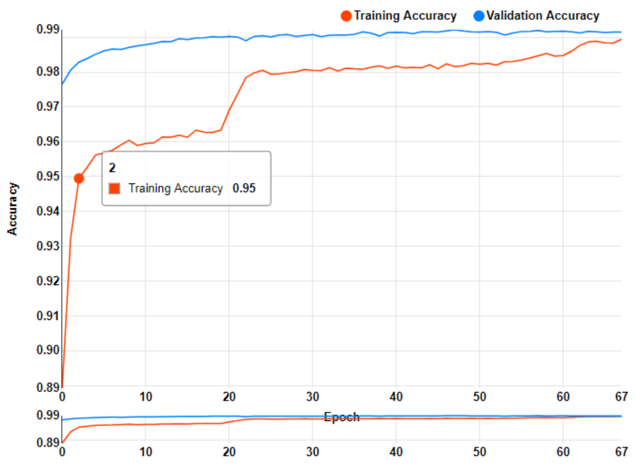
Prediction Score Distribution:

Prediction Diagram | DNN Model | DNN Status | False Prediction | ROC | Distribution | Multiple Run

Error Epoch Diagram:

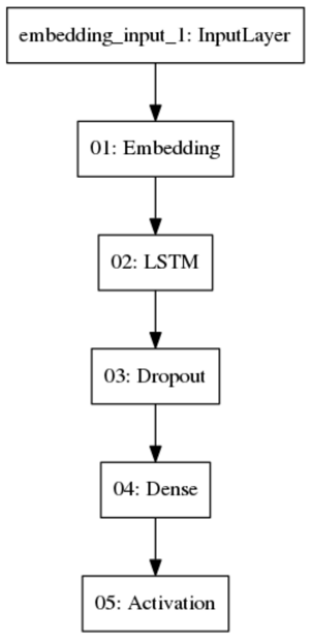


Accuracy Epoch Diagram:



Prediction Diagram | DNN Model | DNN Status

Model Diagram:



<u>Admin's quick reference</u>:

For HDFS cluster: Login to Hadoop master as user, `hadoop`, stop or start HDFS processes

```
# Start hdfs daemon, as sudo; by default service will run after bootup
sudo systemctl start hdfs
# Stop all dfs processes on all nodes, as hadoop
/home/hadoop/hadoop_latest/sbin/start-dfs.sh
# Start all dfs processes, as hadoop
/home/hadoop/hadoop_latest/sbin/stop-dfs.sh
```

For Spark cluster: Login to Spark master as user, `hadoop`, Stop and start Spark processes

```
# Start/stop spark daemon as sudo; by default service will run after bootup
sudo systemctl start|stop spark
# Stop all spark processes on all nodes, as hadoop
/home/hadoop/spark_latest/sbin/stop_all.sh
# Start all spark processes, as hadoop
/home/hadoop/spark_latest/sbin/start_all.sh
```

Restart Django Web: Login to web master as the user, `django`, start web processes

```
# Start web daemon; by default service will run after bootup
sudo systemctl start django_atdml

# start web processes, as django
cd /home/django/myml
python manage.py runserver 0.0.0.0:8000

# OR run as backend proc ===========
nohup python manage.py runserver 0.0.0.0:8000 > /dev/null 2>&1 &

# OR run as a service ===========
systemctl list-unit-files | grep

# stop web processes or daemon
Ctrl-C or kill the python process
```

<u>Sqlite3 DB</u>: `/home/django/myml/db.sqlite3`

Information for table**,** **`atdml_user_profile`**: limit the upload count per user:

- `count_upload`: current upload count
- `count_upload_max`: max upload count for current time window
- `count_upload_date`: starting date/time for time window
- `count_upload_period`: time window length in hour
- `acl_list`: tbd
- `user_id`: key to link to user table `auth_user`

To increase the upload count, increase the `count_upload_max` or decrease the `count_upload_period`

Information for table, **`atdml_document`**: for all data:

Field, `file_type`, define the types of records:

- <u>Dataset</u>: data stored in HDFS or uploaded through web
  `N-gram pattern`
  `N-gram JSON`
  `N-gram pattern gz`
  `ATD`
  `Libsvm Format`
  `Custom`: special custom featuring module
- <u>Classifier</u>: no training dataset, only pretrained ML models
  `ensemble`
  `image-inception`
  `image-yolo`
- <u>Prediction</u>: entry for prediction
  `predict`
  `ensemble_predict`
  `image_predict`
- <u>Emulation</u>: entry for APK emulation [+ prediction]
  `emulate`