Advanced Data Pipelines for Scalable Intrusion Detection in Big Data

[1]Divyesh Vaghani, [2]Amit Siddhpura

Government engineering college, Rajkot

[1]divyeshvaghani96@gmail.com

[2]siddhpuraamit12345@gmail.com

Article Info

Page Number: 44 - 66

Publication Issue:

Vol 67 No. 1 (2018)

Article History

Article Received: *15 October 2018*

Revised: *24 November 2018*

Accepted: *18 December 2018*

Abstract

In an era marked by unprecedented digital transformation, the security of information systems has become paramount. This paper explores the integration of advanced data pipelines for scalable intrusion detection in big data environments, addressing the critical challenges posed by the increasing volume and complexity of cyber threats. By leveraging cutting-edge machine learning algorithms and real-time data process capabilities, organizations can enhance their detection accuracy and response times, ultimately safeguarding their digital assets. The study highlights the importance of collaborative data sharing among organizations to create a unified defense against cyber intrusions. Through a comprehensive review of existing literature and practical applications, this research provides valuable insights into the effectiveness of advanced data pipelines in improving intrusion detection systems (IDS). As cyber threats continue to evolve, this paper serves as a crucial resource for cybersecurity professionals seeking innovative solutions to protect their networks and data.

Keywords: - Advanced Data Pipelines, Anomaly Detection, Big Data, Cybersecurity, Cyber Threats, Intrusion Detection Systems, Machine Learning.

I. Introduction

In today's world, using computers and technology is very important and keeps growing. It is crucial to protect information systems, databases, and networks. This protection helps us in our daily lives, keeps our personal information safe, boosts the economy, and helps businesses run smoothly using digital tools and automated processes (Lee & Lee, 2012). Countries, companies, groups, and important services rely on technology for their everyday work. This dependence has caused technology to grow quickly, but it also makes systems more vulnerable to attacks and intrusions. Cyber-attacks' impact on information systems has increased by 900% worldwide in the last four years (Lornov, 2017). In 2017, Cybersecurity Ventures estimated that the cost of protecting online space will increase by at least 15% annually, reaching almost 11 trillion dollars by 2025 (Terzi et al., 2017). Countries around the world are facing many cyberattacks. Examples include the Stuxnet attack in 2010, which targeted Iran's nuclear program, and the Red October attack in 2012 (Virvilis-Kollitiris, 2015), which hit embassies and government offices. More recently, in September 2016, NATO's secret documents were stolen from the Portuguese Defense Department. In December 2017, an attack called Log4Shell targeted the networks of the U.S. (Terzi et al., 2017).

There have been many more hacking attempts in computer networks recently, and many new hacking tools and methods have been created. Different solutions have been found to keep computer systems safe; one important solution is intrusion detection systems (IDS), which help monitor and deal with suspicious activities in a network (Hafsa & Jemili, 2018).

In the last twenty years, people in schools and businesses have studied and used network intrusion detection systems (NIDS) (Debar et al., 1999). Lately, figuring out when someone is trying to break into a system has become a big data issue. This is because there is a lot more data, which is getting more complicated to understand, all to detect more advanced cyberattacks. Oguntimilehin and Ademola (2014) assert that information technology has advanced a lot, causing a large amount of data to be created quickly from different sources. This information is called big data, and it can be used to find unusual patterns using Big Data Analytics (BDA) and data mining (Razci et al., 2016). An intrusion detection system (IDS) finds and alerts about unusual activities. Myers et al. (2010) described IDS as a system that watches network traffic in real time and provides accurate information to determine if the network is under attack or has already been attacked.

This study assesses advanced data pipelines for scalable intrusion detection in big data. The study provides insights into various data pipelines used for intrusion detection. The rest of the paper is arranged like this. Section 2 has a background of the study, which talks about intrusion detection systems and big data—section 3 reviews related works. Section 4 discusses advanced data pipelines for scalable intrusion detection in big data, and Section 5 concludes the study.

II. Background

Intrusion Detection System

An intrusion is when someone uses a computer system for reasons it was not meant for, usually by getting access in the wrong way. An intruder is usually considered a stranger who takes over a computer system. However, studies show that most problems actually come from people inside the system who already have access (Moustafa & Slay, 2015).

An intrusion detection system (IDS) is a tool that finds and alerts about unusual activities. An intrusion detection system (IDS) is used to find and alert about unusual activities. This system protects things from harmful actions, whether from known or unknown sources. It works automatically to ensure the information is private, accurate, and accessible (Rhodes, et al., 2000). According to Dataricks (2018), an Intrusion Detection System (IDS) has two ways to find problems: anomaly-based detection and signature-based detection. IDS can watch over a company's network, various computers, or software applications.

Intrusion Detection Systems (IDSs) will serve as a backup protection against attacks on big data environments. So, it is important to understand the different types and how they work. In computer science, IDSs (Intrusion Detection Systems) can be grouped into three types based on how they find problems (Debar et al., 1999; Butun, 2013; Butun et al., 2013):

1. Anomaly-based IDS
2. Misuse-based IDS
3. Specification-based IDS

The pros and cons of different types of IDS are listed in Table 1.

Table 1: Comparing Different Types of IDS.

IDS Type	Benefits	Drawbacks
Anomaly detection -based	Can manage unexpected threats does not need to be updated often Simple to set up or apply to different situations	Not very accurate High number of incorrect positive results A lot of missed detections
Misuse detection -based	Very precise Low rates of incorrect positive or negative results	Cannot deal with attacks that are not recognized need regular updates
Specification detection -based	very accurate affordable or low-cost few mistakes in saying something is wrong or right	difficult to create difficult to make general statements

Usually, systems that detect misuse manually write down signs of attacks, known as signatures. These systems are very precise when they send alerts because they can tell what kind of attack the system might face. These systems work well with past attacks that are well-organized and classified. However, they do not help much for new attacks that the old ones cannot explain.

Specification-based systems are created based on rules about how the system should work and are often used for network protocols. These systems work well when there is a clear description of how the system should act and when everyone sticks to that document. People have recommended them for network protocols, but there often isn't a clear set of rules for very complicated systems, or the system's behavior changes a lot. Specification-based IDS is said to be better at finding attacks that target specific processes. However, it can be quite costly to set up in big places like factories, and it does not scale well to larger operations (Fauri, et al., 2017).

Anomaly-based systems often use machine learning to understand how a system usually works without relying on specific rules. An alert is created when the current behavior differs from what was learned. Unlike systems that focus on misuse, anomaly-based systems are not as accurate because they only detect unusual activity and not actual attacks. On the other hand, systems that look for unusual activity can warn us about new types of attacks (like zero-day attacks) if they show noticeable changes in the data being watched (Bhatti, et al., 2012).

