```
Mounted at /content/drive
```

	hash	millisecond	classification	state	usage_counter	prio	static_prio	normal_prio	policy	vm_pgoff	...	nivcsw	min_flt	maj_f
0	42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...	0	malware	0	0	3069378560	14274	0	0	0	...	0	0	1
1	42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...	1	malware	0	0	3069378560	14274	0	0	0	...	0	0	1
2	42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...	2	malware	0	0	3069378560	14274	0	0	0	...	0	0	1
3	42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...	3	malware	0	0	3069378560	14274	0	0	0	...	0	0	1
4	42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...	4	malware	0	0	3069378560	14274	0	0	0	...	0	0	1

5 rows × 35 columns

```
Index(['hash', 'millisecond', 'classification', 'state', 'usage_counter',
       'prio', 'static_prio', 'normal_prio', 'policy', 'vm_pgoff',
       'vm_truncate_count', 'task_size', 'cached_hole_size', 'free_area_cache',
       'mm_users', 'map_count', 'hiwater_rss', 'total_vm', 'shared_vm',
       'exec_vm', 'reserved_vm', 'nr_ptes', 'end_data', 'last_interval',
       'nvcsw', 'nivcsw', 'min_flt', 'maj_flt', 'fs_excl_counter', 'lock',
       'utime', 'stime', 'gtime', 'cgtime', 'signal_nvcsw'],
      dtype='object')
```

Column Description of Malwares

Features Description	Properties
hash APK/ SHA256	file name
milisecond	time
classification	malware/beign
state	flag of unrunable/runnable/stopped tasks
usage_counter	task structure usage counter
prio	keeps the dynamic priority of a process
static_prio	static priority of a process
normal_prio	priority without taking RT-inheritance into account
policy	planning policy of the process
vm_pgoff	the offset of the area in the file, in pages.
vm_truncate_count	used to mark a vma as now dealt with
task_size	size of current task.
cached_hole_size	size of free address space hole.
free_area_cache	first address space hole
mm_users	address space users
map_count	number of memory areas
hiwater_rss	peak of resident set size
total_vm	total number of pages

Features Description	Properties
policy	planning policy of the process
vm_pgoff	the offset of the area in the file, in pages.
vm_truncate_count	used to mark a vma as now dealt with
task_size	size of current task.
cached_hole_size	size of free address space hole.
free_area_cache	first address space hole
mm_users	address space users
map_count	number of memory areas
hiwater_rss	peak of resident set size
total_vm	total number of pages
shared_vm	number of shared pages.
exec_vm	number of executable pages.
reserved_vm	number of reserved pages.
nr_ptes	number of page table entries
end_data	end address of code component
last_interval	last interval time before thrashing
nvcsw	number of volunteer context switches.
nivcsw	number of in-volunteer context switches
min_flt	minór page faults
maj_flt	majór page faults
fs_excl_counter	it holds file system exclusive resources.
lock	the read-write synchronization lock used for file system access
utime	user time
stime	system time
gtime	guest time
cgtime	cumulative group time. Cumulative resource counter
signal_nvcsw	used as cumulative resource counter.

	hash	millisecond	classification	state	usage_counter	prio	static_prio	normal_prio	policy	vm_pgoff	...	
count	100000	100000.000000	100000	1.000000e+05	100000.0	1.000000e+05	100000.000000	100000.0	100000.0	100000.0	...	1000
unique	100	NaN	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	
top	42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...	NaN	malware	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	
freq	1000	NaN	50000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	
mean	NaN	499.500000	NaN	1.577683e+05	0.0	3.069706e+09	18183.900070	0.0	0.0	0.0	...	
std	NaN	288.676434	NaN	9.361726e+05	0.0	2.963061e+05	4609.792765	0.0	0.0	0.0	...	
min	NaN	0.000000	NaN	0.000000e+00	0.0	3.069190e+09	13988.000000	0.0	0.0	0.0	...	
25%	NaN	249.750000	NaN	0.000000e+00	0.0	3.069446e+09	14352.000000	0.0	0.0	0.0	...	
50%	NaN	499.500000	NaN	0.000000e+00	0.0	3.069698e+09	16159.000000	0.0	0.0	0.0	...	
75%	NaN	749.250000	NaN	4.096000e+03	0.0	3.069957e+09	22182.000000	0.0	0.0	0.0	...	
max	NaN	999.000000	NaN	4.326605e+07	0.0	3.070222e+09	31855.000000	0.0	0.0	0.0	...	3

