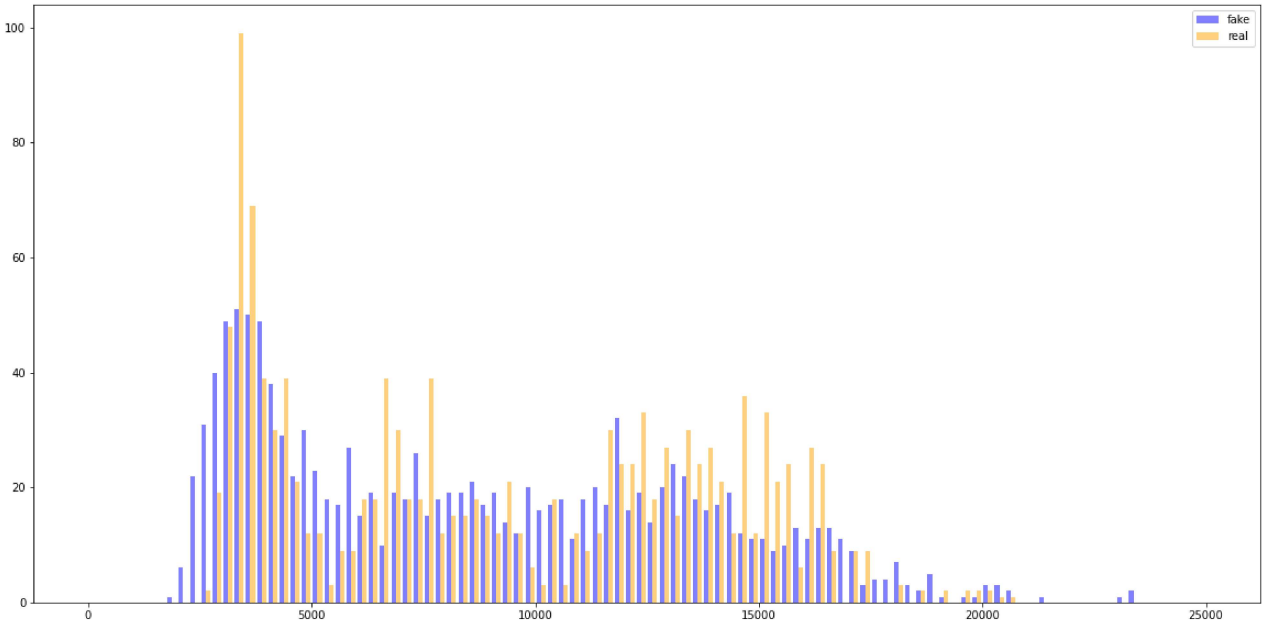


In [38]:
```
# simulate 20 days data using previous 3 days data
def forward_simulate(G, backward_data,n, latent_dim, ts_dim, conditional, use_cuda):
    noise = torch.randn((n,1,latent_dim))
    backward_data = torch.from_numpy(backward_data)

    if conditional>0:
        noise[:,:,:conditional] = backward_data[:,:,-conditional:]

    if use_cuda:
        noise = noise.cuda()
        real_samples = real_samples.cuda()
        G.cuda()

    y = G(noise)
    y = y.float()
    if use_cuda:
        y = y.cuda()
    return y.float()
```

In [39]:
```
data_load_path =r'C:/Users/dai/Code/WGAN_financial_time-series-master/data/^GSPTSE_mont
data = Data(ts_dim, data_load_path)
```

```
(111, 17)
```

In [40]:
```
fake_data, real_data, real_start_prices = get_all_samples(data, generator, latent_dim,
```