According to Tong et al. (2016), the best Intrusion Detection Systems (IDS) method in big data environments is Anomaly-based IDS. This is because it can find new and unknown kinds of attacks. Also, many factory networks behave consistently (like machine-to-machine communication). This means some of their problems are less noticeable when used in factories.

According to Butun (2013), to fully protect against cyber-attacks, a cybersecurity system must have multiple layers of defense. These layers include prevention, detection, and response, as shown in Figure 1.

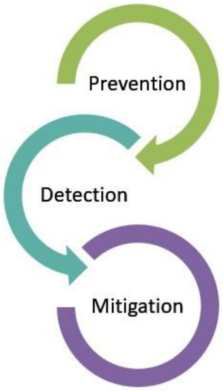

Figure 1: Layers of defense for an intrusion detection system

Prevention: When this layer is used as a system, it is called an Intrusion Prevention System (IPS) and serves as the first line of defense against attacks. Sometimes, security systems like firewalls may not be completely reliable and might not stop every attack on our networks.

Detection: This part of the system is called an Intrusion Detection System (IDS) and serves as the second line of protection against intrusions. History shows us that the first level of protection, called IPSs, can fail. This has happened in serious cyber-attacks on important systems like nuclear plants and power grids, as mentioned earlier. IDSs help system administrators by providing extra solutions to find intrusions in their network quickly, allowing them to deal with threats before they become serious. So, IDSs are just as important as IPSs.

Mitigation: This is the final step in keeping things safe from cyber threats. It involves security actions, like turning off certain ports and restricting internet access, that are taken after a breach is found.

Big Data

Gartner identified three main features of big data in 2001: its large size (volume), how fast it comes in (velocity), and the different types it can be (variety) (Laney, 2001). Volume means how much data there is. Velocity refers to how quickly data is processed. Variety is connected to how complicated the data is. Some people have added two more ideas, Veracity and Value, to the list of characteristics (Zikopoulos et al., 2013). Veracity means how trustworthy and good the data is, including problems like errors or empty values. Value comes from having a lot of data (Zuech et al., 2015).

Big Data is a big challenge for Intrusion Detection and has been an important topic for a long time. In 1994, a study by Frank (1994) about Intrusion Detection showed that a user usually creates 3 to 35 Megabytes of data in eight hours. It can take several hours to examine just one hour's data. They also said that sorting, grouping, and choosing important parts of the data is "important for real-time detection," which can make detecting things more accurate. This example shows that intrusion detection has dealt with big data problems even before the term "Big data" came about.

Big Data Tools and Techniques

Apache Spark

Apache Spark is a strong and fast tool that helps process large amounts of data. It is a popular open-source project in the field of big data. It was created at UC Berkeley in 2009. It was one of the best projects in Apache in 2010 (Microsoft Azure, 2018). Spark offers tools that can be used in Scala, Java, Python, and R programming languages. To manage large amounts of data well, it needs to be processed quickly all at once. So, Spark needs to be available on multiple clusters instead of just one machine. The results from Spark's treatment are stored in memory, not saved on the disk. This ability to use memory fully helps with fast computing for advanced analysis, making Spark 100 times quicker than Hadoop (Daniel & Jacob, 2017).

Many well-known online companies like Netflix, Yahoo, and eBay have started using the popular project. It is built on Scala but also has interfaces for Java, Python, and R. Spark also has a collection of tools that can be used for Machine Learning and running interactive questions. This can greatly affect how much work gets done. The project has been steadily improved to create a complete system, as shown in Figure 2.

Spark Core offers a way to use resilient distributed datasets (RDDs), which allows you to keep data stored in memory. This means users do not have to read the data from the disk every time they need it. Spark Streaming allows Spark to process data in real-time. It turns incoming data into small pieces, called DStreams, so that that information can be delivered quickly and reliably. Spark processes live data in small chunks called micro-batches (Zubair, 2016).

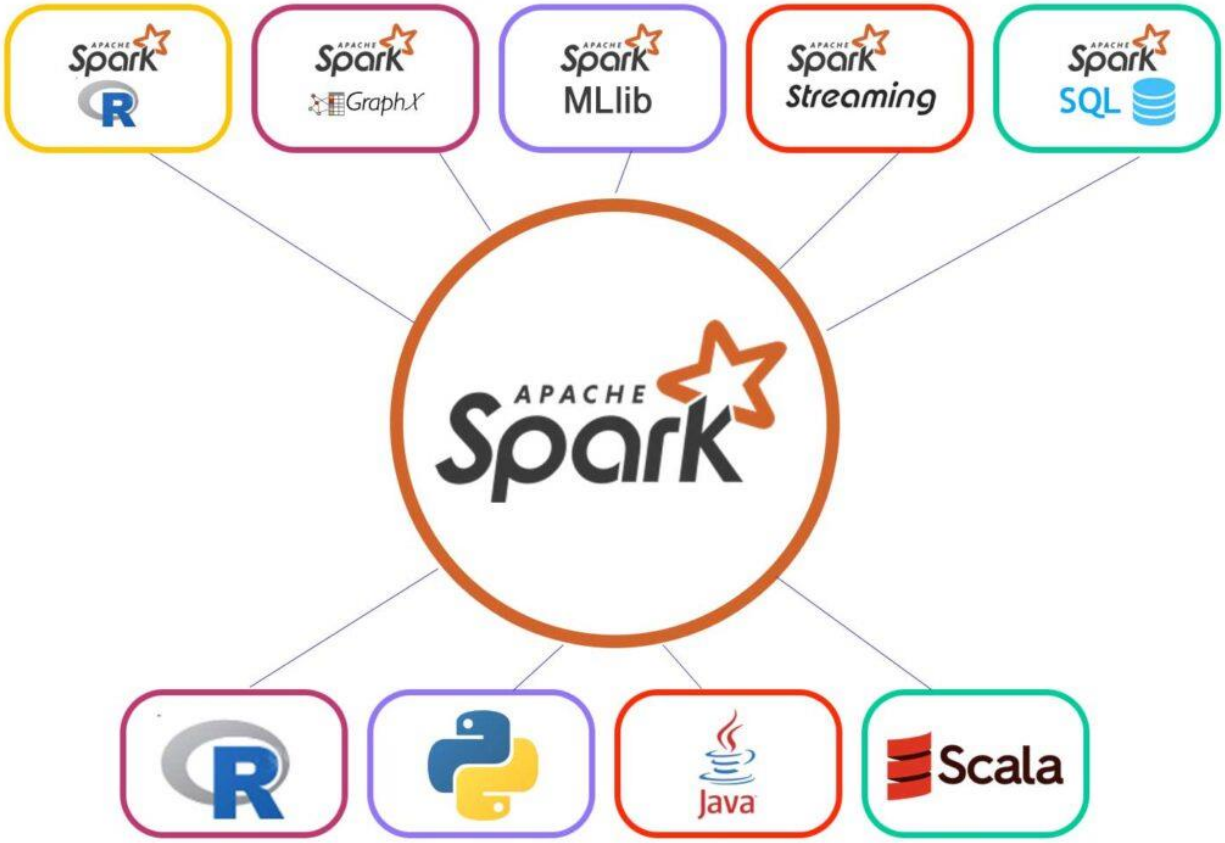

Figure 2: Apache Spark Ecosystem

Microsoft Azure

Microsoft Azure is a cloud computing service from Microsoft. It lets businesses and developers build, store, and manage applications and data in the cloud instead of on local computers. Azure offers many tools and services, including storage, networking, and databases, making it easier for users to create and run their projects (Microsoft, 2018).

