Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.040626
Article

Towards Generating a Practical SUNBURST Attack Dataset for Network Attack Detection

Ehab AlMasri[1], Mouhammd Alkasassbeh[1] and Amjad Aldweesh[2,*]

[1]Princess Summaya University for Technology, Amman, Jordan
[2]College of Computing and IT, Shaqra University, Shaqra, Saudi Arabia
*Corresponding Author: Amjad Aldweesh. Email: a.aldweesh@su.edu.sa
Received: 25 March 2023; Accepted: 13 June 2023; Published: 28 July 2023

Abstract: Supply chain attacks, exemplified by the SUNBURST attack utilizing SolarWinds Orion updates, pose a growing cybersecurity threat to entities worldwide. However, the need for suitable datasets for detecting and anticipating SUNBURST attacks is a significant challenge. We present a novel dataset collected using a unique network traffic data collection methodology to address this gap. Our study aims to enhance intrusion detection and prevention systems by understanding SUNBURST attack features. We construct realistic attack scenarios by combining relevant data and attack indicators. The dataset is validated with the J48 machine learning algorithm, achieving an average F-Measure of 87.7%. Our significant contribution is the practical SUNBURST attack dataset, enabling better prevention and mitigation strategies. It is a valuable resource for researchers and practitioners to enhance supply chain attack defenses. In conclusion, our research provides a concise and focused SUNBURST attack dataset, facilitating improved intrusion detection and prevention systems.

Keywords: SolarWinds orion software; supply-chain-attack; SUNBURST-attack; solar gate; UNC2452

1 Introduction

The world stands on the threshold of a new paradigm shift that will change human behavior, social relations, the cognitive perspective of humanity, and its relationship with nature and the universe. Whereas the Fourth Industrial Revolution relies on using cyber-physical systems and the electronics, computers, information technology, and the Internet produced by the Fourth Industrial Revolution to automate production by machine without human intervention [1].

With the rapid development of technology, the world will witness drastic changes in today's society and life in the future. It has already witnessed how computing systems, sensors, artificial intelligence, and genetics can reshape entire industries and the structure of our daily lives. Moreover, we are witnessing such rapid change as the technology works hard to impose new rules that will change our lives; many of the old assumptions that were made no longer hold [2].

Software is increasingly essential to modern life. Software has become an integral and widespread part of contemporary life. Whether it was the cloud computing that powers email service, the rollout of 5G telecommunications or the system used to keep tabs on an offshore oil rig. One observer summed up the trend by saying, "Software is consuming the globe." [3]. Software is always ongoing, unlike physical systems. It requires regular updates and patches to fix security holes and enhance its functionality. Due to this upkeep, software supply chains are lengthy, disorganized, and constantly changing, posing an enormous and often unrecognized aggregated risk for businesses worldwide. Supply-chain security policy discussions have mainly focused on hardware, despite prominent security community members' warnings and the greater attention they give to supply-chain security in general [4].

Software and system makers offer supply chains to support their products with essential updates, and the role of clients or institutions working on these systems is to provide these companies with the rights to access and upgrade their systems. Application developers all around the globe are turning to supply chains to make it easier for them to get the necessary updates and upgrades.

A cyberattack that aims to destroy a company by exploiting security flaws in its supply chain is called a supply chain attack. Supply chain attacks [5] may happen in any business, from banking and energy to the public sector [6,7]. Software or hardware may be attacked through the supply chain. A common tactic used by cybercriminals is to embed malware or hardware-based surveillance components into the production or distribution chain [8]. In 2018, supply chain threats grew by 78%, according to 'Symantec's 2019 Internet Security Threat Report' [9]. The authors in [10] create proof-of-concept malware that takes advantage of users' confidence and Google's restrictions to bypass popular voice search apps. In [10], findings reveal that attackers may overcome Play Protect by first uploading innocent programs to gain confidence, then slowly adding malicious feature upgrades to propagate extremely intrusive malware into user PCs. The paper argues that this attack method is a supply chain attack, as it involves compromising a legitimate app and exploiting its distribution to reach unsuspecting users.

The SUNBURST attack, also called SolarWinds, Solorigate, and UNC2452, was among the United States government's most significant and most recent cyberattacks. The name SolarWinds is synonymous with high-quality IT services. The company originated in 1999 and has its headquarters in Austin, Texas, USA. SolarWinds focuses on creating software for giant corporations like Microsoft and CISCO. Additionally, it offers software services to US government organizations such as the Department of Defense, the Department of Homeland Security, and the National Nuclear Security Administration. As a result, the attack devastated thousands, and even millions, of individuals [11].

There are several motivations for launching a cyberattack. Examples include any effort to tamper with a computer, network, or system; access or steal private information; delete or leak data; or obtain unauthorized access to data [12]. Malware, phishing, ransomware, denial of service (DoS), man in the middle, crypto-jacking, SQL injection, and zero-day vulnerabilities are just a few examples of cyberattacks. One way to categorize these attacks is by the component of the system that is affected or changed.

Malicious software, or malware, is any program with malicious intent. A few examples of malware include computer viruses, worms, and trojan horses. So, when discussing any cyberattack, the word 'malware' tends to be the most often used. According to investigators, the attack on SolarWinds was a supply chain attack. This indicates that a large portion of the attack was directed at a chain of companies through which weak security points were exploited. The SolarWinds attack was first uncovered by FireEye, a cybersecurity firm based in the United States specializing in identifying risks

and offering actionable solutions. Since then, new details have surfaced almost daily. In this study, we will examine the SolarWinds attack from all angles.

The group of attackers behind this is known as UNC2452. FireEye says that SUNBURST malware was spread with the help of trojanized SolarWinds Orion business security updates [13]. The attacker's post-compromise operation uses various methods to keep themselves from being found out. The movement influences both public and private organizations all over the world.

Since FireEye's initial release on December 13, 2020, further information regarding the SUNBURST backdoor has been uncovered. SUNBURST is a trojanized version of the SolarWinds Orion addon, which is digitally signed. The SUNBURST is a trojanized clone of the 'SolarWinds.Orion.Core.BusinessLayer.dll' [13]. A backdoor in the plugin that interacts with third-party servers using HTTP. SUNBURST can retrieve and execute commands that instruct the backdoor to migrate files, execute files, profile the machine, reboot the system, and disable system services after an initial dormant duration of up to two weeks [14]. By imitating the Orion Improvement Program (OIP) protocol, the malware makes its network traffic look like normal SolarWinds operations. Data about the malware's persistent state is stored in regular SolarWinds plugin configuration files [15].

The campaign's actors received access to a variety of public and private organizations around the world. They obtain access to victims via trojanized upgrades to SolarWinds Orion IT monitoring and management tools. This campaign may have started as early as Spring 2020 and continued until December 2020. Lateral transfer and data leakage have been reported as part of the post-compromise operation surrounding this supply chain breach [16].

In the case of a SUNBURST attack, in particular, no dataset can enable users to infer the features impacted by SUNBURST. Hence, intrusion detection systems (IDS) and intrusion prevention systems (IPS) cannot be used to detect or anticipate the presence of a SUNBURST attack on the network. By creating a real dataset with SUNBURST's affected features, one can help detect this attack in the future. After gathering and reviewing all past research from the time the SUNBURST attack was discovered in December 2020 until now, the study did not find any dataset on SUNBURST that can be used to study and detect the SUNBURST attack, hence the importance of the study.

Developing a new dataset that can analyze SUNBURST attacks and their behavior in networks could provide advanced warning to cybersecurity professionals about potential future attacks. Researchers and the cybersecurity industry could develop better preventative measures by understanding how SUNBURST attacks operate and the damage they may cause. The study's main contribution is the creation of a novel dataset (named SUNBURST) that has yet to be previously studied. The dataset was meticulously processed, tested, converted, and cleaned, and its accuracy was verified using machine learning techniques. The SUNBURST dataset can be applied to gain a deeper understanding of SUNBURST attack dynamics, which can be used to prevent future attacks.

The primary aim of this research is to gather real-world data and validate a dataset that can be utilized by machine learning and deep learning algorithms as a raw dataset instead of focusing on identifying the valuable features of specific datasets. Specifically, our model examines the attack on the client machine and the activation of malware on the device but does not address how the attack infiltrated the machine. This limitation in the scope of our research presents an opportunity for further exploration and improvement in future studies.

We proceed as follows: Section 2, which provides an overview of the security incident, including an introduction to SUNBURST and a walkthrough of the methods used and access gained by responsible parties; Next, in Section 3, literature review, we review previous works. Section 4, "Methodology, "

explains our methodology, how we build our dataset, and the machine learning we use to validate our dataset; Section 5, "Results and Discussion, and Section 6, " Conclusion.

2 Overview

2.1 Supply Chain Attack

While threats from attacks on the supply chain have been there for some time, there has been an increase in well-planned attacks against organizations after 2020. Because of the improved security measures taken by businesses, attackers may have successfully turned their focus to suppliers. They were able to cause severe disruptions in many ways, including system outages, financial losses, and brand harm. Successful attacks may significantly affect many consumers who rely on the impacted provider; that is why supply chains are so crucial. Therefore, the flowing effects from a single attack may have a broadly distributed impact.

With the SolarWinds attack, the world saw the full destructive and rippling impact of supply chain attacks [17,18]. SolarWinds is recognized as one of the most severe supply chain attacks in recent years, particularly given the targeted entities, which included government agencies and major corporations. It sparked policy measures worldwide and was widely covered in the media. For instance, the Kaseya4 incident in July 2021 highlighted the necessity for further and focused attention to supply chain attacks impacting managed service providers. The number of attacks on supply chains, like the one in the example above, has been rising rapidly over the last year. More than ever, it is imperative that governments and the security community work together to develop and implement innovative safeguards to prevent and lessen the effects of future supply chain attacks.

