# A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks

**Keywords: Botnet**

Lyu Jiuyang，Jan 9th, 2022.

## 介绍

**作者：** Mohammad Alauthaman, Nauman Aslam, Li Zhang, Rafe Alasem & M. A. Hossain. Neural Comput & Applic 2018

## 背景

僵尸网络（Botnet）是由攻击者 (Botmaster) 远程管理的受感染计算机 (Bots) 网络。 可以命令僵尸网络执行各种恶意活动，例如发送垃圾邮件、网络钓鱼、点击欺诈、DDoS 和传播恶意软件。

Botmaster 构建了一个通信通道的基础设施，向 Bot 发送命令并接收结果。 此通信通道称为命令和控制 (C&C) 通道。 僵尸网络和其他恶意软件的主要区别在于 C&C 中使用的基础设施。 与用于单独执行恶意行为的其他恶意软件相比，僵尸网络作为一组基于 C&C 通信通道的受感染主机工作。根据 C&C 基础设施，僵尸网络可分为两大类：集中式和分散式 C&C [5]。 在集中式僵尸网络中，Botmaster 通常使用 C&C 服务器向 Bot 发送命令。

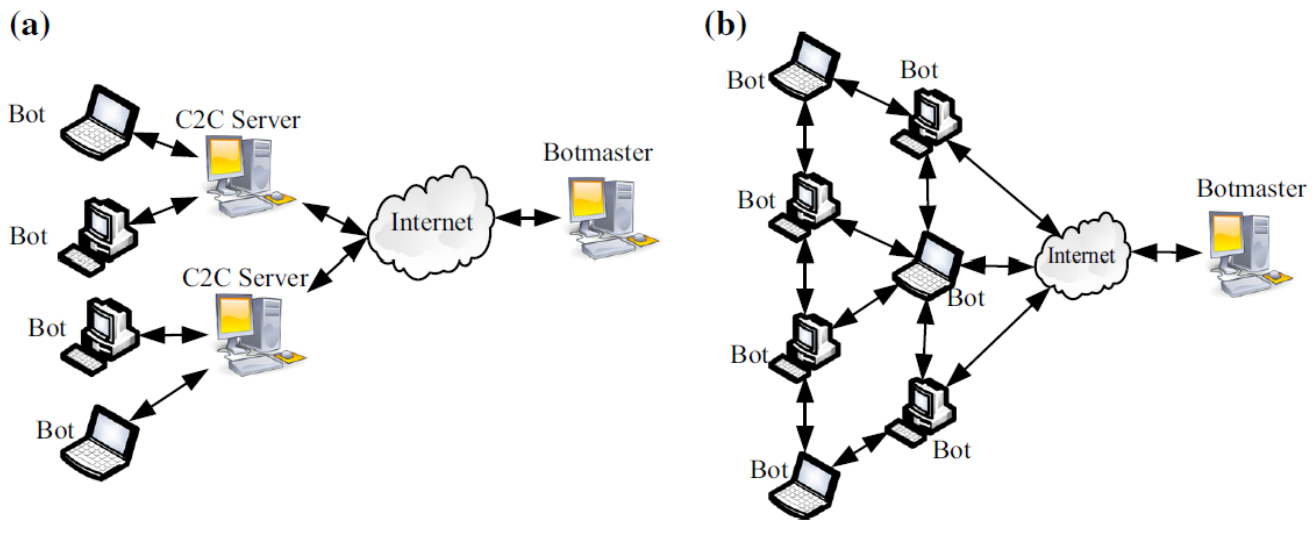


**Fig. 1** Structures of the Botnet. **a** Centralized structure, **b** decentralized structure

## 难度

1. P2P Botnet 的流量与正常流量非常相似。
2. 许多 P2P 僵尸网络使用了加密算法，无法进行数据包检查
3. P2P 僵尸网络中没有中央服务器，使用随机端口通信。

## 贡献

- A network traffic reduction approach that has been designed will be able to increase the performance of the proposed framework.
- A connection-based detection mechanism is independent of payload and uses only the information obtained from the header of TCP control packet. Thus, it does not need deep packet inspection and cannot be confused with payload encryption techniques.
- Adopting the classification and regression trees to select the important connection features in order to decrease the size and dimensionality of the dataset.

