Algorithm 1 Ghost Backdoor based on Neuron Select

Input: central server C_s, a set of all client C, end epoch E_e, current client C_i, learning rate η, dataset D, mask matrix $R_{mask}^{r\times d}$, ghost neurons' values matrix $R_{V_s}^{r\times d}$

Output: a global model with high accuracy, stealth backdoor and high accuracy in main-task

```
$C_s$ select $n$ clients by random into $C_m$
$C_s$ build a global model $G$
$C_s$ send $G$ to each client in $C_m$
choose the ghost neurons
pre-train with benign samples to collect the values of every neurons
choose $V_s$ as trigger
for epoch $<$ $E_e$ do
    for number $k$ of client in $C_m$ do
        Download $G$ as local model $L$ and train $L$ by benign datasets $D$,
        Compute gradient by $D$ on batch $B_i$ of size $\ell$
        $g_i = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \mathscr{L}(\theta_{C_i}, D)$
        if client $C_i$ is advisary and epoch mod $N_{attack}$ = 0 then
            $\hat{g}_i = g_i * R_{mask}^{r\times d} + R_{V_s}^{r\times d}$
            Update $\theta_{C_{i+1}} = \theta_{C_i} - \eta \hat{g}_i$
        else
            Update $\theta_{C_{i+1}} = \theta_{C_i} - \eta g_i$
        Upload $\theta_{C_{i+1}}$ to $C_s$
        $C_s$ recieve $\sum_1^k \theta_{C_{i+k}}$ and generate update gradient $U$ for $G$
$G_{i+1} = G_i - U_i$
return Final global model $G$ with backdoor
```