A supply chain attack is the result of at least two attacks. As shown in Fig. 1. The first is on a supplier, which is then utilized to attack the target to access its assets. The intended receiver might be the end-user or a middleman provider. As a result, for an attack to be supply-chain related, both the supplier and the end customer must be affected.

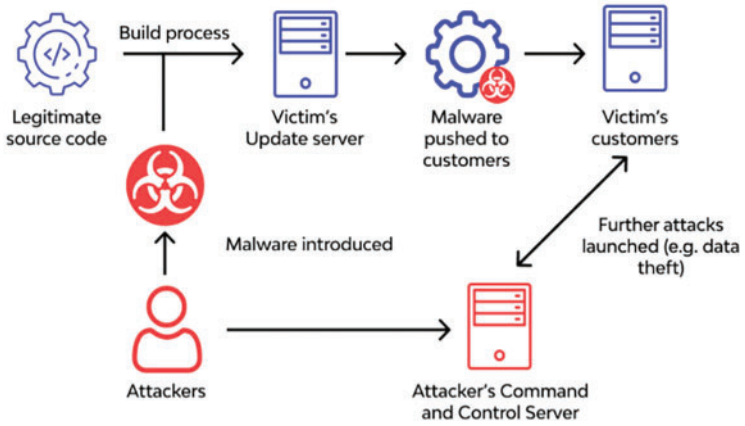

Figure 1: Supply-chain-attacks [19]

Attacks against a company's supply chain aim to exploit the accumulated trust between that company and a small group of its suppliers and vendors. Organizations rely on a broad range of tools created by a wide range of enterprises because they employ third-party software for many diverse purposes, including communication, meetings, and the rollout of websites.

2.2 Advanced Persistent Threats (APT)

As technology developed, advanced persistent threats (APT) emerged as a significant challenge for the stability of cyberspace on a worldwide scale. Attitudes regarding cyber operations have become a topic of controversy as more and more nations use them to exert influence on the policies of other states. Compared to other issues, cybersecurity in the 21st century has received much attention from both the public and governments. Among the many dangers in cyberspace today, advanced persistent threats remain among the highest.

General Services Administration (GSA), an independent body of the United States federal government, defines APT groups as those who lead cyber espionage or cyber sabotage against a country's most valuable information assets. To participate in espionage, acquire intelligence, or degrade the target's capabilities, APT hackers are more sophisticated and organized than traditional hackers. A few of them work for powerful institutions like states [20]. According to the Swiss Cyber Institute, APT groups develop a thorough strategy to achieve a more significant objective than simple incidental attacks. As a result, they can successfully target strategic individuals and information to gain access to intellectual property, state or military secrets, computer science code, or other valuable information. The term 'Advanced Persistent Threat' (APT) is used by the Cybersecurity industry to describe organizations that employ sophisticated and ongoing hacking techniques to gain continued access to a target system.

According to Kaspersky, APTs are premeditated hacks that involve multi-staged campaigns against high-value targets. The goal of APT organizations is not just to disrupt networks or steal data; they also want to exfiltrate information. Governments and cybersecurity firms agree that APT organizations are defined using a combination of advanced tactics and the persistence of their operations over time. There is no limitation on whether APT groups can target state or non-state actors. Nonetheless, the GSA's classification of APT groups as entities conducting attacks on vital information is connected to 'national security' or 'strategic economy.' Also, cybersecurity does not require the information to be secret or country specific. The worth of the data that APT groups are after depends on how its owner values it.

2.3 SUNBURST Attack

The SolarWinds Orion (SUNBURST) cyberattack is the most severe event to have hit the United States. Unprecedented in scale and complexity, the attack penetrated various corporate and administrative entities for months without being discovered [21]. Five days after cybersecurity incident response firm FireEye discovered a breach in their network and the theft of intelligent software tools [22], the business disclosed the breach in December 2020. Due to this alarming occurrence, it was found that malware was being distributed through trojanized upgrades of SolarWinds' Orion Platform software [23].

Microsoft, Intel, Nvidia, and the US Departments of Homeland Security, State, Commerce, and the Treasury are some IT professionals and government agencies that utilize SolarWinds' Orion Platform [24]. Around 18,000 customer networks were delivered malicious malware after the Orion Platform was updated in March 2020 [23]. Conspiracy theorists believe the incident was state-sponsored spying [24]. Despite official denials of culpability, this "state-sponsored" act is still being seen as a breach of state sovereignty.

By gathering intel on SolarWinds and its customers, threat actors were able to get into the SolarWinds Orion Platform and steal sensitive data. With the compromised credentials, the threat actors could enter the SolarWinds network while masquerading as legitimate users. After gaining

administrative credentials, agents infiltrated the Orion Platform's SDLC, where they implanted malicious code that would be run during the build process of the most recent upgrade. When the update was installed on client networks, it compromised those devices, opening a backdoor for the threat agents to use them as a conduit to their command-and-control nodes. Terrorists accessed victims' networks using their servers, stole data, distributed malware, and issued remote instructions.

The widespread penetration of SUNBURST into the nation's governmental structures is sure to have repercussions for the people of that country. Personal information theft, credit card fraud, etc., can potentially ruin people's lives and reputations. People risk being physically harmed when they get access to the codes that run machines. Examining how a breach would affect the parent firm is crucial. A breach may have catastrophic effects on a company's finances and image, posing a danger to the company's very survival. Orion Platform generates around 45% of SolarWinds' overall revenue, or $343M.

2.4 Who are the Attackers

On April 27, 2022, Mandiant announced that it had collected enough information to conclude that APT29 is responsible for the SolarWinds hack tracked under the group UNC2452 in December 2020 [16]. According to this analysis, APT29, an espionage organization in Russia believed to be funded by the Russian Foreign Intelligence Service, was responsible for the infiltration of SolarWinds' supply chain [16]. The assessment in our study is based on information acquired by Mandiant and is the product of a thorough comparison and examination of UNC2452 and in-depth familiarity with APT29.

Merging UNC2452 into APT29 has dramatically increased our understanding of this group, revealing an ever-evolving, disciplined, and highly competent threat actor that maintains high operational security (OPSEC) while gathering information.

Advanced Techniques, Advanced Persistent Threat 29 (APT29), is a highly sophisticated organization that has continuously improved its operational and behavioral tactics, methods, and procedures (i.e., Tactics, Techniques, and Procedures (TTPs)) to conceal activities better and restrict its digital footprint to escape detection. The consolidation raises awareness of APT29's capabilities, including its extensive resources, lengthy history of activity, extensive knowledge of its targets, strict adherence to operational secrecy, ability to adapt to new environments quickly, and stealthy operation. As new technologies arise, the business has consistently improved its TTPs and implemented new procedures.

2.5 Timeline of Cyber-Attacks

SolarWinds is investigating the attack's origins with the help of legal counsel from DLA Piper, security researchers from CrowdStrike, forensic accountants from KPMG, and other experts in the field. It is part of investigating how the SUNBURST malware was introduced and spread among SolarWinds Orion Platform applications. They claim they have discovered the source the attackers used to inject the SUNBURST malware into versions of their Orion Platform software. As SolarWinds mentioned, this source is incredibly clever and innovative, as displayed in [25].

SolarWinds forensic specialists have so far discovered suspicious behavior on SolarWinds' internal systems dating back to September 2019, hence recommending a chronology that starts then, as shown in Fig. 2. The following Orion Platform release in October 2019 seems to have been tampered with to probe the criminals' ability to inject code into SolarWinds builds. The revised source code for injecting the SUNBURST malware into the Orion Platform will be released on February 20, 2020.

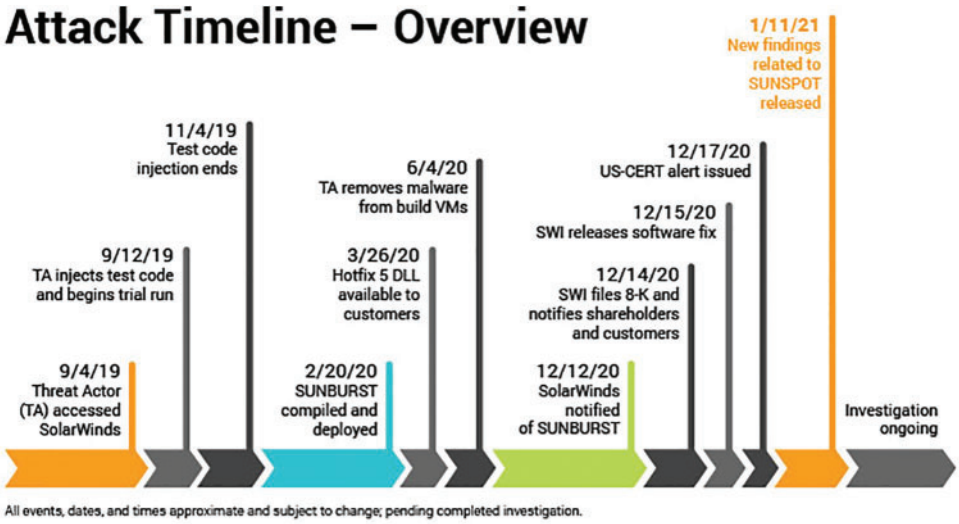

Figure 2: SUNBURST Attack Timeline [25]

Criminals removed the SUNBURST malware from the SolarWinds infrastructure in June 2020 without being noticed. Meanwhile, SolarWinds has been looking at potential holes in the security of its Orion Platform. It either fixed or started fixing the vulnerabilities, which is an ongoing practice. However, the business discovered SUNBURST vulnerabilities in December 2020. After learning of the attack on December 12, 2020, SolarWinds took fast action to alert and safeguard SolarWinds customers and investigate the attack in conjunction with law enforcement, intelligence, and government agencies.