Microsoft Azure, once called Windows Azure, is a service that lets users create, run, and manage applications and services online. It uses a network of managed data centers in 54 locations worldwide. Microsoft's HDInsight is a service in Azure Cloud that helps users use Hadoop, a big data tool. It is based on the Hortonworks Data Platform (HDP). Users can easily change HDInsight clusters by adding extra packages. They can also grow bigger when there is much demand by adding more processing power. With Azure Active Directory, data stays safe even if the cluster is removed.

Related Works

Detecting intrusions has always been an important issue in research papers (Ar et al., 2003; Massimiliano et al., 2013). Since the rise of the modern Internet and the increase in big data and cloud computing, researchers are more eager to find new solutions to this problem. Many ways were used to find intrusions in a network; different tools were also used. This part discusses the best ways to find intrusions in Big Data.

Many studies on Intrusion Detection Systems (IDS) use Big Data methods. Jeong et al. (2012) showed that Hadoop can help spot security breaches and handle large amounts of data by concentrating on unusual intrusion detection systems. In the experience of Lee & Lee (2012), researchers found that using Hadoop technologies is a good option for detecting intrusions because they achieved speeds of up to 14 Gbps with a DDOS detector.

Essid and F. Jemili (2016) took the alerts from KDD99 and DARPA, combined them, and removed the duplicates. They used Hadoop to put the data together. Fekih and F. Jemili (2018) also used Spark to combine and clean up three alert databases: KDD99, DARPA, and MAWILAB. The goal was to find more things accurately and reduce the number of mistakes where something important is missed.

Terzi et al. (2017) developed a new method for finding unusual patterns without guidance. It was used with Apache Spark on Microsoft Azure to use powerful, scalable computing. The new method was tested on a botnet traffic dataset called CTU-13 and achieved an accuracy of 96%.

M Hafsa and F. Jemili (2018) developed a new method for spotting intrusions. They used Apache Spark on Microsoft Azure (HDInsight21) to look at and work with data from the MAWILAB database. Their new method reached an accuracy of 99%. Ren et al. (2009) developed a new method for finding unusual data patterns without supervision. They used the KDD'99 data set to look at and process the information but found that their detection rate was low.

Rustam and Zahra (2018) examined two methods for studying and finding intrusions in the KDD99 database. One method was supervised, called the Support Vector Machine (SVM), and the other was unsupervised, called Fuzzy C-Means (FCM). They discovered that SVM had an average accuracy of 94. 43%, while FCM had an average accuracy of 95. 09% In this work, we suggest using Apache Spark to find security breaches. Our system aims to create a fast and effective way to detect intrusions by using Big Data tools and fuzzy logic to handle uncertainty, leading to improved results.

Rai *et al*. (2016) used a hybrid decision tree classifier to build a predictive model for an intrusion detection system (IDS). The model was improved by selecting important features and dividing values into segments to enhance performance. The author chose sixteen features from the NSLKDD dataset, but the experiment results showed only 79.52% accuracy in predictions, with a low false alarm rate. Krishnan and Raajan (2016) created a model to predict network intrusions. They used a type of smart program called a Recurrent Neural Network (RNN) to analyze all of the Cup99 data and categorize it into four types: DoS, Probe, Root to location, and User to Root. The model's performance was measured using a confusion matrix. The analysis showed that the overall accuracy for DoS is 97. 4%, for Probe is 96.6%, for U2R is 86.5%, and for R2L is 29.73%. Vishwakarma *et al*. (2017), the authors suggest comparing two models that detect security breaches. They use binary classification (two categories) and multiclass classification (more than two categories) on part of the Cup 99 dataset, also known as the NSL-KDD dataset. The study used a method to select important features and the decision tree algorithm for building a model. We checked how well the model worked before and after choosing the important features to get better results. The results were looked at and compared. They showed that the accuracy for classifying multiple groups was lower than for two groups. The accuracy for classifying all groups was 83.7%, and for a smaller set of groups, it was 90.3%. The false alarm rate was 2.5% for the full set and 9.7% for the smaller set.

Mabayoje *et al*. (2015) suggested a system for detecting intrusions that used a decision tree method. It chose important features using the Gain Ratio technique and used the KDD Cup99 dataset for training. The study used a method called 10-fold cross-validation on the complete and smaller data sets, applying a technique for classifying multiple categories. The complete dataset shows that the experiment predicted DoS attacks with 100% accuracy and, probe attacks with 99.49% accuracy, Remote to Local attacks with 98% accuracy, and User to Root attacks with 75% accuracy. In the smaller dataset, the prediction accuracy for DoS attacks was 100%; for probe attacks, it was 99.49%; for Remote to Local attacks, it was 98%; and for User to Root attacks, it was 75%. Using a part of the KDD Cup 99 dataset, Vishwakarma *et al*. (2017) created a model to predict cyberattacks. The study used a hybrid K-Nearest Neighbour classification method called Ant Colony Optimization (ACO) to identify attacks. However, the model's performance was measured using only FAR and classification accuracy. At the end of the experiment, the results showed a low false alarm rate and a classification accuracy of 94.2%. Shakil and Farid (2014) explained that choosing the right features can improve a model by reducing the number of input features in the training data. They also showed how the number of features chosen can impact a model's work, especially in Intrusion Detection Systems (IDS). The method included three groups of important features using Correlation-based feature selection and the SVM classifier on the NSLKDD dataset. The goal was to find out which number of features create the best model. The test results showed that choosing 36 features worked just as well as choosing 41 features, with both getting 99% accuracy in classification. On the other hand, using only three features got 91% accuracy.

Sharifi *et al*. (2015) created two models for detecting intrusions using K-Nearest Neighbours (KNN) with the NSL-KDD dataset. They used Principal Component Analysis to choose important features, selecting only ten from the whole dataset. Also, the study used two different situations to check and compare the models. In one situation, no test data was included in the training set, and the other, some test data was included in the training set. They only looked at how accurate the models were to measure their performance. At the end of the experiment, both situations achieved an accuracy of 90%. Matin and Rahardjo (2018) suggested a design to predict malware attacks using a honeypot to collect data and a machine learning system to classify the data. The study suggested using two methods, SVM and DT, separately on honeypot data to make a model that can tell the difference between malware and good software. They used use a 90:10 split of the data, which will help ensure the results are reliable and improve accuracy in classification. Relang and Patil (2015) created a model to predict network intrusions. They used two decision tree algorithms called C4.5 and C45 DTWP (which cut back unnecessary parts) on the KDD Cup 99 and NSL-KDD data sets. The study used the Information Gain method to choose important features, focusing only on the distinct features for classification. We looked at how well the models worked by checking their accuracy and how often they made mistakes.