11 rows × 35 columns

```
Data columns (total 35 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   hash               100000 non-null  object
 1   millisecond        100000 non-null  int64
 2   classification     100000 non-null  object
 3   state              100000 non-null  int64
 4   usage_counter      100000 non-null  int64
 5   prio               100000 non-null  int64
 6   static_prio        100000 non-null  int64
 7   normal_prio        100000 non-null  int64
 8   policy             100000 non-null  int64
 9   vm_pgoff           100000 non-null  int64
 10  vm_truncate_count  100000 non-null  int64
 11  task_size          100000 non-null  int64
 12  cached_hole_size   100000 non-null  int64
 13  free_area_cache    100000 non-null  int64
 14  mm_users           100000 non-null  int64
 15  map_count          100000 non-null  int64
 16  hiwater_rss        100000 non-null  int64
 17  total_vm           100000 non-null  int64
 18  shared_vm          100000 non-null  int64
 19  exec_vm            100000 non-null  int64
 20  reserved_vm        100000 non-null  int64
 21  nr_ptes            100000 non-null  int64
 22  end_data           100000 non-null  int64
 23  last_interval      100000 non-null  int64
 24  nvcsw              100000 non-null  int64
 25  nivcsw             100000 non-null  int64
 26  min_flt            100000 non-null  int64
 27  maj_flt            100000 non-null  int64
 28  fs_excl_counter    100000 non-null  int64
 29  lock               100000 non-null  int64
 30  utime              100000 non-null  int64
 31  stime              100000 non-null  int64
 32  gtime              100000 non-null  int64
 33  cgtime             100000 non-null  int64
 34  signal_nvcsw       100000 non-null  int64
dtypes: int64(33), object(2)
memory usage: 26.7+ MB
```

```
data['classification'] = data.classification.map({'benign':0, 'malware':1})
data.head()
```

	hash	millisecond	classification	state	usage_counter	prio	static_prio	normal_prio	policy	vm_pgoff	...	nivcsw	min_flt	maj_f
0	42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...	0	1	0	0	3069378560	14274	0	0	0	...	0	0	1
1	42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...	1	1	0	0	3069378560	14274	0	0	0	...	0	0	1
2	42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...	2	1	0	0	3069378560	14274	0	0	0	...	0	0	1
3	42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...	3	1	0	0	3069378560	14274	0	0	0	...	0	0	1
4	42fb5e2ec009a05ff5143227297074f1e9c6c3ebb9c914...	4	1	0	0	3069378560	14274	0	0	0	...	0	0	1

5 rows × 35 columns

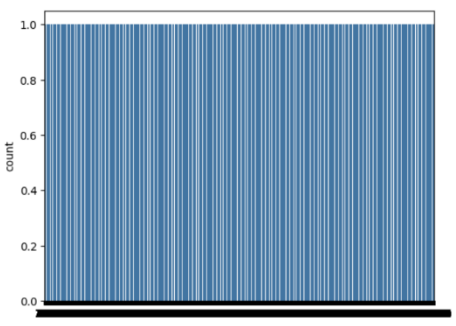
```
[7] data["classification"].value_counts()
```

```
classification
1    50000
0    50000
Name: count, dtype: int64
```

```
data.head()
```