3 Literature Review

In [26–28] set out to investigate machine learning and deep learning systems for identifying attack data packets in a network, the focus of recent studies. In addition, existing machine learning algorithms can only recognize previously identified risks. While the number of cyberattacks and zero-day attacks continues to rise, the ability to exist algorithms to identify previously unseen threats is becoming more limited. The study concentrates on discovering uncommon attacks using transfer learning from an existing dataset of known attacks. Deep learning achieves at least 21% better results for the given dataset than explicit statistical modeling techniques.

In [29], to categorize attack data packets with 99.65% accuracy, a Convolutional Neural Network (CNN) architecture has been widely proposed after a preliminary study of possible deep learning architectures and transferability testing. To gauge transferability, the proposed CNN architecture underwent training with a known attack and subsequently tested with unknown threats. This model requires additional information in the training samples to extract helpful information for generalization. Current threat information only accounts for 20% of the data collection. Training on novel synthetic datasets is one method, while bootstrapped dataset training is another that has been created to deal with limited data. To maximize one's capacity for learning, one must first choose an optimal selection of training attacks. Results from this research reveal that there are training-testing attack combinations that facilitate the effective transfer of knowledge. DoS attack training-testing pairs have the most robust and consistent connections. Some speculations on generalization from the models are

also presented in this investigation. The Recursive Feature Elimination (RFE) technique was used to examine the dataset's features and the attacks' characteristics to verify the findings.

Reference [30] provides a GAN-based system to identify abnormalities (GANs). Time-series reconstruction is used. Reconstruction errors were calculated using cosine similarity. They have proposed two data encoding methods. The GAN model trained using both encoding approaches performed equally in our experiments.

Their firewall, Fortigate, provided the data. Four million log messages were in the dataset. This dataset, text messages only, is 80% training data. 20% is testing data. This project aims to find abnormal Fortigate firewall log messages. Abnormal communication may be harmful. Fortigate firewall network traffic logs including several fields. Reference [30] provides a GAN-based system to identify abnormalities (GANs). Reconstruction errors were calculated using cosine similarity. The researchers proposed two data encoding methods. The GAN model trained using both encoding approaches performed equally in our experiments.

Malware detection is a large-scale data-mining challenge in SIEM systems, according to [31]. They suggest analyzing executable/process activity such as file reads/writes, process creations, network connections, or registry updates to identify sophisticated stealthy malware. To discover outliers, they represent processes as directed acyclic graph streams. They achieve this by converting behavioral graph streams into documents, embedding them using a state-of-the-art Natural Language Processing model, and then conducting unique outlier identification on the documents' high-dimensional vector representation. They compare their method to a big multinational company's SIEM system (over 3 TB of EDR logs). The suggested technique detects unknown dangers.

The researchers tested behavior-based malware detection using EDR logs from an enterprise SIEM. Modeling process behavior as graph streams was suggested. Universal Sentence Encoder converts graph streams to documents. Running Isolation Forest in high-dimensional vector space detects abnormal documents and probable harmful occurrences.

The method was tested in a real-world situation next to a big company's SIEM system. The researchers found two additional instances, giving this technique an accuracy of 94%. While their technique may detect outliers, they recognize vast false positives since the abnormal activity is not always harmful.

Machine learning techniques to classify malware by family, as stated by [32], help analysts save time when responding to incidents. To be effective, these methods must choose characteristics resilient to idea drift, which explains the evolution of malware over time.

In today's cybersecurity landscape, detecting malware and preventing intrusions are crucial challenges. Security Information and Event Management (SIEM) systems are essential to monitor and safeguard networks effectively and ensuring successful malware detection is a significant concern.

References [32,33] propose deep learning-based botnet detection approaches. In [33], a deep neural network model is developed to predict botnet activities by analyzing network traffic patterns. The study demonstrates the model's effectiveness in accurately predicting botnet behaviors, contributing to botnet detection and prevention. Similarly, reference [33] introduces a deep neural network approach for distinguishing between regular and botnet traffic. By training the model on various network traffic features, DNNBot achieves high detection accuracy and robustness, making it a valuable tool for identifying and classifying botnet activities.

In [34], authors focus on developing an intrusion detection system (IDS) specifically designed for edge computing environments. The study proposes a Parallelized Convolutional Neural Network

(PCCNN) architecture to analyze network traffic and detect potential intrusions. By leveraging parallel computing techniques, the PCCNN-based IDS overcomes the challenges of edge computing and achieves efficient and accurate intrusion detection. This research contributes to the field by addressing the unique requirements of edge computing and providing an effective solution for detecting network intrusions in such environments.

In [35], authors propose a novel approach for insider threat detection by combining natural language processing (NLP) word embedding and machine learning techniques. The study analyzes textual data, such as emails or chat logs, to extract semantic representations of user behaviors. The proposed method effectively identifies potential insider threats by applying machine learning algorithms to these representations. The research highlights the effectiveness of leveraging NLP and machine learning in detecting insider threats and provides valuable insights into enhancing security measures against such risks.

Using Windows handles (such as files and registry keys) assesses a dynamic feature set for malware family categorization. In particular, they look at how susceptible the characteristics are to idea drift and how resilient they are to that phenomenon. The researchers assembled a new dataset to mimic the attacks that may be carried out on malware samples. Moreover, they show that, compared to conventional API call-based characteristics, their machine learning classifiers are more resistant to idea drift when tested on manipulated samples of malware that were acquired from the wild.

Additionally, they probe time deterioration because of idea drift using temporally consistent assessments that do not presuppose access to newer data. The testing demonstrates that our features can withstand the obfuscation techniques used by malware. Additionally, they provide empirical evidence of how malware naming practices (malware kind or family) might affect outcomes and give suggestions for dataset development. They assembled the corpus from many sources, including Variant [36] and Contagio2.

In [37] observed that people must examine and learn from their own experiences to find new ways of avoiding, detecting, and remediating these attacks. The SolarWinds Orion (SUNBURST) data breach may be the most significant data breach the United States has ever experienced. They were afraid that information stolen could be used in other criminal activities such as identity theft, credit card fraud, and harming the reputation of individuals. Access to programs controlling physical machinery can also harm individuals physically. According to the researchers, there were many stages to the attack procedure. However, there is insufficient evidence to establish the identity of the threat agents or actors responsible for the security breach.

Threat agents started their reconnaissance assignment by launching a Supply Chain Attack on a third-party client having access to SolarWinds resources rather than hacking SolarWinds' network. To avoid detection by security and antivirus software, the hackers disguised themselves as legitimate users to infiltrate the network and remain undetected. In this small-scale attack, threat agents discovered they could change SolarWinds' signed-and-sealed software code and publish it into a functional update using a short code fragment as a proof-of-concept. When they realized they could carry out a large-scale supply chain attack undetected by the intrusion detection systems, the threat agents were confident they could carry out the strike further.

The protocols used to interact between the Orion Platform and business servers were reverse engineered by threat agents. Hackers got around this by writing their code to replicate the syntax and structure of communication packets. The threat actors' external domains became active after the upgrade was installed and activated on client servers.

The hackers utilized the command-and-control server to interact with computers that had updated Orion product updates, giving them backdoor access to these systems. The threat agents could access all of SolarWinds' resources because they had devised a method of entering the network as an authorized identity. Additionally, this attack significantly affected the human, financial, and cybersecurity sectors. They suggested ways to prevent another attack like this in the future and recommended doing the following fixes: Remediation Actions, Patches, and Continual/Future Solutions.

On the other hand, prior research [38], 'SolarWinds Hack: In-Depth Analysis and Countermeasures,' has shown that SolarWinds' attack was characterized as a supply chain attack. In other words, some firms were utilized as entry points for a more significant attack because they had lax security measures. A broad review of supply chains is also included in the paper, as are the implications that victims face because of hacker attacks on these systems.

Interestingly, these researchers addressed how much money it would cost to get back to normal after the attack. SolarWinds plans to spend $25 million to enhance the security system and purchase cyber insurance. An additional $3 million was spent on the attack's aftermath in the fourth quarter. On the other hand, CRN calculated the cost based on two factors: indemnification for clients and the inquiry. SolarWinds will have to pay about $90 million in client indemnification and $312 million in investigative costs.

On the other hand, Insurance Business Magazine looked at insurance costs, forensic services, and incident response, estimating that the total would be roughly $90 million. A hundred billion dollars is expected to be spent on Roll Call software and system recovery. Forensic services can only be afforded by organizations with cyber insurance, according to BitSight's findings.

Moreover, other research on a co-evolutionary model of cyberattack patterns and mitigations using public datasets [39]. The researchers proposed a coevolving model of cyberattacks and mitigation strategies. In the cyber world, APTs (Persistent Advanced Attacks) are a type of cyber threat conducted by recognized groups of actors who are identifiable by the tactics, methods, and processes they routinely use. While standard cyber protections continue to be overcome by advanced persistent threats (APTs), community effort continually increases defenders' hands. Victims of advanced persistent threats (APTs) may engage in various forms of community action, determining who may have attacked them, how the damage was caused, and where the attack was directed. Victims who believe in the power of open communication open their ordeals to a broad audience via various government and private sector media sources. The security community is notified of possible dangers methodically assembled from attack reports in large repositories. Solarigate (SUNBURST) is a current and well-known example of an APT. They detected this worldwide hacking effort that targeted their networks and stole their intellectual property via a cybersecurity company called 'Fireye.'

There are several critical challenges connected to current cybersecurity regulations that the Great Eclipse of US cybersecurity case study is meant to throw light on [40]. The case study delves further into two contentious and potentially dire political issues. It is also worth noting that, despite earlier preparations by US government cybersecurity entities, all were focused on the 2020 presidential elections. It goes into detail on what happened during and after that SUNBURST attack, as well as how invaders were able to get their hands on official secrets.