Moore *et al.* (2017) introduced a method that used classification and feature reduction to detect threats in cyber networks. This method applied the Artificial Neural Network (ANN) classification algorithm to analyze network traffic data from the Department of Defense's Cyber-Defense Exercises (CDX). After removing less important features using a signal-to-noise ratio, we extracted 248 features, which were then reduced to 18. The researchers looked at different features when analyzing the data. The results showed that using 18 features worked well, achieving 97.29% accuracy with a low false alarm rate of 2.71%. In contrast, using all 248 features resulted in 82.56% accuracy but a higher false alarm rate.

Muhammad et al. (2016) created a system to detect intrusions in networks as they happen, using Apache Storm. This project uses the Support Vector Machine (SVM) technique on live data from the KDD Cup 99 dataset. The system can handle 13,600 packets every second on one computer, and it works correctly 92.60% of the time when tested. Although this study provided performance measurements for one machine, it has not been tested on a system with multiple machines to check how well it works. Not having a shared space is the important part that is missing.

Mustapha et al. (2018) used Apache Spark and MLlib to check how well different machine-learning methods could detect intrusions. They tested four algorithms: Support Vector Machine (SVM), Naïve Bayes, Decision Tree, and Random Forest, using a dataset called UNSW-NB15. Their study shows that Random Forest gives the best results with an accuracy of 97.49%, sensitivity of 93.53%, and specificity of 97.75%. Next comes Decision Trees, and Naïve Bayes had the lowest accuracy at 74.19%. This work used Apache Spark, but only for batch processing and not for stream processing to sort data.

Terzi et al. (2017) developed a new method for finding unusual patterns without specific training and used Apache. Use Spark on Microsoft Azure (HDInsight) to take advantage of Spark's ability to process data efficiently. The new method was checked using the CTU-13 dataset containing botnet traffic and got 96% accuracy. The downside of this work is that it cannot find unusual activity that looks like normal traffic. NetFlow data used in their method is usually collected by internet service providers (ISPs) to check performance and for audits.

Gupta et al. (2016) created a new system for detecting intrusions using Spark technology. They used two methods to choose important features: one based on correlation and another using Chisquared. To check their performance, they used five types of Machine Learning methods (Logistic Regression, SVM, Naive Bayes, Random Forest, and GB Tree) on the NSL-KDD and DARPA 1999 datasets. Even though they used Spark's batch processing for their tasks, their results revealed that the Random Forest classifier gave the best accuracy but took the longest time to make predictions. On the other hand, Naïve Bayes had the lowest accuracy but was quicker in training and making predictions. Unfortunately, using the DARPA dataset, which is quite old and has repeated and fake network traffic data, causes incorrect predictions. As far as we know, the shared test setup is not available.

Key Considerations for Spotting Intrusions

With a better security monitoring system, we can do more than just find security breaches in computer systems. This system could be improved to stop security breaches by working with tools like Intrusion Prevention Systems (IPSs) and using a "Defense in Depth" approach (Information Assurance Solutions Group, 2015). An IPS needs to find problems almost instantly. This study is not just about real-time Intrusion Detection. It also looks at offline investigation and security analysis.

This survey differs from past Intrusion Detection surveys because it focuses on combining security sensor data from various systems and devices. The goal is to make security alerts more accurate. We also look at the big data problems that arise when working with different types of security data (Zuech et al., 2015).

In the early days of computers, system administrators kept an eye on security by looking at the log files of their servers. In the 1980s, people came up with an Intrusion Detection System. This special device watches for unusual activity on a network or computer. In 1987, Denning created an important research paper often seen as the starting point for studying intrusion detection systems (IDS). A good example of an IDS is Snort, which is a popular and well-known open-source IDS (Sourcefire, 2015; Roesch, 1999).

Intrusion Detection is a busy field of study that has significant effects. The Center for Strategic and International Studies and McAfee (2013) conducted a study on money lost due to cybercrime and spying online. They estimated that the United States might lose around $100 billion yearly. They think the total losses worldwide could be about $300 billion yearly. In 2012, the Verizon RISK Team worked with over 250 clients and discovered over 47,000 security problems. They found that people inside the organization caused 92% of data breaches. A study by the Ponemon Institute (2012) found that the most costly type of cybercrime was "Detection," which comprised 26% of the costs. The other types, in order of expense, were Recovery, Ex-post Response, Containment, Investigation, and Incident Management. These studies show that cybersecurity, especially Intrusion Detection, greatly affects the economy.

Julisch and Dacier (2002) explain that Intrusion Detection Systems (IDS) often create a lot of false alarms. They can give off thousands of alerts daily, up to 99% of which may be false alarms. Because of this, security analysts can become desensitized to many of these meaningless warnings. Xu and Ning say that when it comes to finding attacks, intrusion detection systems (IDSs) are usually not perfectly accurate and can miss some attacks, which are called false negatives.

According to Suthaharan and Panchagnula (2012), Intrusion Detection is mainly about handling a lot of data. They say the main difficulty is managing the huge amount of network traffic data collected to detect intrusions. Bhatti et al. (2012), it was noted that today's technologies struggle to handle the challenges of detecting intrusions in large amounts of data. The authors explained that analyzing security in a big data setting comes with special problems that the current security monitoring systems, which usually rely on several traditional data sources like firewalls and intrusion detection systems, do not effectively solve. A study by Enterprise Strategy Group in late 2012 showed that 44% of big companies now think of their security analytics as "big data." Another 44% believe that in the next two years, their security analytics needs will also be seen as "big data." Finding intrusions can be a big problem with large amounts of data.

How to Keep Big Data Environments Safe from Intrusion

Kumar et al. (2019) provided some basic steps to protect users and networks in industrial settings from cyber-attacks:

For the network:

Importance of data and commands: Keeping data safe and correct is very important because it can impact the way factories work, the readings from AMI meters, and the commands sent to machines.

Protection from DoS/DDoS attacks: DoS attacks overload the network resources, which means they use too much of the network's ability to function. They can overload the system by sending fake requests to the server or the entire network, affecting how well it works (like speed and capacity). On the other hand, Distributed DoS (DDoS) attacks happen when attackers use many devices they have taken control of, like smart meters, firewalls, routers, and even household appliances, to attack one specific target. A DDoS attack is a serious threat to industrial networks and is difficult to avoid (Yan et al., 2018). For example, having access to pricing information and reliable power is important for smart grid systems. So, it's necessary to ensure these factors are reliable—providers must set fair prices, and consumers need a steady power supply.