好长的related work...

首先，本系统被动监控网络流量。 其次，它利用了这样一个事实，即在传播阶段，Bots 将与其 C&C 服务器/对等方显示频繁的通信行为，以便发现其他对等方并接收最新的任务更新。

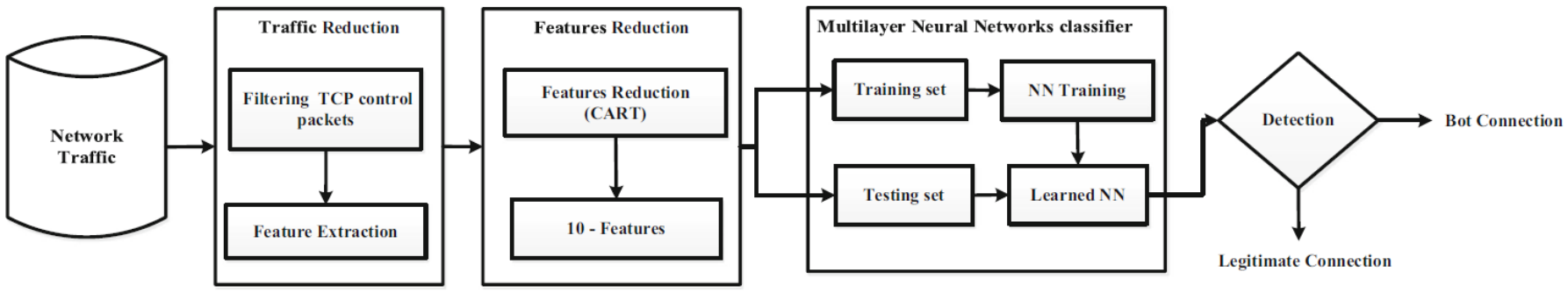


**Fig. 2** Block diagram of the proposed system

## 核心流程

先用CART降维，再扔到NN里分类。

# 方法论

## 减少网络流量

只保留具有SYN, ACK, FIN 和 RST的TCP包。

## 特征提取

基本数据格式：`（源 IP 地址、目标 IP 地址、源端口和目标端口）`四元组

基于 30秒连接 的29个属性。

**Table 1** Selected features of network traffic connections

| Features | Description |
|---|---|
| F1 | Number of control packets per flow in a given time interval |
| F2 | Number of control packets transmitted per flow in a given time interval |
| F3 | Number of control packets received per flow in a given time interval |
| F4 | Number of transmitting bytes per flow in a given time interval |
| F5 | Number of received bytes per flow in a given time interval |
| F6 | Number of transmitted SYN packets per flow in a given time interval |
| F7 | Number of received SYN packets per flow in a given time interval |
| F8 | Number of transmitted ACK packets per flow in a given time interval |
| F9 | Number of received ACK packets per flow in a given time interval |
| F10 | Number of transmitted duplicate ACK packets per flow in a given time interval |
| F11 | Number of received duplicate ACK packets per flow in a given time interval |
| F12 | Average length of transmitted control packets per flow in a given time interval |
| F13 | Average length of received control packets per flow in a given time interval |
| F14 | Average length of control packets per flow in a given time interval |
| F15 | Number of transmitted failed connection per flow in a given time interval |
| F16 | Number of received failed connection per flow in a given time interval |
| F17 | Number of transmitted ACK packets have a sequence one per flow in a given time interval |
| F18 | Number of received ACK packets have a sequence one per flow in a given time interval |
| F19 | Number of transmitted SYN–ACK packets per flow in a given time interval |
| F20 | Number of received SYN–ACK packets per flow in a given time interval |
| F21 | Total number of bytes per flow in a given time interval |
| F22 | Ratio of incoming control packets per flow in a given time interval |
| F23 | Ratio of average length of outgoing packets over the average length of control packets per flow in a given time interval |
| F24 | F6–F20 |
| F25 | Number of transmitted FIN–ACK packets per flow in a given time interval |
| F26 | Number of received FIN–ACK packets per flow in a given time interval |
| F27 | Number of transmitted RST–ACK packets per flow in a given time interval |
| F28 | Number of received RST–ACK packets per flow in a given time interval |
| F29 | Average time between an attempt to create connection per flow in a given time interval |