This attack, which impacted more than 300,000 devices throughout the globe and was able to capture vast and sensitive data, was not discovered by the organizations responsible for cybersecurity, as the researcher details in his piece and blames APT29 for it.

Even when there has been much debate on SUNBURST and its global ramifications and the need for new approaches to cybersecurity, only some of the recommendations have been made public. Even though this method of attack has been recognized for over a decade, no company can be immune from it.

Since the date of the SUNBURST discovery, a search has been conducted of all scientific articles discussing any information for detecting dangerous programs or creating the dataset. In this research, we looked for a dataset that covers the SUNBURST attack and could not find any. No studies of the SUNBURST attack's behavior in operation on the network were present in any research we came across.

All investigations agreed that the SUNBURST attack was a worldwide one, with the United States bearing the effect of the damage. This attack was also categorized as a cyber war against the United States. The investigations also verified that the SUNBURST highlighted several flaws in the US cybersecurity system, prompting the US to spend more money to sustain the cybersecurity system and scientific research.

4 Methodology

This section will explain our methodology and practical steps for generating the SUNBURST dataset. This lab's primary objective is to record SUNBURST network anomaly traffic. One of our motivations for this research is the lack of a dataset that includes a SUNBURST attack, which, in turn, incentivizes us to create a unique dataset for this global attack and combine it with other datasets to support intrusion detection systems (IDS).

It is difficult to put a percentage on how much progress has been made in detecting malicious activity by (IDS). Training machine learning based IDSs requires appropriate datasets; however, securing a trustworthy sample for comparison is challenging. An essential component of machine learning-based IDS models is the dataset. The first step is to get the data from the traffic, either as a packet or a flow. Once the traffic has been recorded, it is assembled into a particular form of data comprising details connected to the network, such as labeling. For this dataset, the labeling procedure is essential. It is challenging to deal with real situations when specialists cannot tell whether incoming traffic is malicious. Therefore, labs rely on fake traffic for their experiments. However, this means that the simulated traffic does not accurately reflect the conditions in the actual world.

In a nutshell, a dataset is created by capturing traffic and concludes with the pre-processing step. A labeled dataset is the product of the pre-processing procedure. Every information is sorted into one of two categories: malicious or safe. Data tables are represented in the file either in a human-readable format (like a CSV) or a binary format (like an IDX). The amount of malicious or false alarms found determines the dataset's quality. Thus, we reviewed several datasets in IDS that were generated, studied the scientific methodology for their construction, highlighted many requirements [40–42]., and came up with a set of recommendations for creating datasets.

There is much overlap between the various specifications. While encrypted traffic in the dataset is emphasized as a key necessity, efforts have yet to do so. We separated the needs into two categories: content and procedure. The content requirements include a complete collection of network traces and a realistic capture of network activity, and the functional requirement concentrates on what is required to generate a dataset. The necessary components of the procedure need thorough documentation. While more is needed, there currently needs to be available data on how to create a new dataset that is

both useful and feasible to implement. We took into our consideration these requirements in Fig. 3 to create our dataset:

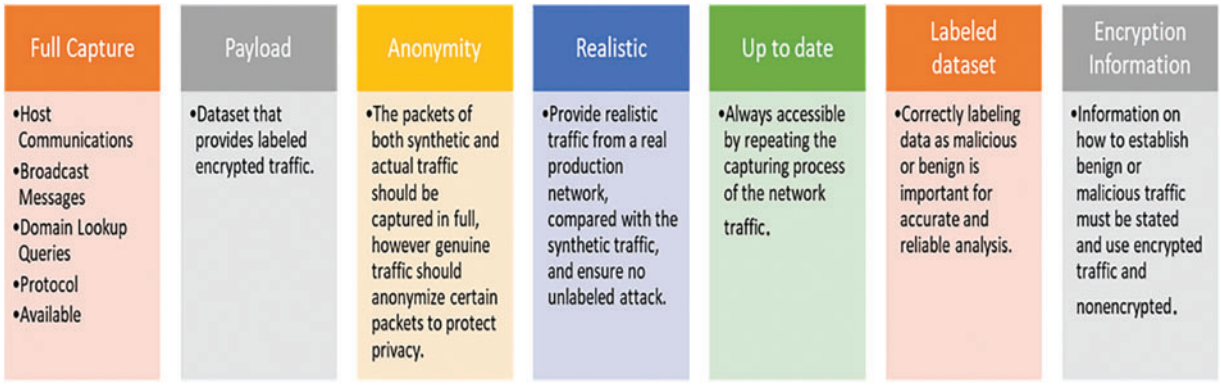

Figure 3: Generate Dataset Requirements

Generating a new dataset with well-defined parameters and feasible applications is essential. This means that the methods should detail how the dataset is created and how the benign and attack traffic is generated, the background traffic is gathered, and the labeling process is carried out. The solution is implemented in the network. More work is required to establish what conditions would call for a synthetic attack and how it will be implemented in the network. In addition, which features and how many of them may be disclosed from packet traces must be disclosed. A detailed and doable guide on creating a new dataset should be accessible.

4.1 Lab

The experiment had three key phrases: setting up the lab, recording regular network traffic, and recording infected network data. Fig. 4 shows the methodology process workflow. We set up a lab with four PCs in the first section. The experiments were run on a four-computer system linked to the network using a flat-switch configuration, with the lab itself linked to a wireless router connected to the Internet. For this test, we used four computers of the same brand and model; all outfitted with Intel Core i7-4770 processors running at 3.40 GHz, 3401 motherboards with 8.00 GB of RAM, and the same kind of network interface card. The computers were set up using Microsoft Windows.

4.1.1 Normal Traffic Capturing

We installed Wireshark on each machine to enable recording network traffic for four computers. After launching Wireshark, we performed a series of steps across all computers to ensure that the collected data represented a diverse range of network activity. Table 1, the list of activities below, describes our chosen activities.

In the end, we stopped the Wireshark, compiled the list of tasks on every computer in the lab, and then saved the PCAP files taken from each computer, labeling them with the computer's name and the phrase 'normal.'

4.1.2 Activating SUNBURST and Capturing Infected Traffic

First, we powered down the lab's network by removing the network cables from NIC cards from each machine. Before beginning the traffic collection, we ensured that no data was being sent between computers or to the Internet; no data would be transferred after enabling the DLL file of

the SUNBURST attack. This was a necessary action to take in order not to get disconnected. We will activate the attack on PC04, which represent the SolarWinds Server.

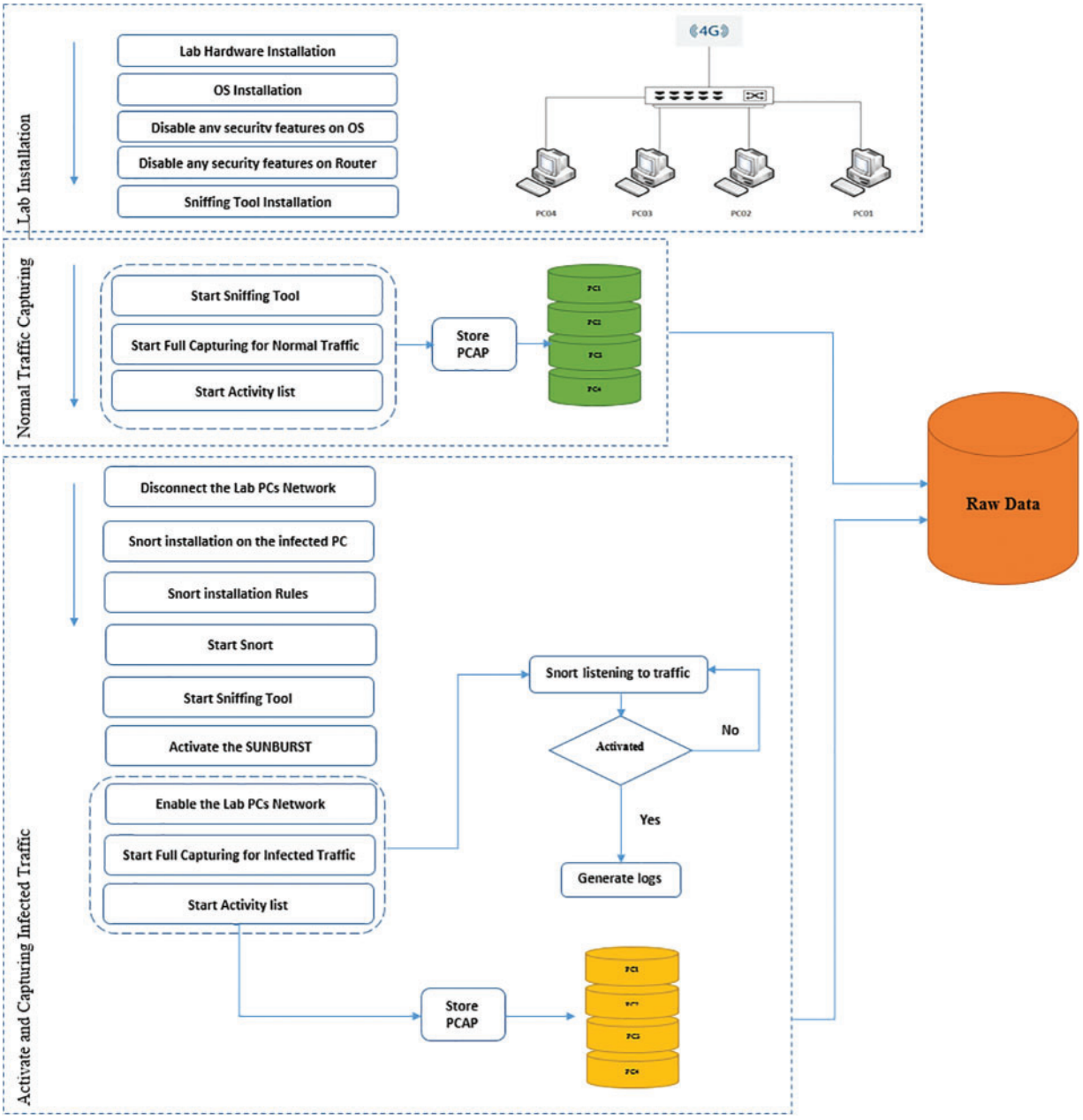

Figure 4: Methodology Process Workflow

In SUNBURST, contact is conducted at random with unrelated domains. FireEye recommends that any company concerned about SUNBURST infection install Snort and apply for the corresponding roles from the GitHub repository. If contact with the malicious domains with which SUNBURST communicates has occurred, FireEye will flag this as evidence of the SUNBURST attack and notify the system administrator. Our infected machine had SUNBURST enabled and functioning with the

help of FireEye's approach. Once the SUNBURST DLL file was enabled, we checked the Snort log to determine whether the SUNBURST actives infecting the traffic with the SUNBURST attack.