	hash	millisecond	classification	state	usage_counter	prio	static_prio	normal_prio	policy	vm_pgoff	...	nivcsw	min_flt	maj_
0	1efc135b8f924076b56564ee93f5a82794671e0f0da08b...	263	1	0	0	3069583360	22185	0	0	0	...	19	0	
1	com.imangi.templerun2.apk	299	0	0	0	3069620224	19641	0	0	0	...	2	1	
2	116ae92ecfacb70146fe643d92878e522f71af393702f3...	56	1	4096	0	3070173184	14020	0	0	0	...	0	1	
3	54b860b1c538d915a68a1980afd25c029458ade80e4175...	605	1	4096	0	3069480960	22191	0	0	0	...	25	1	
4	32effc5a6bc3b7319b5b7da02a7cc3576d44c1794b335b...	279	1	0	0	3069267968	13996	0	0	0	...	0	1	

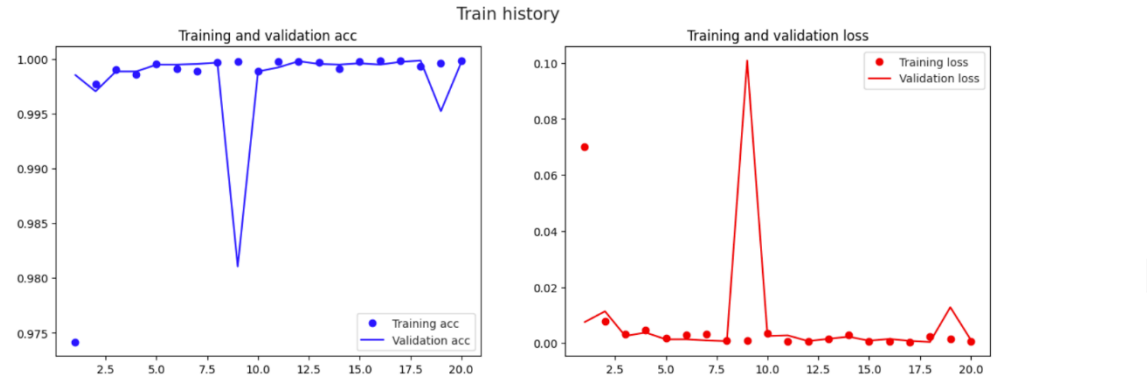
5 rows × 35 columns



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 50)                1400

 dense_1 (Dense)             (None, 50)                2550

 dense_2 (Dense)             (None, 50)                2550

 dense_3 (Dense)             (None, 50)                2550

 dense_4 (Dense)             (None, 50)                2550

 dense_5 (Dense)             (None, 50)                2550

 dense_6 (Dense)             (None, 2)                 102

=================================================================
Total params: 14252 (55.67 KB)
Trainable params: 14252 (55.67 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
Epoch 1/20
640/640 [==============================] - 5s 5ms/step - loss: 0.0700 - accuracy: 0.9742 - val_loss: 0.0075 - val_accuracy: 0.9986
Epoch 2/20
640/640 [==============================] - 2s 4ms/step - loss: 0.0077 - accuracy: 0.9977 - val_loss: 0.0113 - val_accuracy: 0.9971
Epoch 3/20
640/640 [==============================] - 3s 4ms/step - loss: 0.0032 - accuracy: 0.9990 - val_loss: 0.0025 - val_accuracy: 0.9989
Epoch 4/20
640/640 [==============================] - 2s 3ms/step - loss: 0.0047 - accuracy: 0.9986 - val_loss: 0.0038 - val_accuracy: 0.9989
Epoch 5/20
640/640 [==============================] - 2s 3ms/step - loss: 0.0016 - accuracy: 0.9996 - val_loss: 0.0013 - val_accuracy: 0.9995
Epoch 6/20
640/640 [==============================] - 2s 3ms/step - loss: 0.0028 - accuracy: 0.9991 - val_loss: 0.0014 - val_accuracy: 0.9995
Epoch 7/20
640/640 [==============================] - 3s 5ms/step - loss: 0.0032 - accuracy: 0.9989 - val_loss: 9.4262e-04 - val_accuracy: 0.9996
Epoch 8/20
640/640 [==============================] - 2s 4ms/step - loss: 8.8020e-04 - accuracy: 0.9997 - val_loss: 7.0321e-04 - val_accuracy: 0.9997
Epoch 9/20
640/640 [==============================] - 2s 3ms/step - loss: 7.8528e-04 - accuracy: 0.9998 - val_loss: 0.1009 - val_accuracy: 0.9811
Epoch 10/20