For the users:

Keeping information private: The data about how the services from the industrial network provider are used should remain private. For example, in a smart grid system, a business can get information about how much electricity it uses in the short and long term. An industrial company should keep its information secret to protect its production secrets from being stolen by competitors.

Keeping user information private: Customers' personal information, like their names and addresses, should not be shared with anyone outside the company. No one should learn about the individual users of the network without their permission. The most sensitive personal information includes ID numbers, passport numbers, online banking passwords, credit card details, and other financial information (Butun, 2017). In smart grid systems, data about how much electricity is used at different times can be private because it might show personal things, like whether someone is home or awake. Network operators, like electricity companies in smart grid systems, should keep sensitive user information safe from people who should not see it. Future smart grid systems must follow the EU's new General Data Protection Regulation (GDPR). This means they should tell people how their data is collected and get their permission. They should also have clear rules about storing and handling data. The utility center needs to know how much energy everyone uses for billing. However, it should keep daily usage details based on people's privacy preferences (Abdallah & Shen, 2018).

Privacy and Confidentiality Goals: It is really important to keep business customers' information private and to protect the personal information of individual consumers in industrial networks. After the GDPR law, privacy is very important. If broken, network operators, like utility providers in smart grid systems, can be expensive. When setting up industrial networks, the people in charge should focus on these privacy and confidentiality goals (Kumar et al., 2019):

- **Anonymity**: A user should not be recognizable among a group of people.
- **Unlinkability**: No consumption data should be connected to the customer after the billing service.
- **Hard to detect**: People trying to cause harm should not be able to see the consumption data.
- **Unobservability**: Someone outside should not be able to see if certain important messages or actions, like sending usage messages or bidding messages, are happening or not.
- **Pseudonymity**: In smart grid communication, people might want to see the usage data from smart meters. So, a smart meter should have a fake name or identifier to protect privacy. Only the specific groups or people talking or sending messages to the smart meter can use these special ID numbers.

A Selection of Different Types of Advanced Intrusion Detection Systems

This section summarizes different types of Intrusion Detection systems found in research that work with different data sources. Since the last part showed the importance of using different sources for better cybersecurity, this section will examine the design problems researchers have found in these systems. Here are five different examples of designs suggested by researchers to handle various event sources.

A study by Fessi et al. (2010) looks at how to detect intrusions from different types of sources. A simple example of this is shown in Figure 3. In this example, several "Observers" collect information from different sources, like network monitoring and checks on individual computers. Then, a "Global Analyzer" decides if the events reported by the "Observers" are actual security problems. To make its final decision, the "Global Analyzer" will combine information from different "Analyzers" to better understand the situation, especially during widespread attacks. One interesting part of this model is that the "Analyzers" can be of different types. For example, we can simultaneously use different kinds of "Analyzers," like those that detect misuse or spot unusual behavior, for the same events observed. In short, each observer can be linked to one or more "Analyzers" to help find different types of attacks. This model can handle large amounts of data effectively because features allow adding more "Observers" and "Analyzers" to improve capacity. However, if there is just one main "Global Analyzer," it could slow things down when there are many tasks.

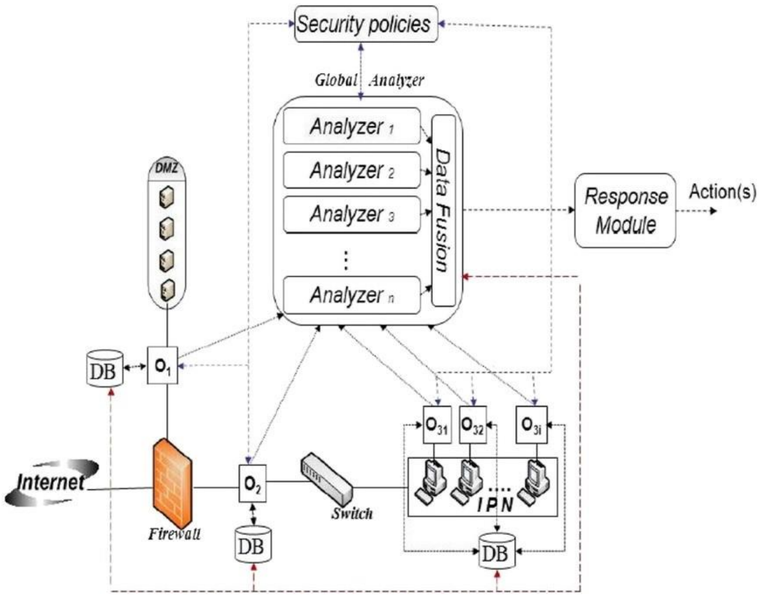

Figure 3: System Architecture (Fessi, et al., 2010).

Big Data can be useful but can also cause problems if it gets hacked or has trust issues.

Ganame et al. (2008) built on their previous work by creating a centralized Security Operation Center (SOC) named SOCBox to look at Intrusion Detection from a more worldwide perspective. They then developed an improved version called the Distributed Security Operation Center (DSOC). Their design helps an organization expand the system online, making connecting and sharing information easier across different locations. This also improves protection if one site is attacked. This design could also work for many different companies.

One reason Ganame et al. (2008), the original centralized Security Operations Center (SOC) setup was changed to something called a Distributed SOC (DSOC). This was done because attackers could target and overwhelm one location with too much traffic, preventing the centralized SOC from seeing all the security alerts. As a result, attackers could avoid being caught. They showed a few examples of how to weaken the SOC with "flood" attacks, proving that the original SOC setup could be affected by these attacks and had a big volume issue. The DSOC solved this problem using a Local Analyzer (LA) at each location. The LA checks for security threats by gathering, analyzing, and connecting security alerts where it is located. Each LA would send a simpler and clearer set of alerts to a Global Analyzer (GA). The GA would then combine and analyze alerts from all LAs to improve understanding of possible intrusions. The GA can also have backup systems to ensure reliability.

Ganame et al. (2008) explained that using different information sources helps link events from various places, which is important for successfully spotting attacks. For example, many similar network intrusion detection systems (NIDS) might miss detecting certain complicated attacks in several steps. They showed through experiments that using different sources better detects intrusions than the usual single-type systems like NIDS, especially when dealing with more complex attacks. The DSOC system uses different sources to monitor all network parts, like "IDS, IPS, firewalls, routers, workstations," and so on. This helps to understand better what is happening in the network. Look at Figure 4 for examples of different sources that a Local Analyzer might use. The system uses protocol and application agents to help gather information clearly from the original events. It does this in a way that prevents data loss and keeps the information secure.

Another interesting point they discussed was the need for the same message formats to be used across different devices and systems, like the Intrusion Detection Message Exchange Format (IDMEF). They discovered that the XML bus used for IDMEF was "too big and took up too many resources," especially when trying to connect events. The authors created a different way to translate and tackle this fast-moving problem.

This study shows that Big Volume faced two main problems. First, their original system was at risk of "flood" attacks. Second, they could not use the regular IDMEF format because it did not work well for matching events. Using different types of data sources led to better detection accuracy than using the same data type in some situations.

Bye et al. (2010) introduced a teamwork-based system to detect intrusions called a Collaborative Intrusion Detection System (CIDS). In 2010, several "participants" like intrusion detection systems (IDSs) formed teams. They worked together to improve how they detect intrusions. As IDS technology has become more common, using several IDS systems in the same area has also increased. A CIDS system lets different IDSs work together as a team. This helps everyone see the bigger picture by working together. The authors introduce the Collaborative Intrusion Detection Framework (CIDF), which works with various sources. They use several methods to help different intrusion detection systems (IDSs) work together to detect or analyze threats.

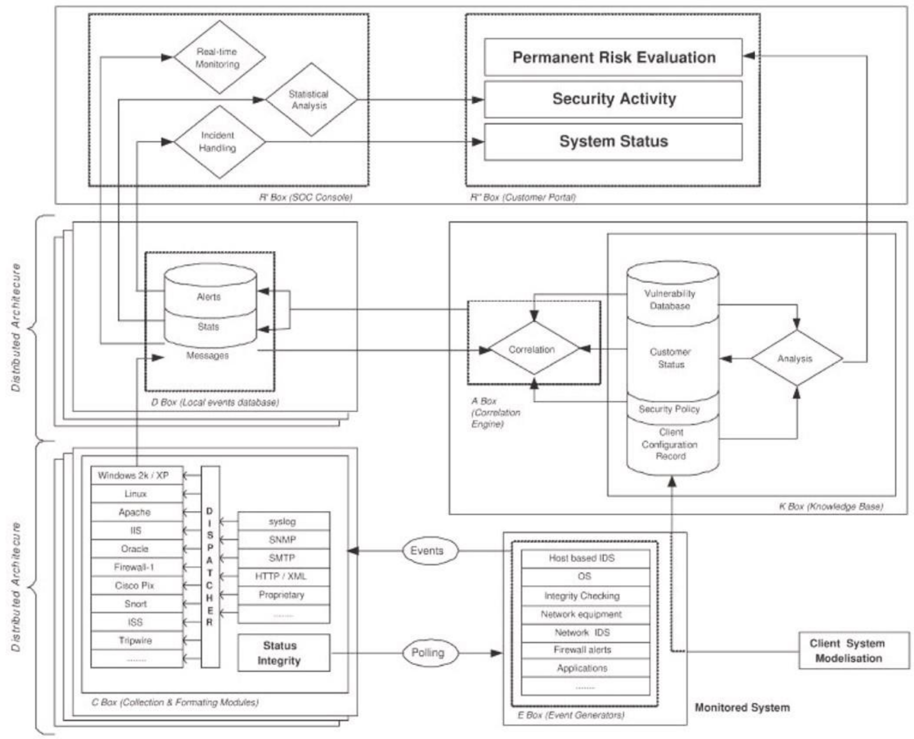

Figure 4: The internal architecture of a local analyzer (Ganame, et al., 2008)

An "agent" is a person involved in the CIDF who is part of a "detection group," which might also have other agents. Some agents work toward the same goal, like finding unusual patterns, while others have different goals, like catching misuse. This study is important because the authors clearly explain how CIDSs work and how to handle more complicated problems like security when working together. The authors also provide examples of different sources being used, like DSHIELD. Another interesting part is the variety within the framework, not just the different sources of events. Members of a group can have different kinds of "agents" (like using various Intrusion Detection Systems), and even the "detection groups" can have different roles in finding threats.

Bartos and Rehak (2012) suggested a new system called a Distributed Intrusion Detection System (DIDS) to fix a big problem with regular intrusion detection systems (IDSs), which is that they work alone. Their main goal is to improve accuracy and find more dangers. Importantly, their suggested DIDS can work with different types of sources, and they provide examples of various event sources in their study. The distributed IDS nodes are called "sensors." They can combine data from different events, even if the events are in different formats. Generally, every "sensor" IDS can talk to other sensors in the network. This is done to have backup and to be better protected against attacks. Each IDS "sensor" can adjust itself to focus on detecting specific types of attacks better while depending on other IDS "sensors" to look for different types of attacks. The IDS "sensors" can ask for help when they find suspicious behavior. Bartos and Rehak experimented on the suggested design and found that they could make detection more accurate without increasing the number of false alarms. Their research is interesting because combining data from different sources can improve detection accuracy. However, it's also noteworthy that it didn't reduce false alarms, even though their design has a broader perspective. Testing how well this method works on a bigger scale would be interesting. However, it is a new way of using distributed IDSs with different data sources.

Cai and Wu (2010) talk about using software agents to check the security of computers. These agents keep an eye on important information on the computer, like files, logs, and the core part of the system. They talk about the parts of NIDS, which shows how using different types of information helps analysts see a wider picture of what is happening on their entire network. Cai and Wu talk about how helpful it is to connect IDS alerts from different places on the Internet, similar to what Ganame et al. (2008) and Bartos and Rehak also shared the same worldwide idea about Intrusion Detection in 2012. Other studies, like those by (Zhou et al., 2009; Metzger et al., 2011), suggest that sharing warnings across different regions is an important strategy for protecting businesses online. These studies show that using various sources can improve the ability to detect intrusions by helping to connect events better and understand cyber threats more clearly.

Advantages of Advanced-Data Pipelines

Real-time data systems across the industry help improve efficiency and find risks. These pipelines help to gather, process, and analyze data without stopping, allowing organizations to meet strict efficiency and security standards.

Better detection accuracy

Machine learning can make real-time data pipelines more accurate at detecting things. These algorithms quickly look at large amounts of data to find complicated patterns and unusual things that could be harmful (Daniel & Jacob, 2017). Machine learning programs can effectively find good and bad actions because they learn from past information. Finding threats early in cybersecurity can help stop big problems. Machine learning programs can detect small clues of advanced cyberattacks or insider threats that regular methods might overlook. Models that learn and adjust to new dangers help find threats better and give organizations a way to protect themselves from cyberattacks before they happen (Tejedor et al., 2017).

Better Work Efficiency

Real-time data pipelines make work faster and better in many businesses. These pipelines easily send data between different parts, giving quick updates on what is happening. A steady data flow helps us use resources better, save money, and improve processes. Live data streams can help find problems, machine breakdowns, and slowdowns in manufacturing. Manufacturers can work better and save money by quickly fixing these problems and minimizing downtime. Real-time data pipelines help banks find fraud and keep an eye on transactions. This reduces losses and helps them follow the rules (Lee & Lee, 2012).