Table 1: List of activity

	Activity type
1	File sharing
2	Ping
3	Video streaming
4	Video conferencing
5	Browsing HTTP
6	Browsing HTTPS
7	Uploading files
8	FTP
9	Idle mode

Snort is a free and open-source Network Intrusion Prevention System (NIDS) that can monitor and record IP network traffic in real-time. Among the many attacks and probes, it can detect are buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting efforts, and many more. It can also do content searching and matching.

Download Snort rules from Snort website https://www.snort.org/downloads; the rules will be downloaded as a compressed file. Unzip the downloaded file, then copy the rules and preprocessor rules folders to the Snort directory on PC04 c:\Snort. Copy and replace the folders. Configure the snort.config file. To do this, go to directory c:\Snort\etc\, then open snort.config with notepad++. Reconfigure the lines as in table one line by line.

Now we must prepare backdoor rules to capture the SUNBURST backdoor traffic. One of the countermeasures that FireEye takes to let you know if the SUNBURST backdoor infects you is by making 23 rules for the Snort tool that can give you alerts for any traffic that looks like a SUNBURST backdoor traffic when it comes in or out of your computers. To ensure that SUNBURST is correctly configured on PC04 and traffic is being captured, we utilized Snort rules as a warning system and used the link in [43] to have the roles.

After running the command, the Snort will be ready to listen to the PC04 network traffic. While PC01, PC02, and PC03 were disconnected from the network, we needed to download the SUNBURST file onto PC04 to experiment. In order to combat the SUNBURST attack, FireEye has made available in GitHub all of the indicator release hashes that were used to detect it. Using the URL in [44], you can copy the hash value of the SUNBURST DLL file to help you find the DLL file.

We need to use Any to download the DLL as a. exe file. Run. Any. The run is a dynamic and static internet malware analysis tool designed to study various cyber dangers. The fact that Any. Run was built as an interactive analysis tool, giving it a significant leg up on the competition. The goal was to reveal everything necessary for completing the job. When using Any. Run, the researcher may watch the simulation running, and he may monitor the formation of different processes in real time.

Any. Run aims to offer a comprehensive method of interactive testing with real-time access to the sandbox simulation since some contemporary malicious applications may deceive automated analysis. Download the sample by clicking on Get Sample; the sample will be downloaded to the computer as

a compressed file. When extracting the infected file, use the password 'infected.' Mind your step. Do not extract the file on your PC or other machines linked to your environment network; doing so might infect and compromise your local network. Instead, it would help if you did it in a lab with an isolated network. After decompressing the sample, copy the file to the path below:

C:\Users\admin\AppData\Local\Temp\

We have two actions to activate the sample (the infected DLL) in our lab. After copying the DLL file, open the CMD as administrator and use the command rundll32.exe. The executable file known as "rundll32.exe" is used in the process known as "running DLLs," which stands for dynamic link library.

In order to launch the extracted sample, we must be prepared with the PEStudio tool. With PEStudio, we can load and examine the malicious DLL into the operating system. PEStudio is a tool that can analyze malware statically and launch malicious files to simulate the behavior. Once the DLL file is uploaded to PEStudio, the launched DLL File will be activated on PC04.

Before we run the sample, enable the NIC card on PC01, PC02, and PC03, run the Wireshark on all computers, and start capturing the traffic when the Snort and Wireshark are ready and running on PC04. Check the Snort to see if it gives any alerts, check if Wireshark works on all computers, and capture the traffic. We left the lap running until we had an alert on Snort. After we had activated the sample, we noted that the Snort began to alert.

Follow the steps in the list of activities Table 1, and after you are done, save all the captured traffic on the lab's computers as PCAP files labeled with the computer's name and the word 'infected.'

4.2 Dataset

Here, we detail the steps used to compile our dataset, dubbed SUNBURST2022. The first step is to set up a small lab with four computers, each with an internet connection and the ability to collect regular traffic without activating the infected DLL file. The attack is then activated by initializing the SolarWinds DLL file on the PC04 target network, where background traffic is collected, and attackers manufacture actual innocuous traffic as in the activity list.

The intruder's network has been set up to record all incoming and outgoing communication. This is done so that we can tell the difference between natural and malicious communications and those that are artificial. Several factors may be used to identify safe and malicious communications. The packet traces are then processed to hide the background traffic's identity and to extract characteristics. In addition to the documentation, the packet traces and extracted characteristics comprise the final dataset.

Our attention is focused on SUNBURST Backdoor attacks at the application layer. According to the 2021 Data Breach Investigation findings, application layer attacks account for most (80%) of all attack vectors. Full Capture, Payload, Anonymity, Realistic, Up to Date, Labeled Dataset, and Encryption Information are the seven pillars of the new data collection method. The dataset was compiled with the help of an analysis of various related research, which included the techniques used to identify other IDS datasets successfully. Since the PCAP format is widely utilized for malicious software detection model development, other related research was also examined for the data format. Packet capture is a method used in networking that involves snatching data packets from different OSI levels that ran the lab in the first stage for one d in our experiment. The activity list takes less than half an hour. We keep ladled for less than an hour. In stage two (infected traffic), we keep the lab running for 14 days in idle mode. As explained in Section 2.1.5, the SUNBURST attack keeps hiding

for 14 days until it studies the network environment. As was indicated, the methods used to obtain the information primarily focused on two types of data, which we summarize as follows:

Normal Traffic data is required during model training since it reflects 'typical' data flow. In this group, information was gathered without any attack activation.

Infected Traffic data shows network activity while an attack is being activated. Information from this study may be used to create a model for detecting infections after they have already occurred. Here, we gathered information that may be utilized for hybrid approach detection, such as DLL package file traffic and packets transferred between C2C and the infected device.

Pre-processing, or data preparation, is used before ML analysis to clean and combine raw data. Although it is not the most fun part of the job, getting the data ready for analysis is crucial. Properly verifying, cleaning, and enhancing raw data is essential for gaining valuable insights. To a large extent, the reliability and usefulness of any ML analysis depend on how well the data for the study was prepared in advance. Data preparation consists primarily of four steps: collecting, analyzing, cleaning, and reformatting. We used Wireshark to capture data in our lab, which we then stored in PCAP files. Our research was structured around a series of predetermined tasks.

The term 'data exploration,' which refers to verifying data quality and analyzing data distribution, has been defined at length. In this section, we talk about testing the collected data using the suggested detection model. The data in a dataset is 'cleaned up' when mistakes, broken data, duplicates, and improperly organized data are removed.

We utilized Weka to correct minor structural issues in some of the gathered packets and to filter out unnecessary duplicate packets, particularly for DNS. To preserve the integrity of the data, we have eliminated any incomplete packets. We have already embraced the PCAP format, the CICFlowmeter program that creates CSV files, and the raw process packets for analyzing data. Equipment utilized for data cleaning and manipulation is listed in Table 3.

CICFlowMeter was used to extract the following characteristics from the PCAP file. CIC's CICFlowMeter can create 84 different characteristics of network traffic. It can take a PCAP file, both input, and output, as a graphical report of the retrieved characteristics and a CSV file. It is free to use and may be downloaded from GitHub, a Java open-source program. Integrating its source codes into a project allows for further customization regarding which features are calculated, which are included, and how long the flow timeout lasts. See Table 2.

Table 2: Dataset preparation tools

Stage	Used tools
Acquiring	Wireshark
Exploring	CICFlowmeter
Cleaning	Weka
Transforming	CICFlowmeter

Finally, we add a class column to each CSV file to indicate whether its characteristics should be labeled as normal or abnormal.

- Class A: This group represents the typical information flow experienced by computers and is used to train the classification model. In this group, we gathered data under normal conditions without initiating any attack and classified the results as 'normal traffic.'

- Class B: In this class, data were collected by operating a SolarWinds-infected DLL file. In this class, a unified list of activities was applied over all lab computers to guarantee consistency within all data collected from experiment phase two. These data can be used to develop a post-infection detection model.

After collecting the data, we utilize CICflowmeter to transform the PCAP files into CSV files, as shown in Fig. 5. We merge each normal and abnormal characteristic into a single CSV file.CICflowmeter is then used to extract features from the PCAP files and generate CSV files of the dataset, followed by labeling the instances in the dataset. The SUNBURST dataset has 81 features, 80 network features, and one labeled feature. Table 3 displays the two label features present. The dataset consists of 50,910 records, 7197 of which are regular traffic and 43713 are anomaly records. Fig. 6 displays the distribution of anomaly and benign instances in the SUNBURST dataset.