640/640 [==============================] - 2s 3ms/step - loss: 0.0036 - accuracy: 0.9989 - val_loss: 0.0025 - val_accuracy: 0.9989
Epoch 11/20
640/640 [==============================] - 2s 3ms/step - loss: 7.2168e-04 - accuracy: 0.9998 - val_loss: 0.0027 - val_accuracy: 0.9992
Epoch 12/20
640/640 [==============================] - 2s 3ms/step - loss: 6.9124e-04 - accuracy: 0.9998 - val_loss: 7.1819e-04 - val_accuracy: 0.9998
Epoch 13/20
640/640 [==============================] - 2s 4ms/step - loss: 0.0015 - accuracy: 0.9997 - val_loss: 0.0015 - val_accuracy: 0.9996
Epoch 14/20
640/640 [==============================] - 3s 5ms/step - loss: 0.0029 - accuracy: 0.9992 - val_loss: 0.0023 - val_accuracy: 0.9995
Epoch 15/20
640/640 [==============================] - 2s 3ms/step - loss: 7.0402e-04 - accuracy: 0.9998 - val_loss: 8.6271e-04 - val_accuracy: 0.9996
Epoch 16/20
640/640 [==============================] - 2s 3ms/step - loss: 5.5257e-04 - accuracy: 0.9998 - val_loss: 0.0015 - val_accuracy: 0.9995
Epoch 17/20
640/640 [==============================] - 2s 3ms/step - loss: 4.8064e-04 - accuracy: 0.9998 - val_loss: 7.9443e-04 - val_accuracy: 0.9998
Epoch 18/20
640/640 [==============================] - 2s 3ms/step - loss: 0.0025 - accuracy: 0.9994 - val_loss: 4.0347e-04 - val_accuracy: 0.9999
Epoch 19/20
640/640 [==============================] - 2s 3ms/step - loss: 0.0016 - accuracy: 0.9996 - val_loss: 0.0128 - val_accuracy: 0.9952
Epoch 20/20
```

```
<ipython-input-20-57c12cf59d48>:18: UserWarning: color is redundantly defined by the 'color' keyword argument and the fmt string "bo" (-> color='b'). The keyword argument will take pre
  ax2.plot(epochs, loss, "bo", label = "Training loss", color = 'red')
<ipython-input-20-57c12cf59d48>:19: UserWarning: color is redundantly defined by the 'color' keyword argument and the fmt string "b" (-> color=(0.0, 0.0, 1.0, 1)). The keyword argument
  ax2.plot(epochs, val_loss, "b", label = "Validation loss", color = 'red')
```



```
[21] print('\nTest loss: {0:.6f}. Test accuracy: {1:.6f}%'.format(test_loss, test_accuracy*100.))
```

```
625/625 [==============================] - 1s 2ms/step - loss: 7.4599e-04 - accuracy: 0.9998

Test loss: 0.000746. Test accuracy: 99.980003%
```

Further train the model using SGD with lr=0.001

+ Code + Text

```
[22] from keras.optimizers import SGD
     sgd = SGD(lr=0.001, momentum=0.9, nesterov=True)
     model.compile(optimizer = sgd, loss = "sparse_categorical_crossentropy", metrics=['accuracy'])
```

```
WARNING:absl:`lr` is deprecated in Keras optimizer, please use `learning_rate` or use the legacy optimizer, e.g.,tf.keras.optimizers.legacy.SGD.
```