Growing and Adapting

Real-time data pipelines are useful and work well because they can easily grow and adapt. These pipelines can be adjusted to fit the needs of an organization and the types of data they use because they work with different kinds and amounts of data. To handle more work easily, we need automation to lessen human errors and keep our data safe. When there is a lot of data to handle, automated systems can change their size to manage it better (Krishnan & Raajan, 2016). This helps make things more dependable and quicker. These pipelines can connect to messy social media posts and organized databases because they are flexible. Groups that manage and compile different data sources for analysis need to change. Real-time data pipelines can look at how cities change using tools that check the environment, public transport systems, and traffic sensors in smart cities. This helps share resources and make smart decisions (Kohol, 2017).

Real-time data pipelines are great for fast analysis and quick reactions. With this information, businesses can quickly find and fix dangers to lower risk and keep important data safe. Analyzing data quickly helps to lower the chances of attacks by speeding up the response to them (Moore, et al., 2017). Keeping an eye on network activity with live data streams helps find unauthorized access and stolen information. Real-time alerts help security teams fix issues before they get worse. Getting information quickly can help patients improve, so healthcare needs to act in real time. Watching vital signs in real-time can warn doctors about important changes in a patient's condition. This allows them to act quickly and possibly save the patient's life (Rai, et al., 2016).

Problems with Setting up Advanced Data Pipelines

Building a real-time data pipeline must tackle problems to ensure the system works well. Problems include roads and buildings, costs, data rules, connecting different systems, and safety.

Data Security and Privacy

Live data pipelines can have problems with privacy and security while they are being processed. Real-time systems work with important information, which puts them at risk of being hacked. Keeping sensitive data safe is very important, and we need to control who can access it. To keep data safe when it is stored and when it is being sent, we need to use strong encryption (Gao et al., 2018). Control data access only allows users to see it to reduce internal risks. We need to keep an eye on the data pipeline all the time. Strange changes in real-time data monitoring could show a security weakness. Complicated systems with different kinds of data can make these security tasks hard. Companies find it hard to balance fast data transfer with keeping that data safe. We need a good mix.

Integration with Existing Systems

Adding real-time data streams to current systems is hard because they are complicated. Many companies use outdated data processing that does not happen in real time. Connecting old systems with real-time features can be difficult and take time. To integrate, we need to sort out problems with how data moves and how well the systems work together. However, personalized solutions and big infrastructure upgrades can be expensive and require many resources. Data must move smoothly between new and old systems without any breaks or problems for everything to work well together. We need to plan carefully and test everything to find and fix any problems with how things work together. It is harder to combine data when there are many sources and types, and each has different needs (Relan & Patil, 2015).

Cost and Facilities

Setting up and monitoring real-time data systems can be costly for small companies with few resources. Buying high-quality servers, storage, and networks costs much money (Khan et al., 2016). Real-time data pipelines require special software and skilled workers to manage and maintain them. Hiring new employees or training staff might raise prices. Besides setting it up, a real-time data pipeline requires updates to the system, regular checks, and security measures. Smaller organizations might find it hard to explain these costs if getting fast data updates takes a long time. Money problems might make it hard to start using real-time data systems (Shakil & Farid, 2014).

Data Standardization

Ensuring data formats and protocols are the same helps real-time data systems handle information from different sources (Sharifi, et al., 2015). The requirements might be hard to meet. Many places give information to organizations in various formats and ways. Data needs to be uniform before it can be combined and analyzed, which could slow things down. Delays and mistakes in converting data can reduce the advantages of processing data in real time. Data standardization requires collaboration from different groups with goals and rules. Big companies or those with inconsistent data rules might face issues. Organizations must collaborate to create and use common data formats and standards (Sharifi, et al., 2015).

Conclusion

The rapid evolution of technology and the increasing reliance on digital systems have made organizations more vulnerable to cyber threats. As highlighted throughout this paper, the need for effective intrusion detection systems (IDS) has never been more pressing. The integration of advanced data pipelines into the cybersecurity framework offers a promising solution to enhance the scalability and efficiency of intrusion detection in big data environments.

The study has explored various aspects of intrusion detection, emphasizing the importance of leveraging big data analytics to identify and mitigate potential threats. Traditional IDS often struggle to keep pace with the volume, velocity and variety of data generated in modern digital ecosystems. However, by employing advanced data pipelines, organizations can process vast amounts of data in real-time, enabling them to detect anomalies and potential intrusions more effectively.

One of the key findings of this research is the significance of utilizing machine learning algorithms within data pipelines. Techniques such as decision trees, support vector machines, and ensemble methods have shown promising results in improving detection accuracy and reducing false positives. The ability to continuously learn from new data allows these systems to adapt to evolving threats, making them more resilient against sophisticated cyber-attacks.

Moreover, the paper underscores the necessity for collaboration among organizations to establish common data formats and standards. As cyber threats become increasingly complex, a unified approach to data sharing and analysis can enhance the overall effectiveness of intrusion detection efforts. By working together, organizations can create a more comprehensive understanding of the threat landscape, leading to improved detection capabilities and a stronger defense posture.

The implementation of advanced data pipelines also facilitates the integration of various data sources, including network traffic, user behavior, and system logs. This holistic view of the environment enables security teams to correlate data from multiple sources, providing deeper insights into potential threats. Additionally, the use of cloud-based solutions, such as Apache Spark on platforms like Microsoft Azure, allows for scalable processing power, further enhancing the capabilities of intrusion detection systems.

Despite the advancements in technology, challenges remain. Organizations must invest in regular updates and maintenance of their IDS to ensure they remain effective against emerging threats. Furthermore, the complexity of setting up and managing these advanced systems requires skilled personnel who can navigate the intricacies of big data analytics and cybersecurity.

References

[1] Ar, L.; Levent, E.; Vipin, K.; Aysel, O.; Jaideep, S. (2003) A comparative study of anomaly detection schemes in network intrusion detection. In Proceedings of the SIAM Conference on Applications of Dynamical. Systems.

[2] Bartos K, Rehak M (2012) Self-organized mechanism for distributed setup of multiple heterogeneous intrusion detection systems. In: Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2012 IEEE sixth international conference on. IEEE, Lyon, France. pp 31–38

[3] Ben Fekih, R., & Jemili, F. (2018) Distributed Architecture of an Intrusion Detection System Based on Cloud Computing and Big Data Techniques.

[4] Bhatti R, LaSalle R, Bird R, Grance T, Bertino E (2012) Emerging trends around big data analytics and security: Panel. In: Proceedings of the 17th ACM Symposium on Access Control Models and Technologies. SACMAT ’12. ACM, New York, NY, USA. pp 67–68. doi:10.1145/2295136.2295148. http://doi.acm.org/10.1145/2295136.2295148

[5] Butun, I. (2013). Prevention and detection of intrusions in wireless sensor networks.