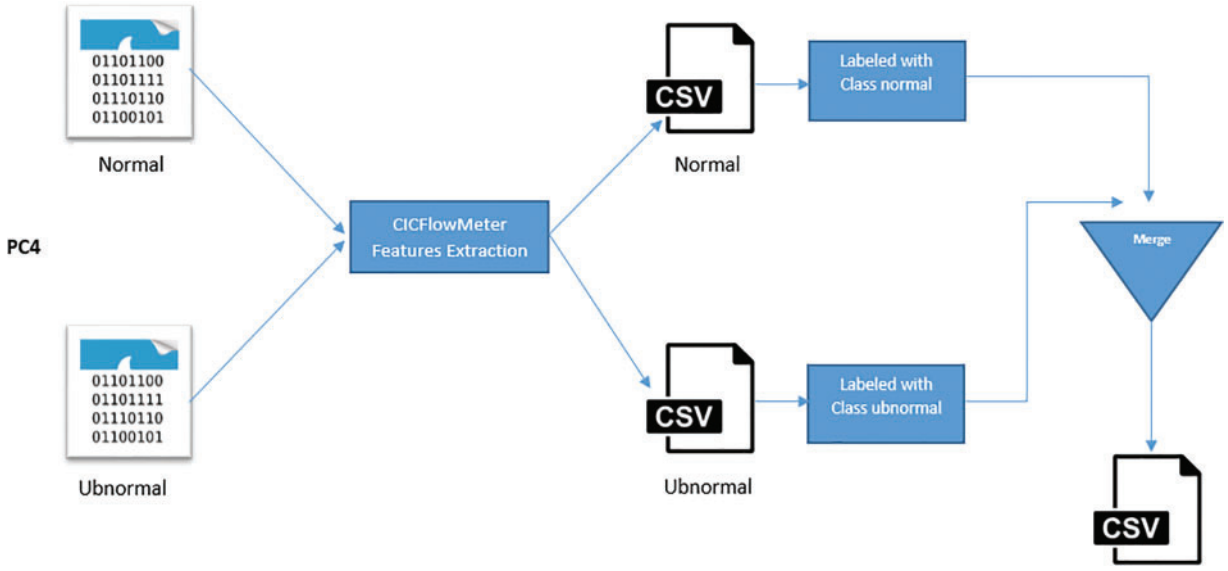

Figure 5: Convert and label PCAP file workflow

Table 3: SUNBURST label features

Binary	Class
Normal	Normal
Abnormal	SUNBURST

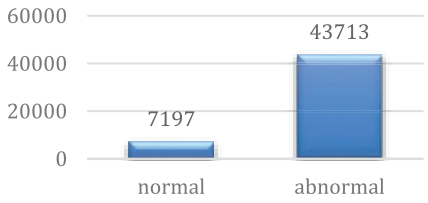

Figure 6: Binary class distribution

We need to extract the characteristics utilized in machine learning from the PCAP file, from which the CICFlowMeter already retrieves more than 80. We must use the Ranker method, whose attributes are placed in order depending on each Ranker's subjective opinions. Then, they are utilized with assessors of attributes (ReliefF, GainRatio, Entropy, and so on).

4.3 Machine Learning

This section provides a comprehensive overview of the decision trees (J48 Algorithm) that are most often used [45,46]. J48 is a classifier that employs the C4.5 algorithm and is a component of the classification strategy used in data mining. If you have numerical and categorical data to classify, you may rely on the tried-and-true C4.5 approach. The estimated worth of the new record's discrete type attribute is often derived from the results of the categorization function inside the sort rules. To put it simply, the C4.5 algorithm is ID3's successor. Improvements are made in handling missing data, continuous data, and pruning. Concerning the ID3 algorithm, the C4.5 algorithm excels. The benefit is in the information it can collect. C4.5 is better than other options since it uses the gain ratio to prioritize attributes [46].

Using the attribute values of items in the training dataset, the J48 approach classifies the data into ranges. The J48 technique disregards rows with missing values for elements whose values can be inferred using information from other rows' attribute values [47].

As the name implies, J48 employs a greedy approach wherein decision trees are constructed by recursive attribute separation, with the attribute at the top being the most important of the attributes below it. The predicted error rate is used to decide whether or not to prune a branch of the tree in J48's pessimistic pruning method. When an attribute variable is tested to determine whether it satisfies the test value, the J48 method begins with a root node and divides it into two halves based on the results. A leaf, also known as a label or class, is the result.

The J48 algorithm builds a decision tree with the following steps:

1. Make attribute the root attribute.
2. For each value, make a branch.
3. Separate the cases into branches.
4. Repeat the process for each branch until all cases have the same class.

$$Gain\,(S, A) = Entropy\,(S) - \sum_{i=1}^{n} \left(\left(\frac{|Si|}{|S|} \right) * Entropy\,(Si) \right) \tag{1}$$

where:

S = a set of instances

A = attribute n = number of partitions of

A |Si| = number of cases on partition i

|S| = number of cases in S

The basic formula for entropy is as follows:

$$Entropy\,(S) = \sum_{i=0}^{n} -\text{pi} * \text{log2pi} \tag{2}$$

where: S = case set, A = feature, n = number of partitions, S pi = proportion of Si to S

4.3.1 Accuracy

Accuracy is a well-known performance parameter that distinguishes a robust classification model from a poor one when assessing a Binary Classifier. Expressed accuracy is the percentage of observations for which predictions were accurate. Four major components make up the mathematical formula for determining accuracy, namely TP, TN, FP, and FN, and these components allow us to investigate other ML Model Evaluation Metrics [48]. The accuracy could be calculated as in the equation:

$$\textbf{Accuracy} = \frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})} \tag{3}$$

- The number of 'True Positives' is represented by the TP. The total number of observations in the 'positive class' predicted correctly is referred to in this number.
- The number of True Negatives (TN) is shown in the TN column. There are a total number of negative observations that have been accurately anticipated.
- False Positives (FP) are the total number of incorrect results. Kind 1 Errors are another name for this type of error. All observations projected to be positive but negative are included in this total number of observations.
- False Negatives (FN) is the total number of them. Type 2 errors are what they are called. This is the total number of observations anticipated to fall into the negative category but instead, fall into the upbeat category.

4.3.2 Precision

This is the percentage (total number) of all truly positive observations. The Precision Evaluation Metric is calculated as in the equation:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

4.3.3 Recall or True Positive Rate

It refers to the percentage of observations expected to fall into the positive category. We may infer from this that the model can randomly pick out a positive observation. Recall that TPR is calculated as in the equation:

$$TPR = \frac{TP}{TP + FN} \tag{5}$$

4.3.4 F1 Score

This evaluation metric averages the results and produces a ratio. Evaluation Metrics is another name for the F1 Score. This evaluation metric measures the total accuracy that our model has attained in a positive prediction environment, i.e., the proportion of positive observations among all that our model has classified as such (Mishra & Mishra, 2016). The F1 Score Evaluation Metric's formula is calculated as in the equation:

$$F1\ Score = 2 \times \frac{\mathrm{Precision}\ \times\ \mathrm{TPR}}{\mathrm{Precision}\ +\ \mathrm{TPR}} \tag{6}$$

4.3.5 Confusion Matrix

In a confusion matrix, the number in each cell represents the fraction of times the model correctly or mistakenly identifies a given set of classes. When explaining the confusion matrix, a binary classification issue is usually used.

4.3.6 Training the Model

In this stage, the model is trained on a range of smaller datasets and evaluated against a smaller testing set. This is referred to as K-fold cross-validation. Cross-validation is a statistical approach for evaluating the performance of machine learning models.

The number of groups into which a given data sample is partitioned is specified by a single parameter called k. As a result, the procedure is often referred to as k-fold cross-validation. When a specific value for k is chosen, it may be replaced in model references. In this model, k = 10 is utilized to represent 10-fold cross-validation. As illustrated in Fig. 7, this stage entails randomly shuffling the dataset, splitting it into ten groups for each unique group, using the group as a holdout or test dataset, then using the remaining group as a training dataset, and finally fitting the model on the training set and evaluating it on the test set.

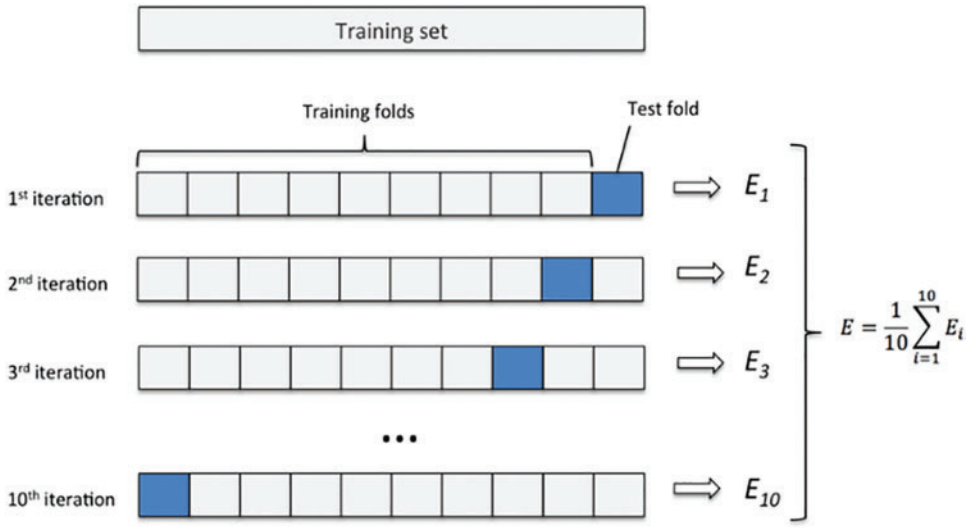

Figure 7: K-Fold cross validation technique [47]

5 Results and Discussion

Metrics for verifying this research's findings have been selected per the assessment metrics described in the preceding chapter. Therefore, we will provide the outcomes of the whole suite of metrics, including accuracy, precision, recall, and the F1 score. This study contains a suggested data-collecting process, a new dataset, and a proposed detection model, as described in the goals section of this research. We will also validate the suggested data-gathering technique and the dataset by testing them with a machine-learning model. That is why the J48 algorithm was used. The J48 algorithm is frequently used in machine learning for discrete and continuous data analysis. To evaluate clinical data for the diagnosis of coronary heart disease, categorize E-governance data, and many other applications, the C4.5 algorithm (J48) is widely utilized, we will break down the outcomes of

our experiments into two distinct sections: (1) Classification Results, where we provide a detailed accounting of our findings; and (2) Discussion, where we analyze and interpret those findings.

5.1 Classification Results

Our dataset will be evaluated and analyzed to see whether it suits a detection model. The model findings are a real-world and scientific analysis of the dataset, particularly those acquired in a real-world setting without using virtual data production techniques. This section looks at our dataset, a model for detecting SUNBURST attacks.

Throughout the construction of our dataset, there were three major experimental milestones. The first phase was to install the hardware and software; the second was to collect regular and SUNBURST traffic in a real-world context; and the third was to evaluate the obtained SUNBURST data for attack detection and categorization using machine learning.

In this part, we detail the innovative dataset and data-collecting approach we used to generate it and the experimental findings we gathered to verify our suggested model. The measures will be evaluated and described in full. Classification findings, accurate precision by class, and confusion matrix outcomes will be presented as portions of the experimental results.

Step one: We detailed how we adopted the Ranker method to rank the characteristics of the SUNBURST dataset. The features were evaluated using a ranking algorithm (supervised, Class (nominal): 81 class).

Step two: The classifiers adapted to the SUNBURST dataset were J48. We experimented with three phases. In the first phase, we selected all features we had (81 features); in the second phase, we selected the top 10 features; and in the final phase, we selected the top five features. All readings were taken from the three experiments and were as shown in Fig. 8 below:

Fig. 8 displays the findings of all features, which show that despite the F-measure (0.856) being lower than Recall, the precision gave an excellent indication (0.899) when we picked all characteristics.

Figure 8: F-measure results for all features

Table 4 shows confusion matrix results for selecting all features.

Table 4: Confusion matrix 1

Confusion matrix			
	Abnormal	Normal	
Abnormal	43685	28	43713
Normal	5651	1546	7197
	49336	1574	

Fig. 9 shows that when ten features are chosen from the Ranker evaluator, both the precision and recall were two closes. In addition, Table 5 displays a confusion matrix for ten features.

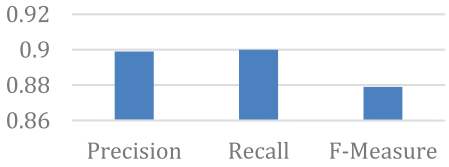

Figure 9: F-measure results for top 10 features

Table 5: Confusion matrix 2

Confusion matrix			
	Abnormal	Normal	
Abnormal	43412	301	43713
Normal	4810	2387	7197
	48222	2688	

Additionally, the ten features are shown in Table 6.

Table 6: Top 10 features

	Rank	Feature no.	Feature name
1	0.587808	4	Timestamp
2	0.237408	1	Src port
3	0.232171	2	Dst port
4	0.223438	5	Flow duration
5	0.216513	41	Bwd Pkts/s
6	0.204244	20	Flow IAT mean
7	0.204074	56	Pkt size avg
8	0.201044	22	Flow IAT max
9	0.197691	44	Pkt len mean
10	0.193187	16	Bwd Pkt len mean

When just five characteristics from the Ranker evaluator table are employed, as shown in Fig. 10, the J48 classifier obtains a respectable percentage, almost flawless accuracy, and recall. This may be observed by referring to the accompanying graph. Nevertheless, if we consider the best outcome with the top 10 characteristics. Additionally, the five features selected are shown in Table 8.

5.2 Discussion

This part goes through the results reported in the previous sections. We investigated several observations about the presented findings. Furthermore, we assessed the conclusions of this investigation considering prior studies that presented comparable detection strategies.

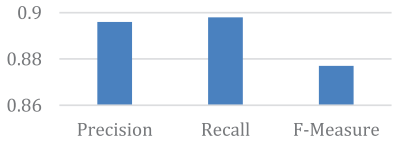

Figure 10: F-measure results for top 5 features

Table 7: Confusion matrix 3

Confusion matrix			
	Abnormal	Normal	
Abnormal	43395	318	43713
Normal	4868	2329	7197
	48263	2647	

Table 8: Five features

	Rank	Feature no.	Feature name
1	0.587808	4	Timestamp
2	0.237408	1	Src port
3	0.232171	2	Dst port
4	0.223438	5	Flow duration
5	0.216513	41	Bwd Pkts/s

According to the findings, it is abundantly evident that there is a disparity in the accuracy rate of the suggested model for the SUNBURST attack, and the reason for this disparity is related to several factors, all of which can be summed up in the following points:

In this experiment, the classifier J48 was applied to the SUNBURST dataset, and Figs. 8–10 respectively showed the performance of the three phases regarding Recall, Precision, and F-measure based on selected attributes from the Ranker evaluator result. The figures above indicate that the J48 classifier achieved excellent results in correctly identifying abnormal traffic records. Fig. 11 shows all the results together.

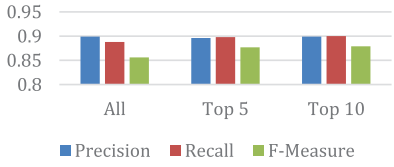

Figure 11: Compare results

From Fig. 11, we also note that by selecting the top ten features, J48 classifiers had the highest score, which means that choosing the top ten features is enough to detect abnormal traffic. After reviewing all the results mentioned above, we can summarize the results as in Table 9:

Table 9: Results summary

	Precision	Recall	F-measure
All	0.899	0.888	0.856
Top 5	0.896	0.898	0.877
Top 10	0.899	0.9	0.879

The confusion matrices in Tables 4, 5, and 7 for the J48 classifier confirm the performance of these classifiers. The values in the matrices demonstrate that all attack records and average records in the testing set are correctly classified. Also, the confusion matrix for the J48 classifier indicates that most of the records in the testing set are correctly classified, except for a small number of misclassified attack records.

Based on the reported results, the proposed model is successful and has achieved acceptable results without going into the details of the feature engineering, which undoubtedly results in better results. The results show that it is feasible to use network traffic data analysis to build reliable detection models for this attack, which can be used at different times during the system's life cycle.

6 Conclusion

In this study, we introduced a novel SUNBURST attack detection dataset and a machine learning-based detection model. We began by exploring the supply chain attack concept, its impact on the ICT sector, and its history to comprehend related security threats. The gap in the literature was identified, and our proposed dataset, generated from real lab network traffic, was designed to fill this need. We utilized 81 features, and the J48 algorithm validated the dataset's usefulness as a benchmark. The detection model demonstrated exemplary performance detecting the SUNBURST attack in binary-class classifications. Our findings indicate that the SUNBURST dataset and machine learning techniques effectively contribute to network anomaly detection and network security. Moving forward, we will focus on enhancing backdoor detection models by examining reverse engineering applications, expanding feature engineering analysis, applying various algorithms, updating the SUNBURST dataset, and experimenting with simultaneous attacks. By adopting new machine learning algorithms and feature selection techniques and investigating balanced and unbalanced datasets, our research can contribute to improved prevention and mitigation strategies for SUNBURST attacks, advancing the field of cybersecurity.

Acknowledgement: The author (Amjad Aldweesh) would like to thank the Deanship of Scientific Research at Shaqra University for supporting this research.

Funding Statement: The authors received no funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Schwab, The Fourth Industrial Revolution. Crown Business, 2017. [Online]. Available: https://shorturl.at/ctESW

[2] M. Kaku, The future of the mind: The scientific quest to understand, enhance, and empower the mind. Anchor, 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5179617/

[3] G. S. Sriram, "Resolving security and data concerns in cloud computing by utilizing a decentralized cloud computing option," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 4, no. 1, pp. 1269–1273, 2022.

[4] J. S. Nye, *The Regime Complex for Managing Global Cyber Activities*. vol. 1. USA: Belfer Center for Science and International Affairs, John F. Kennedy School of Government, Harvard University, 2014.

[5] M. Ohm, H. Plate, A. Sykosch and M. Meier, "Backstabbers knife collection: A review of open-source software supply chain attacks," in *Detection of Intrusions and Malware, and Vulnerability Assessment: 17th Int. Conf. DIMVA 2020*, Lisbon, Portugal, pp. 23–43, 2020.

[6] M. Andreessen, "Why software is eating the world," *Wall Street Journal*, vol. 20, no. 2011, pp. C2, 2011.

[7] Mandiant, "Remediation and hardening strategies for Microsoft 365 to defend against UNC2452: Blog," Mandiant. [Online]. Available: https://www.mandiant.com/resources/blog/remediation-and-hardening-strategies-for-microsoft-365-to-defend-against-unc2452 (Accessed 26 February 2023).

[8] A. A. Mawgoud, M. H. N. Taha, N. E. M. Khalifa and M. Loey, "Cyber security risks in MENA region: Threats, challenges and countermeasures," in *Proc. of the Int. Conf. on Advanced Intelligent Systems and Informatics*, Cairo, Egypt, pp. 912–921, 2019.

[9] G. Ye, Z. Tang, D. Fang, Z. Zhu, Y. Feng *et al.,* "Yet another text captcha solver: A generative adversarial network-based approach," in *Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security*, Copenhagen, Denmark, pp. 332–348, 2018.

[10] P. Datta, "Hannibal at the gates: Cyberwarfare and the Solarwinds sunburst hack," *Journal of Information Technology Teaching Cases*, vol. 12, no. 2, pp. 115–120, 2022.

[11] S. Vaughan-Nichols, "SolarWinds: The more we learn, the worse it looks," 2021. [Online]. Available: https://www.zdnet.com/article/solarwinds-the-more-we-learn-the-worse-it-looks/

[12] M. A. Haq, G. Rahaman, P. Baral and A. Ghosh, "Deep learning based supervised image classification using UAV images for forest areas classification," *Journal of the Indian Society of Remote Sensing*, vol. 49, no. 2, pp. 601–606, 2021.

[13] L. Cerulus, "SolarWinds is "Largest," Cyberattack Ever, Microsoft President Says," 2021. [Online]. Available: https://rb.gy/ws6ti

[14] Mandiant, "Cyber Threat Defense Solutions: Threat Intelligence Services," Mandiant, 2023. [Online]. Available: https://www.mandiant.com/?utm_source=google&utm_medium=cpc&utm_campaign=BRND%7CRelentless%7CEMEA%7CEMER%7CEN%7CSearch&utm_content=all&utm_term=en&cid=global&gad=1&gclid=Cj0KCQjw7uSkBhDGARIsAMCZNJvgE8iehXXXE-QuhzsmJmCvJmrd6FTEroD6vaxJxWADtpTHw0hAonsaAtxUEALw_wcB (Accessed 01 January 2023).

[15] Mandiant, Assembling the Russian Nesting Doll: UNC2452 Merged into APT29, 2022. [Online]. Available: https://www.mandiant.com/resources/webinars/assembling-russian-stacking-doll-unc2452-merged-apt29

[16] A. Villarreal, "Russian solarwinds hackers launch email attack on government agencies," The Guardian, 2021. [Online]. Available: https://www.theguardian.com/technology/2021/may/28/russian-solarwinds-hackers-launch-assault-government-agencies (Accessed 18 November 2022).

[17] M. Lella, E. Theocharidou, A. Tsekmezoglou, A. Malatras and European Union Agency for Cybersecurity, "Enisa threat landscape for supply chain attacks," European Union Agency for Cybersecurity (ENISA)," 2021. [Online]. Available: https://industrialcyber.co/download/enisa-threat-landscape-for-supply-chain-attacks/

[18] ODSCCommunity, "3 ways to protect your code from software supply chain attacks," ODSC Community," 2023. [Online]. Available: https://opendatascience.com/3-ways-to-protect-your-code-from-software-supply-chain-attacks/

[19] General Services Administration (GSA), "Advanced Persistent Threat buyer"s Guide: What are Advanced Persistent Threats?," General Services Administration (GSA)," 2021. [Online]. Available: https://www.gsa.gov/cdnstatic/APT_Buyers_Guide_v11_20210121.pdf

[20] A. Nardozza, "Unpacking an unprecedented cyberattack: What is the solarwinds breach and how did it happen," 2021. [Online]. Available: https://jost.syr.edu/unpacking-an-unprecedented-cyberattack-what-is-the-solarwinds-breach-and-how-did-it-happen/#_ftn3

[21] L. Sterle and S. Bhunia, "On solarwinds orion platform security breach," in *2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI)*, Virtual Conference, pp. 636–664, 2021.

[22] P. Baker, "The solarwinds hack timeline: Who knew what, and when?," CSO Online, 2021. [Online]. Available: https://www.csoonline.com/article/3613571/the-solarwinds-hack-timeline-who-knew-what-and-when.html (Accessed 12 December 2022).

[23] S. M. K. Saheed Oladimeji, "Solarwinds Hack explained: Everything you need to know," WhatIs.com, 2022. [Online]. Available: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know (Accessed 07 January 2023).

[24] S. Ramakrishna, "New findings from our investigation of Sunburst," Orange Matter, 2022. [Online]. Available: https://orangematter.solarwinds.com/2021/01/11/new-findings-from-our-investigation-of-sunburst/ (Accessed 11 October 2022).

[25] M. Alkasassbeh, M. Almseidin, K. Alrfou and K. Szilveszter, "Detection of IoT-botnet attacks using fuzzy rule interpolation," *Journal of Intelligent and Fuzzy Systems*, vol. 39, no. 1, pp. 421–431, 2022.

[26] M. Almseidin and M. Alkasassbeh, "An accurate detection approach for IoT botnet attacks using," *Information*, vol. 13, no. 6, 2022.

[27] M. Almseidin, J. Al-Sawwa, M. Alkasassbeh and M. Alweshah, "On detecting distributed denial of service attacks using fuzzy inference system," *Cluster Computing*, vol. 26, no. 2, pp. 1337–1351, 2023.

[28] P. A. Das, "A deep transfer learning approach to enhance network intrusion detection capabilities for cyber security abhijit das," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, 2022.

[29] S. P. Kulyadi, S. Pai, S. K. S. Kumar, M. J. S. Raman and V. S. Vasan, "Anomaly detection using generative adversarial networks on firewall log message data," in *2021 13th Int. Conf. on Electronics, Computers and Artificial Intelligence (ECAI)*, Bucharest, ROMANIA, pp. 1–6, 2021.

[30] A. Duby, T. Taylor, G. Bloom and Y. Zhuang, "Evaluating feature robustness for windows malware family classification," in *2022 Int. Conf. on Computer Communications and Networks (ICCCN)*, Virtual Conference, pp. 1–5, 2022.

[31] M. A. Haq, "DBoTPM: A deep neural network-based botnet prediction model," *Electronics*, vol. 12, no. 5, pp. 1159, 2023.

[32] M. A. Haq and M. A. R. Khan, "DNNBoT: Deep neural network-based botnet detection and classification," *Computers, Materials & Continua*. vol. 71, no. 1, pp. 1729–1750, 2022.

[33] M. A. Haq, M. A. R. Khan and A. H. Talal, "Development of PCCNN-based network intrusion detection system for EDGE computing," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1769–1788, 2022.

[34] M. A. Haq, M. A. R. Khan and M. Alshehri, "Insider threat detection based on NLP word embedding and machine learning," *Intelligent Automation and Soft Computing*, vol. 33, no. 1, 2022.

[35] J. Upchurch and X. Zhou, "Variant: A malware similarity testing framework," in *IEEE Conf. on Malicious and Unwanted Software (MALWARE)*, Fajardo, PR, USA, pp. 1–8, 2015.

[36] R. Alkhadra, J. Abuzaid, M. AlShammari and N. Mohammad, "SolarWinds hack: In-depth analysis and countermeasures," in *2021 12th Int. Conf. on Computing Communication and Networking Technologies (ICCCNT)*, IIT Khargpur, West Bengal, India, pp. 1–6, 2021.

[37] M. Shlapentokh-Rothman, J. Kelly, A. Baral, E. Hemberg and U. M. O'Reilly, "Coevolutionary modeling of cyber-attack patterns and mitigations using public datasets," in *Proc. of the Genetic and Evolutionary Computation Conf.*, Lille France, pp. 1–8, 2021.

[38] M. Santaniello, "Sunburst. La grande eclissi della cybersecurity USA," *Rivista di Digital Politics*, vol. 1, no. 1, 2021.

[39] M. Alkasassbeh and S. Al-Haj Baddar, "Intrusion detection systems: A state-of-the-art taxonomy and survey," *Arabian Journal for Science and Engineering*, vol. 2022, no. 2, pp. 1–44, 2022.

[40] C. Luo, L. Wang and H. Lu, "Analysis of LSTM-RNN based on attack type of KDD-99 dataset," in *Int. Conf. on Cloud Computing and Security*, Haikou, China, pp. 326–333, 2018.

[41] GitHub, "all-snort.rules. FireEye," [Accessed: 15-Dec-2023]. Available: https://github.com/fireeye/sunburst_countermeasures/blob/main/all-snort.rules

[42] FireEye, "Indicator_release_hashes," [Accessed: 10-Jan-2023]. Available: https://github.com/fireeye/sunburst_countermeasures/blob/main/indicator_release/Indicator_Release_Hashes.csv

[43] D. R. Hermawan, M. F. G. Fatihah, L. Kurniawati and A. Helen, "Comparative study of J48 decision tree classification algorithm, random tree, and random forest on in-vehicle coupon recommendation data," in *2021 Int. Conf. on Artificial Intelligence and Big Data Analytics, ICAIBDA 2021*, Jawa Barat Indonesia, pp. 76–81, 2021.

[44] N. Saravanan and V. Gayathri, "Performance and classification evaluation of J48 algorithm and Kendall's based J48 algorithm (KNJ48)," *International Journal of Computer Trends and Technology*, vol. 59, no. 2, pp. 73–80, 2018.

[45] S. Goyal, "Evaluation Metrics for Classification Models," [Accessed: 04-Dec-2022], 2023. [Online]. Available: https://medium.com/analytics-vidhya/evaluation-metrics-for-classification-models-e2f0d8009d69

[46] A. K. Mishra and B. K. Ratha, "Study of random tree and random forest data mining algorithms for microarray data analysis," *International Journal on Advanced Electrical and Computer Engineering*, vol. 3, no. 4, pp. 5–7, 2016.

[47] S. Raschka, *Python machine learning*. Packt Publishing Ltd., 2015.

[48] J. Zhou, A. H. Gandomi, F. Chen and A. Holzinger, "Evaluating the quality of machine learning explanations: A survey on methods and metrics," *Electronics*, vol. 10, no. 5, pp. 593, 2021.