```
[23] result = model.fit(x=x_train,
                        y=y_train,
                        batch_size=batch_size,
                        epochs=30,
                        verbose=1,
                        initial_epoch=10, #start from epoch 11
                        callbacks=[early_stopping], #prevent overfitting
                        validation_split=0.2)
```

```
Epoch 11/30
640/640 [==============================] - 3s 3ms/step - loss: 0.0039 - accuracy: 0.9990 - val_loss: 7.5472e-04 - val_accuracy: 0.9998
Epoch 12/30
640/640 [==============================] - 2s 4ms/step - loss: 9.0451e-04 - accuracy: 0.9997 - val_loss: 7.8910e-04 - val_accuracy: 0.9997
Epoch 13/30
640/640 [==============================] - 3s 5ms/step - loss: 5.3807e-04 - accuracy: 0.9998 - val_loss: 4.4936e-04 - val_accuracy: 0.9998
Epoch 14/30
640/640 [==============================] - 2s 3ms/step - loss: 1.2546e-04 - accuracy: 1.0000 - val_loss: 8.6132e-04 - val_accuracy: 0.9998
Epoch 15/30
640/640 [==============================] - 2s 3ms/step - loss: 5.8184e-04 - accuracy: 0.9999 - val_loss: 5.6563e-04 - val_accuracy: 0.9998
```

```
625/625 [==============================] - 1s 2ms/step - loss: 1.9545e-04 - accuracy: 0.9999

Test loss: 0.000195. Test accuracy: 99.994999%
```

	hash	millisecond	classification	state	usage_counter	prio	static_prio	normal_prio	policy	vm_pgoff	...	nivcsw	min_flt	maj_
0	1efc135b8f924076b56564ee93f5a82794671e0f0da08b...	263	NaN	0	0	3069583360	22185	0	0	0	...	19	0	
1	com.imangi.templerun2.apk	299	NaN	0	0	3069620224	19641	0	0	0	...	2	1	
2	116ae92ecfacb70146fe643d92878e522f71af393702f3...	56	NaN	4096	0	3070173184	14020	0	0	0	...	0	1	
3	54b860b1c538d915a68a1980afd25c029458ade80e4175...	605	NaN	4096	0	3069480960	22191	0	0	0	...	25	1	
4	32effc5a6bc3b7319b5b7da02a7cc3576d44c1794b335b...	279	NaN	0	0	3069267968	13996	0	0	0	...	0	1	

5 rows × 35 columns

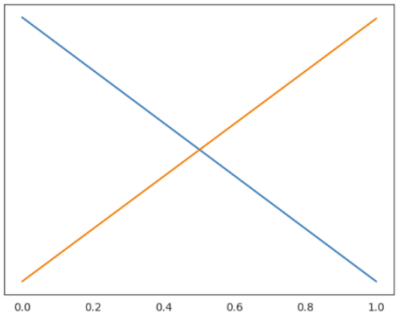
```
Mean Absolute Error: 1.375e-05
Mean Squared Error: 2.9375000000000007e-06
Root Mean Squared Error: 0.0017139136501002612
```

```
RandomForestClassifier()
```

```
array([0, 1, 0, ..., 1, 0, 0])
```

```
[[40088     0]
 [    0 39912]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     40088
           1       1.00      1.00      1.00     39912

    accuracy                           1.00     80000
   macro avg       1.00      1.00      1.00     80000
weighted avg       1.00      1.00      1.00     80000

1.0
```

```
[<matplotlib.lines.Line2D at 0x789a7b488910>]
```



	precision	recall	f1-score	support
0	1.00	1.00	1.00	40088
1	1.00	1.00	1.00	39912
accuracy			1.00	80000
macro avg	1.00	1.00	1.00	80000
weighted avg	1.00	1.00	1.00	80000

```
<Axes: ylabel='count'>
```