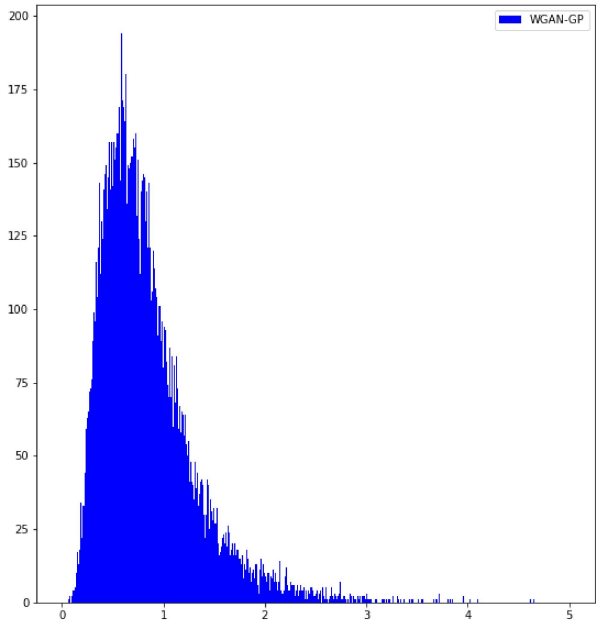
In [41]:
```
real_data[-1]
```

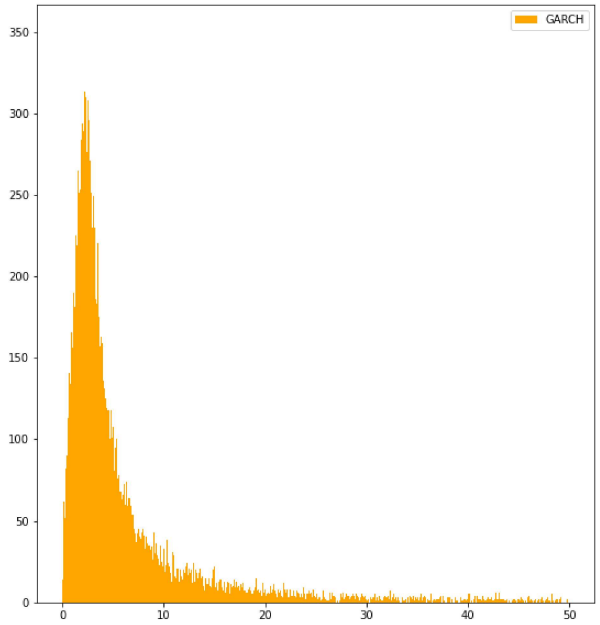
Out[41]:
```
tensor([[-3.3239e-01,  6.2045e-01, -4.6673e-01,  4.2272e-01,  4.0818e-02,
         -1.9390e-01, -1.8552e+01, -1.4172e-02, -1.4172e-02, -1.4172e-02,
         -1.4172e-02, -1.4172e-02, -1.4172e-02, -1.4172e-02, -1.4172e-02,
         -1.4172e-02, -1.4172e-02]])
```

In [42]:
```
real_start_prices[-1]
```

Out[42]:
```
20150.9
```

In [43]:
```
data.post_processing(real_data[-1].reshape(ts_dim),real_start_prices[-1])
```



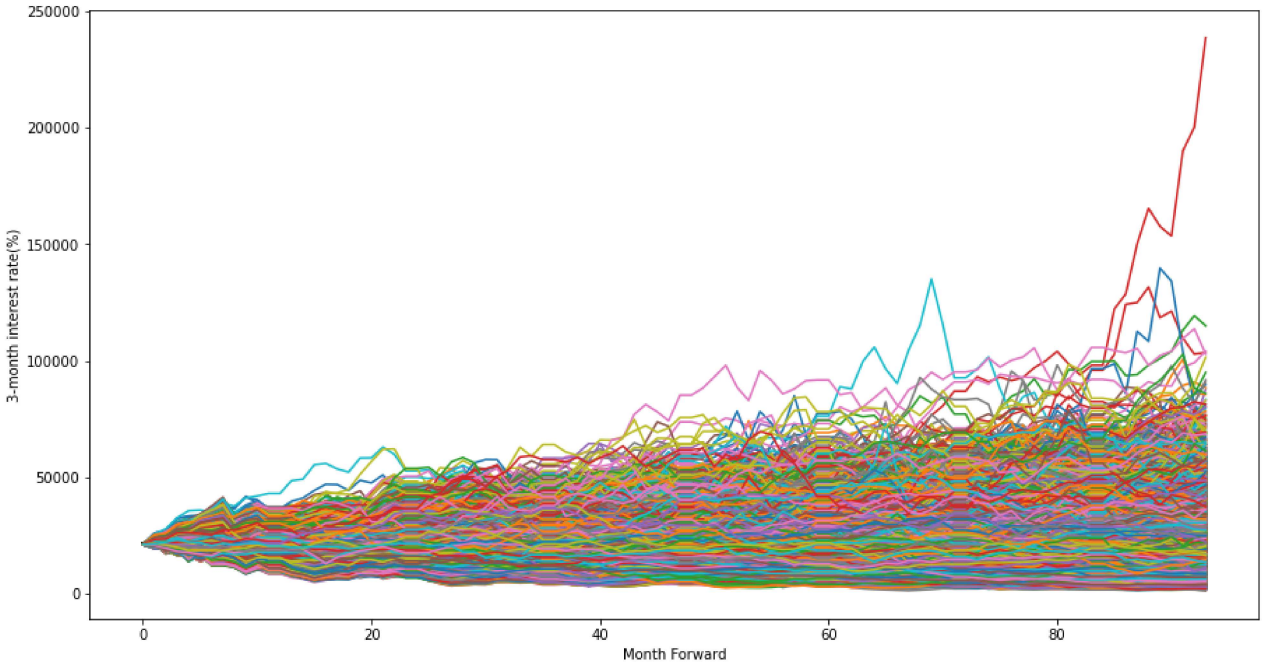


In [ ]:

In [69]:
```
plt.figure(figsize=(15, 8))
for i in range(0,28000):
    plt.plot(all_simulation[i,:94])


plt.xlabel('Month Forward')
plt.ylabel('3-month interest rate(%)')
```

Out[69]: Text(0, 0.5, '3-month interest rate(%)')



In [ ]:

In [ ]:

In [ ]: