

DDoS Detection and Mitigation SDN using Support Vector Machine

Prajwal S, Siddhartha M, Charan S, Girish L
Department of Computer Science & Engineering, Visveswaraya Technological University,
Channabasaveshwara Institute of Technology, Gubbi, Tumkur,
Karnataka, India
prajwalkeerthi13@gmail.com, charan.siddu0817@gmail.com

***Abstract* — In recent years, the rise of software-defined networks (SDN) have made network control more flexible, easier to set up and manage, and have provided a stronger ability to adapt to the changing demands of application development and network conditions. The network becomes easier to maintain, but also achieves improved security as a result of SDN. The architecture of SDN is designed for Control Plane and Forwarding Plane separation and uses open APIs to realize programmable control. SDN allows for the importing of third- party applications to improve network service, or even provide a new network service. In this paper, we present a defense mechanism, which can find attack packets previously identified through the Sniffer function, and once the abnormal flow is found, the protection mechanism of the Firewall function will be activated. For the capture of the packets, available libraries will be used to determine the properties and contents of the malicious packet, and to anticipate any possible attacks. Through the prediction of all latent malicious behaviors, our new defense algorithm can prevent potential losses like system failures or crashes and reduce the risk of being attacked.**

Keywords—Firewall; Packet Sniffer; Software Defined Networks; SDN; OpenFlow; Controller; Defense Mechanism

I. INTRODUCTION

In traditional network architecture, it is a very complicated thing that to modify the configuration of the network devices such as switches, routers and firewalls after the deployment according to the needs of the business. In a rapidly changing of the network demand of the business environment, high stabilityand high performance network is not enough to meet the business needs, flexibility and agility but more critical. SDN separates the control of network devices from a centralized controller. Networks can operate without relying on the underlying network devices such as routers, switches, firewalls and so on, and can shield the differences from the underlying network equipment. The user can customize any network routing and define the policy rules [1], [2]. The control, which is completely open, is more flexible and intelligent.

The concept of SDN, which is also its advantage, is to separate the control layer and data layer of the switch. Then use SSL encryption technology to establish a secure transmission channel between the control layer and the data layer. OpenFlow controller will transmit the routing table that has been set completed to the network equipment in the data layer for packet delivery through the transmission channel [3], [4]. With centralized management, which allows the network equipment of the data layer obey controller in the control layer is just responsible for a simple task of packet forwarding. Since the completion of the transmission path ispre-set, the switch does not need to find a packet transmission path through keeping learning. This will greatly enhance the transmission efficiency, and reduce latency time. Through an open architecture to improve network performance and management flexibility, hoping to improve the deficiency of traditional network such as the lack of flexibility, speed of response to changes in demand retardation, expensive management and equipment costs and other issues [5]. Also, by through the network interface of SDN to assist operators or software developers, to combine the network status information and network resource utilization, and lead to a great development innovative service.

In SDN architecture, the path of the packet is no longer in accordance with the packet header and the routing table, but defined by the software. SDN architecture not only solves the blind spot in traditional network architecture, the controller also provide API that allows third-party users can develop policies and related applications that based on what business need [6]. Such as network security management, load balancing, and QoS of bandwidth management [7], [8]. That is, SDN is a programmable open network architecture, which allows network managers can easily change the logic topology without modify any network entity. It makes the network behavior can be dynamically and more quickly to adjust to cope with the services and applications.

The basic operation mode of SDN is to determine how the packet should be forwarded by querying the flow table when

OpenFlow switch receives a packet. First, to compare the value in each field of the packet header with the header field of the flow entry in the flow table. If the compare is successful, then the motion set in Actions field of the flow entry will be performed, and accumulate the value in the counter. If it could not find any match flow entry, the packets shall be passed to the controller through Packet-In message of OpenFlow switch. After the controller determines how to handle the packet, it will inform OpenFlow switch through OpenFlow protocol [9]. The result may be to write a flow entry in the flow table or to order OpenFlow switch to forward the packet directly and so on.

OpenFlow technology considers the packet forwarding with flow-based routing path, and transmit a packet by an exclusive transmission path [10], [11]. Administrators can set the network management functions and pre-establish the logic network to determine the packet transmission, for example, to determine which switch to pass through or how much bandwidth is required for transmission. Then to set the transmission path to OpenFlow routing table, that is, the flow table [12].

In this paper, we make use of the characteristics of SDN to design an attack detection and defense mechanism. To detect possible attacks by through the detection system, and perform automated defense policies by the firewall with network function virtualization when an attack is detected. It can timely and efficient manner to resolve attack or even do the complete defense. The rest of the paper is organized as follows. Next section mentions related work. The structure of the proposed method in detail is described in section III. Section IV shows the efficiency analysis of the proposed method. Then section V demonstrates the use case, and also evaluates the performance of our method. Finally, section VI is the summary and conclusion of this paper, also mention about the future works.

II. RELATED WORK

There are many vendors have introduced hardware devices and controllers that have the ability to support OpenFlow. The development of the applications and services of the upper layer of SDN controller are also being taken seriously increasingly, one of which is the breakthrough in network security. The security detection methods to capture packets in accordance with switch port entity in a traditional network are difficult to implement in a large cloud data center network because of its high cost. SDN network can do more fine-grained control. The administrators can set policies layered to do filter and to do capturing and monitoring only for certain connection [13]. SDN rely on its characteristics of data plane and control plane separation and advantages of programmable network configuration that shows superior performance in many respects. There are many studies propose network security mechanisms based on SDN architecture. Strategy contains a moving target defense (MTD) is to establish an attribute diverse and constantly changing network system which is operating by external opinion that allows hackers before the attack is difficult to detect targets, thereby increasing difficulty reaching attacks and defense. Random variation can attribute contains IP address, port number, routing path, host identity, instruction set and so on [14]. To compare with traditional network, the characteristics of SDN that separation of control and forwarding make it easier to implement such a network system like that. Furthermore, SDN switch can match several fields of packet header according to flow table in the design of self-defense network system. To decide whether to forward the packet or do other disposal such as speed limits by determine the source and destination IP addresses, MAC addresses, port numbers and so on [15]. Thus can built the firewall access lists (ACL) functions which lists of bad actors provided by security analysis or anomaly detection systems that monitor for suspicious activity and some basic security mechanisms into the network, and significantly enhance the network of self-defense capability. For example, SDN switch can check the validity of source IP address or to set the packet capacity threshold in a specific time for the packet with specific source or destination address, and to discard the packets or clean the forwarding traffic when time out to withstand the threat such as the flooding attack [16].

SDN is able to make contingency measures quickly when the network is being attacked, which is an important key feature of SDN. This is because SDN controller can execute scripts and quickly change the switch port configuration, which is more flexible and has faster ability to resist strain than traditional network. SDN can build such as the out of path deployment to do the mitigation from distributed denial-of- service (DDoS) attacks, or even to suppress DDoS attacks launched by botnets through the detection and protection system [16]. In SDN switch returns network status through the OpenFlow protocol so that the controller can immediately detect DDoS attacks. Then immediately modify the flow table and import into the traffic flow cleaning system. Unlike traditional network, the same mitigation mechanism or flow based attack monitoring can be start completed within a few time.

In many of defense systems or mechanisms based on SDN architecture mostly concentrated in a few key design, including attack detection, attack defense, attack mitigation, load balancing, traffic filtering and source tracking of attack. In the past, in order to trace and record the full path from the real source of the attacker by IP trace back while being attacked, it might need to add some fields for recording the path from the attacker to the victim. And when the path is very long, packet size will be much bigger than the packet only with pure information, because the header records whole paths information.

This way will bring to a great overhead, and lead to the significant reducing of the performance of networks. In response to these bottlenecks, the rise of new network architecture, SDN, who's feature is just to alleviate this problem, can do the resist and blockade action to achieve the purpose of defense, or even to counterattack. To trace the real source of the attacker by through a special packet forwarding ways, not only can mitigate the attack, but also find the exact position of the attacker.

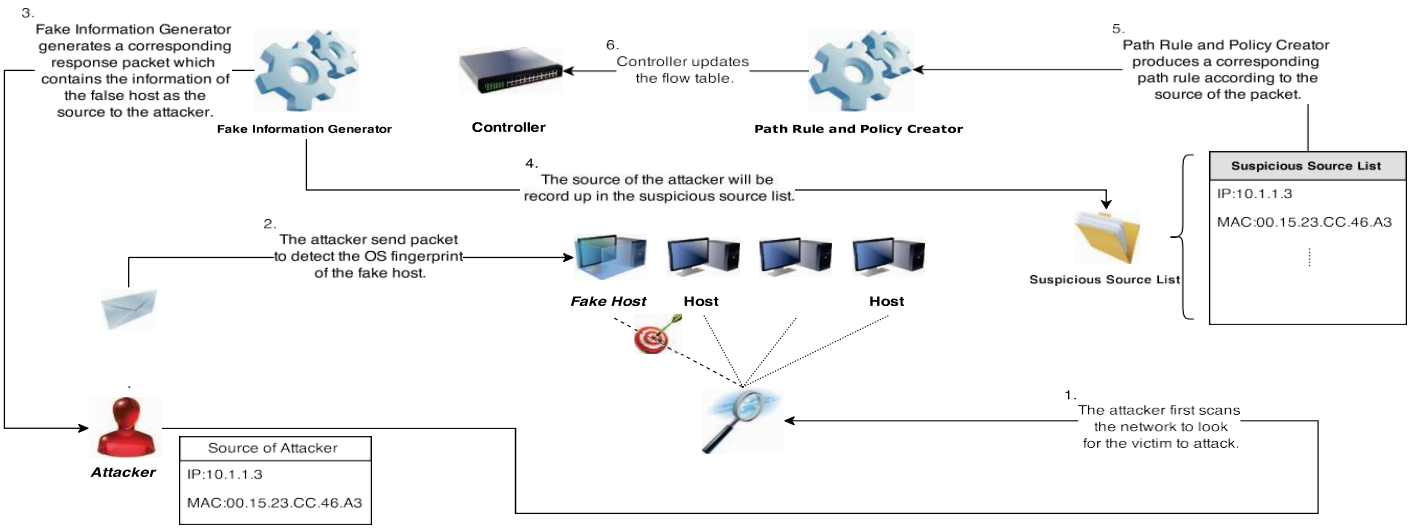

Fig. 1. The schematic diagram of the process of proposed method.

III. PROPOSED METHOD

Cyber attacks usually follow a behavioral model, including investigation, attack and invade, quit three stages. Cyber attacks are generally started from the targeting, information gathering, and then doing the vulnerability analysis to the target system. An attacker can search for objects of attack by through the port scanning, services scanning and a series of scanning tools. After the attacker has obtain the information such as the service of the port and the address of the service, the attacker will lock a certain object and launch a series of attacks behavior.

The proposed method of this paper assumes that the attacker will use the scan tools to reconnoiter the network behaviors before doing attack operation. It is easily and fast to learn the services performed on the remote host, and can even guess operating system and version of the remote host via scan tools. It can also perform the scanning on the subnet and to detect hosts exist on the subnet and to detect the services on them. The schematic diagram of the process has been described in Fig. 1, and the architectural of the proposed method contains several major components; it contains a controller, the path rule and policy creator, fake information generator and so on.

A. Attack steps and order

The first step of the attack behavior is to detect the operating system of the target host, while the attacker is scanning the network to look for the IP address of the victims on the network, it scans on the fake host that was generated by the fake information generator and regard it as the target. Sinceevery operating system has different ways to handle packet, so detection software can take advantage of this feature to identify the different operating systems. This way is like human fingerprints, as it is also known as the operating system fingerprinting identification.

The operating system detection can be divided into some different types, the first type of detection software will detected via transmitting the specially built TCP, UPD or ICMP packets to the host side, and then determine operating system of the host according to detecting the feature of the packet the host backhaul over. Another type of the detection software does not take the initiative to transmit the packet to the detected host, but determine the operating system used by the host by monitoring the features of exchanges packets passing through the network.

After determining the operating system of the host, the next step is to confirm the information of the services operating on the host in order to study out attack strategies. Detection of the services on the host is to determine whether the port of the host is open by analysis the response of the host to the packets. At this stage, the whole connection detection methods using the normal connection, and detection software confirms whether the port of the detected host is in the open state by using the completion of the three-way handshake with the detected host. Once the detection software detected that the host has successfully completed the three-way handshake and had complete the connection with the target host, it can learn that the state of the port is open, and vice versa, which represents the detected port is now in the closed state. Such scanning manner that is known as a full connection scan, however, due to the scanning use the whole connection which is very easy to be recorded by a firewall or IDS, so most port detection technology generally will not complete the normal connection to avoid being recorded. The detection techniques used usually included TCP connect scanning, TCP SYN scanning, TCP FIN scanning and so on.

B. Defensive and resist approach

Algorithm 1 Fake Host

```
Start timer;
WHILE ( timer < ∅ )
    IF ( packet_in.dst_ip == fake_host.ip OR
         packet_in.dst_mac == fake_host.mac )
         THEN counter ++;
    END IF
    IF ( counter >= k )
         THEN add packet_in.info into suspicious_src_list;
    END IF
    IF ( count_lsit >= θ )
         THEN
             create flow_entry according to suspicious_src_list;
    END IF
END WHILE
```

As algorithm1 shows, in the proposed method, the "fake information generator" will randomly generate m "false hosts" in the initial stage of network setup. Once the attacker sends packets through the detection tools to more than k "false hosts" that he thought they were really exist but actually were generated by the "fake information generator", the source that the attacker sent the packet to the fake hosts will be record up in the "suspicious source list". For a limited time ∅, if the same source of the attack has repeatedly been recorded in the "suspicious source list" more than θ times, then the "path rule and policy creator" would be issued to produce a corresponding path rule according to the source of the packet. At the same time, the "fake information generator" will generate a corresponding response packet that contains the information of the "false host" as the source to the attacker, but do not wait for the complete connection establishment is complete. The purpose of this is to let the attacker cannot notice that his attempt has been seen through.

Algorithm 2 Fake Port

```
Start timer;
WHILE ( timer < φ )
    IF ( packet_in.port_num == fake_host.port_num )
   /*attacker want to establish TCP connection with fake host*/
         THEN counter ++;
    END IF
    IF ( counter >= τ )
         THEN add packet_in.info into suspicious_src_list;
    END IF
END WHILE
```

However, if the attacker does not send packets to the "fake host", but perform the port scanning to detect the services working on one of the real hosts, and look for the weaknesses of which port is open but not any service work on it to attack. As algorithm2 show, at this moment, the information of the n "fake ports" that was generated by the "false information generator" has already been deployed and placed in the response message in response to the attacker, who has sent over the detection packet. Once the attacker lock on the "fake port", which state is open, into the point of the attack, and want to establish connection with it, the action will trigger the defense system. Within a specific time φ, once more than r "fake ports" are requested to establish connection with the attacker, the source of the attacker that who wants to establish the connection with the "fake ports" will be recorded into the "suspicious source list", and do not even wait the connection established complete. At the same time, the "path rule and policy creator" will be issued to produce the corresponding path rule to packets from these sources.

C. Architecture of defense mechanisms

In the assumption of the proposed method, we let the attacker always use the same address as its source. After all, if an attacker constantly changing its address to hide its source and performs attack, this will only cause the attacker to spend a mount of resources, but cannot achieve an effective attack. Therefore, the source of the attacker in the implementation and experiments of the proposed method is also fixed at the same IP address. The source code of the proposed method in this paper is briefly described in below. In the source code, function Fake_Information_Generator plays a very important role in whole operation of the defense mechanism. It uses the return value of function time() as the seed to generate fake information with random numbers, then sets a trap with the fake information generated by it and determines who might be the attacker. Suspicious_Source_List uses structure as its data type and consist of the information that has been regarded as the sources of the attackers. Finally, still the very important part is the function Path_Rule and Policy_Creator, which creates path rules and routing policy according to summarizing the "suspicious source list". It adds flow entries by through the standard of OpenFlow protocol, and effectively resists the damage from the attackers. The overall architecture of the defense systems in proposed method is design from the perspective of an attacker, which includes the whole process from the preparatory action of the attack to the actual execution of the attack.

IV. IMPLEMENTATION

In this section, we demonstrate the simulation to present our approach, and work in an environment with several physical entities. As shown in Fig. 2, the environment of the experiment consists of a Pica8 switch P-3290 (OpenFlow-supported SDN switch) with an operation system PicOS version 2.3.4 and three PCs as the real hosts, several real ports with some services are open on Real Host1.

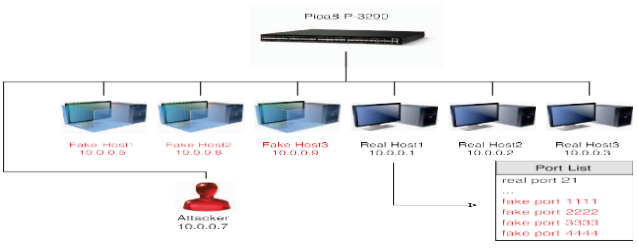

Fig. 2. The topology of the experiment environment in the implementation.

In this scenario, the "fake information generator" randomly generates a=3 "fake hosts", and let the source of the attacker be record up in the "suspicious source list" if it sends packets through the detection tools to more than a=2 "false hosts". Then if the same source has been repeatedly recorded in the "suspicious source list" more than a=3 times within the time limit Ø=10 milliseconds, the "path rule and policy creator" will add a corresponding flow entry to deny all incoming packets from this source. Assume that the IP address of the attacker is always 10.0.0.7. The attacker had scanned both the real hosts and the "fake hosts" on the network by using the scanning tool and wanted to ping all the hosts on the network. As shown in Fig. 3, the attacker can connect with all host on the network at first time. Then the attacker received the ACK packet with fake IP and MAC address of the Fake Host1 while it sent ping packet to the Fake Host1. After it did the same thing to Fake Host2, its IP address immediately been recorded. The whole process repeated three times and took about 7.8 milliseconds. After that, the attacker cannot connect to any host on the network as shown in Fig. 3, all the packets with source IP address 10.0.0.7 will be dropped because of adding the corresponding flow entry.

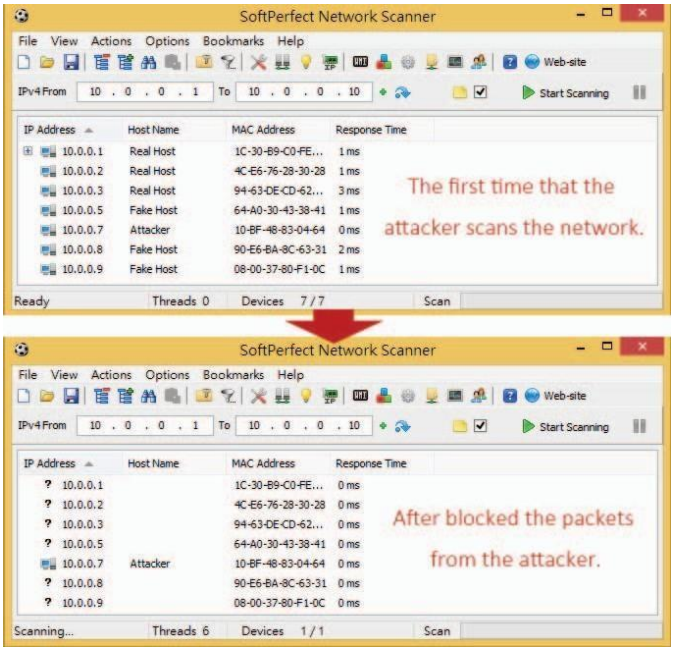

Fig. 3. The process that the attacker being blocked by the defense mechanism

In the other scenario, as shown in Fig. 4, the "fake information generator" generates □=4 "fake ports" initially in Real Host1 which is using the IP address of 10.0.0.1, and let the specific limit time □=10 milliseconds. The attacker uses port scan tools to find out ports that stay in the open state, and try to connect with each of them. Then set a=1, that is, if any "fake port" has been requested to establish a connection with it, the source that sent the request to the "fake port" will be recorded, and all incoming packets from this source will been deny because of the adding of the corresponding flow entry.

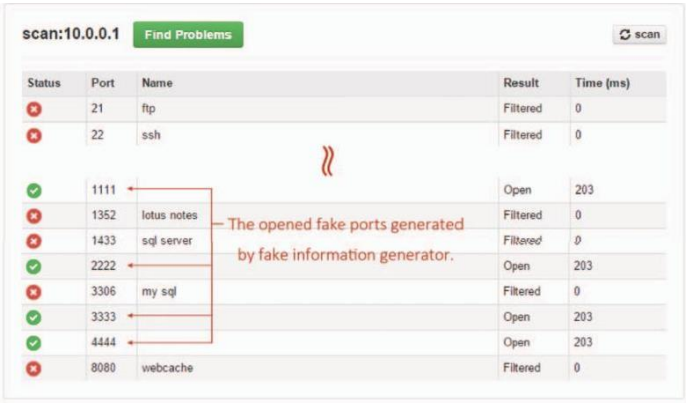

Fig. 4. The port scan result performed by the attacker.

In the implementation of the proposed method, the source IP address that the attacker used has been blocked successfully because of the attack actions detected by the defense mechanisms.

V. CONCLUSION

In this paper, the proposed method let attackers thought that there exist some of the objects that they can attack on the Internet by through the fake information shown by "fake information generator", but in fact the target they detected are the virtual bait designed as the trap to lure the potential attackers. Use the fake hosts to deceive the attackers fooled while they operating the investigation before they perform the attack. Once the attackers send packets to the fake hosts or try to establish the connection with fake hosts, they hooked! Then the sources of the attackers will be recorded into the "suspicious source list", and the "path rule and policy creator" will use the information on it to create the corresponding path rules. This is all in order to protect the network to suffer from attacks. Through rigorous simulation and experiment, the proposed method has demonstrated a satisfactory performance. The result shows that the proposed method can indeed prevent attacks from occurring, and can effectively enhance the defense capability. In future works, we hope to plus more elements to our approach such as the solution for the situation that when encountered a large number of attacks against sudden temporary. The mention of the defense mechanism, we also look forward to be able to put the capabilities that to trace the source of the attacker by making use of the characteristics and ability of SDN in our research projects. How to reinforced and enhance the effectiveness of the proposed method is one of the main research directions in the future, and we will be committed to expand our approach.

REFERENCES

[1] "Software- Defined Networking: The New Norm for Networks", Open Networking Foundation, April 2012.

[2] Gangadhar S, Sowmya. “The Real Time Environmental Time Series Data Analysis Using Influx DB”. International Journal of Advanced Scientific Innovation, vol. 1, no. 1, Dec. 2020, doi:10.5281/zenodo.4641703

[3] Girish L, Rao SKN (2016) Mathematical tools and methods for analysis of SDN: a comprehensive survey. In 2nd international conference on contemporary computing and informatics (IC3I), Noida, pp 718–724.https://doi.org/10.1109/IC3I.2016.7918055

[4] Shambulingappa H S. “Crude Oil Price Forecasting Using Machine Learning”. International Journal of Advanced Scientific Innovation, vol. 1, no. 1, Mar. 2021, doi:10.5281/zenodo.4641697.

[5] Y. Yu, C. Qian and X. Li, "Distributed and collaborative traffic monitoring in software defined networks", Proc. 3rd Workshop Hot Topics Softw. Defined Netw, pp. 85-90, 2014.

[6] F. Giroire, J. Moulierac and T.K. Phan, "Optimizing rule placement in software-defined networks for energy-aware routing", Global Communications Conference (GLOBECOM) 2014, pp. 2523-2529, 8-12 Dec. 2014.

[7] Girish L, Rao SKN (2020) “Quantifying sensitivity and performance degradation of virtual machines using machine learning.”, Journal of Computational and Theoretical Nanoscience, Volume 17, Numbers 9-10, September/October 2020,pp. 4055-060(6) https://doi.org/10.1166/jctn.2020.9019

[8] Eun-Do Kim, Seung-Ik Lee, Yunchul Choi, Myung-Ki Shin and Hyoung-Jun Kim, "A flow entry management scheme for reducing controller overhead", Advanced Communication Technology (ICACT) 2014 16th International Conference on, pp. 754-757, 16-19 Feb. 2014.

[9] Girish, L., Rao, S.K.N. Anomaly detection in cloud environment using artificial intelligence techniques. Computing. https://doi.org/10.1007/s00607-021-00941-x.

[10] P. Shivam, A. Demberel, P. Gunda, D. Irwin, L. Grit, Automated and On-Demand Provisioning of Virtual Machines for Database Applications, In ACM Proc. Int. Conf. Of Management of Data (SIGMOD 2007) pp. 1079-1081, 2007.

[11] Nayana, Y., Justin Gopinath, and L. Girish. "DDoS Mitigation using Software Defined Network." International Journal of Engineering Trends and Technology (IJETT) 24.5 (2015): 258-264.

[12] Rashmi T V. “Predicting the System Failures Using Machine Learning Algorithms”. International Journal of Advanced Scientific Innovation, vol. 1, no. 1, Dec. 2020, doi:10.5281/zenodo.4641686.