## 特征缩减

选择合适的特征，降低模型复杂度，提高性能。。

本文使用了三种方法进行降维：

- CART（分类和回归树）（<- 也是本文提出的）
- RelieF （一个比较经典的分类权重算法）

Algorithm 2 Pseudo code of ReliefF

**Input**: the dataset contains an instance with class labels.
**Output**: W (*f*) features ranking.
Number of features = *n*;
Set all weight W (*f*)=0;
Number of iterations =*m*;
For *i* = 1 to *m* do
    Randomly select an instance R*i*;
    Find *k* nearest hit H*i*;
        Foreach class *c* <> class (R*i*) do
        From class *c* find *k* nearest misses M*j*(*c*);
        End For
    For *f*=1 to *n*

$$W(f) = W(f) - \sum_{j=1}^{k} \frac{\mathrm{diff}(f, Ri, Hj)}{m \times k} + \sum_{c <> class(Ri)} \frac{\left[\frac{P(c)}{1-P(class(Ri))} \sum_{j=1}^{k} \mathrm{diff}(f, Ri, Mj(c))\right]}{m \times k};$$

    End For
End For
Return W(f);

$diff(A, R_1, R_2)$ 样本$R_1$和$R_2$在特征$A$上的差。

- PCA（但没说是基于特征值分解还是SVD）。本文降到了10维，保留0.95的信息。

**Table 3** Feature reduction with the CART, PCA and ReliefF algorithms

| Feature selection algorithm | Features number | Feature list |
|---|---|---|
| CART | 10 | F3, F13, F23, F21, F14, F29, F12, F1, F4, F15 |
| PCA | 10 | Linear combination of features |
| ReliefF | 10 | F27, F25, F15, F6, F22, F24, F29, F23, F26, F19 |

各特征权重结果忽略。

## 神经网络

使用Rpop（弹性反向传播）算法（为啥不用RMSProp捏）

$$\text{权重变化量}\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}(t), & \text{if } \frac{\partial E(t)}{\partial w_{ij}} > 0 \\ +\Delta_{ij}(t), & \text{if } \frac{\partial E(t)}{\partial w_{ij}} < 0 \\ 0, & \text{else} \end{cases}$$

$$\text{每一轮更新的值}\Delta_{ij}^{(t)} = \begin{cases} \eta^{+} \cdot \Delta_{ij}(t), & \text{if } \frac{\partial E(t-1)}{\partial w_{ij}} \cdot \frac{\partial E(t)}{\partial w_{ij}} > 0 \\ \eta^{-} \cdot \Delta_{ij}(t), & \text{if } \frac{\partial E(t-1)}{\partial w_{ij}} \cdot \frac{\partial E(t)}{\partial w_{ij}} < 0 \\ \Delta_{ij}(t-1), & \text{else} \end{cases}$$

输入维度10（上文提及的10个特征），输出维度2，（又没写有几层[1]...）

[1] Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth Inc., Belmont, California.

## 实验

数据集：ISOT，LBNL

结论 CART 表现最好且用时最短，优于sota，PCA用时短于RelieF但是效果略差。

## 其他

文章写的废话比较多，对零基础读者比较友好，但是本身用的核心的东西只有CART，其他更多的只是用来比较。

只使用TCP包以进行网络流量减少是一个很好的操作，对于去中心化的、加密的Botnet检测（目测）很有效。