[6] Butun, I. (2017, January). Privacy and trust relations in internet of things from the user point of view. In *2017 IEEE 7th annual computing and communication workshop and conference (CCWC)* (pp. 1-5). IEEE.

[7] Butun, I., Morgera, S. D., & Sankar, R. (2013). A survey of intrusion detection systems in wireless sensor networks. *IEEE communications surveys & tutorials*, *16*(1), 266-282.

[8] Bye R, Camtepe SA, Albayrak S (2010) Collaborative intrusion detection framework: characteristics, adversarial opportunities and countermeasures. In: Proceedings of CollSec: Usenix Workshop on Collaborative Methods for security and privacy. USENIX, Washington, DC, USA

[9] Cai H, Wu N (2010) Design and implementation of a dids. In: 2010 IEEE International Conference on Wireless Communications, Networking and Information Security. IEEE, Beijing, China. pp 340–342

[10] Center for Strategic and International Studies (2013) The economic impact of cybercrime and cyber espionage. Technical report. McAfee http://www.mcafee.com/us/resources/reports/rp-economic-impact-cybercrime.pdf

[11] Daniel, N., & Jacob, R. (2017) Intrusion Detection Techniques in Big Data: A Review.

[12] Dataricks. About Databricks. Available online: https://databricks.com/ spark/about (accessed on 6 May 2018).

[13] Debar, H., Dacier, M., & Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. *Computer networks*, *31*(8), 805-822.

[14] Debar, H., Dacier, M., & Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. *Computer networks*, *31*(8), 805-822.

[15] Essid, M., & Jemili, F. (2016, October). Combining intrusion detection datasets using MapReduce. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 004724-004728). IEEE.

[16] Fauri, D., dos Santos, D. R., Costante, E., den Hartog, J., Etalle, S., & Tonetta, S. (2017, November). From system specification to anomaly detection (and back). In *Proceedings of the 2017 workshop on cyber-physical systems security and privacy* (pp. 13-24).

[17] Fessi B, Benabdallah S, Hamdi M, Rekhis S, Boudriga N (2010) Data collection for information security system. In: Engineering Systems Management and Its Applications (ICESMA), 2010 second international conference on. IEEE, Sharjah, United Arab Emirates. pp 1–8

[18] Fran
k J (1994) Artificial intelligence and intrusion detection: current and future directions. In: Proceedings of the 17th national computer security conference. Vol. 10. Citeseer, Baltimore, MD, USA. pp 1–12

[19] Gao P, Xiao X, Li D, Li Z, Jee K, et al. (2018) {SAQL}: A stream-based query system for {Real-Time} abnormal system behavior detection. In 27th USENIX Security Symposium (USENIX Security 18) pp 639-656.

[20] Hafs
a, M., & Jemili, F. (2018). Comparative study between big data analysis techniques in intrusion detection. *Big Data and Cognitive Computing*, *3*(1), 1.

[21] Information Assurance Solutions Group (2015) Defense in depth. Technical report, National Security Agency. http://www.nsa.gov/ia/_files/support/defenseindepth.pdf. Accessed 2015-1-10

[22] Jeong H, Hyun W, Lim J, You I (2012) Anomaly teletraffic intrusion detection systems on hadoop-based platforms: A survey of some problems and solutions. In: Network-Based Information Systems (NBiS), 2012 15th international conference on. IEEE, Melbourne, Australia. pp 766–770

[23] Julisch K, Dacier M (2002) Mining intrusion detection alarms for actionable knowledge. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, Edmonton, Alberta, Canada. pp 366–375

[24] Karim Ganame A, Bourgeois J, Bidou R, Spies F (2008) A global security architecture for intrusion detection on computer networks. Comput Secur 27(1):30–47

[25] Khan R, Maynard P, McLaughlin K, Laverty D, Sezer S (2016) August. Threat analysis of blackenergy malware for synchrophasor based real-time control and monitoring in smart grid. In 4th International Symposium for ICS & SCADA Cyber Security Research BCS pp 53-63.

[26] Kohol, V. I. (2017). *Applying Emerging Data Techniques and Advanced Analytics to Combat Cyber Threat* (Doctoral dissertation).

[27] Krishnan, R. B., & Raajan, N. R. (2016). An intellectual intrusion detection system model for attacks classification using RNN. *Int. J. Pharm. Technol*, *8*(4), 23157-23164.

[28] Lee, Y., & Lee, Y. (2012). Toward scalable internet traffic measurement and analysis with hadoop. *ACM SIGCOMM Computer Communication Review*, *43*(1), 5-13.

[29] Mabayoje M., Abimbola A., Balogun A., and Opeyemi A. (2015) “Gain Ratio and Decision Tree Classifier for Intrusion Detection,” *International Journal of Computer Applications*, vol. 126, no. 1, pp. 56-59. DOI: 10.5120/ijca2015905983

[30] Massimiliano, A.; Erbacher, R.F.; Jajodia, S.; Persia, M.C.F.; Picariello, A.; Sperli, G.; Subrahmanian, S.V. (2013) Recognizing unexplained behavior in network traffic. Netw. Sci. Cybersecur.

[31] Microsoft. Azure Regions. Available online: https://azure.microsoft.com/enus/global-infrastructure/ regions/ (accessed on 5 May 2018).

[32] Moore, K. L., Bihl, T. J., Bauer Jr, K. W., & Dube, T. E. (2017). Feature extraction and feature selection for classifying cyber traffic threats. *The Journal of Defense Modeling and Simulation*, *14*(3), 217-231.

[33] Moustafa, N., & Slay, J. (2015, November). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 military communications and information systems conference (MilCIS)* (pp. 1-6). IEEE.

[34] Myers, S., Musacchio, J., & Bao, N. (2010). Intrusion Detection Systems: A Feature and Capability Analysis. *Baskin School of Engineering: Santa Cruz, CA, USA*. Gupta, G. P., & Kulariya, M. (2016). A framework for fast and efficient cyber security network intrusion detection using apache spark. *Procedia Computer Science, 93*, 824-831.

[35] Oguntimilehin, A., & Ademola, E. O. (2014). A review of big data management, benefits and challenges. *A Review of Big Data Management, Benefits and Challenges*, *5*(6), 1-7.

[36] Ponemon Institute LLC (2012) 2012 cost of cyber crime study: United states. Technical report. Ponemon Institute http://www.ponemon.org/local/upload/file/2012_US_Cost_of_Cyber_Crime_Study_FINAL6%20.pdf

[37] Rai, K., Devi, M. S., & Guleria, A. (2016). Decision tree based algorithm for intrusion detection. *International Journal of Advanced Networking and Applications*, *7*(4), 2828.

[38] Relan, N. G., & Patil, D. R. (2015, January). Implementation of network intrusion detection system using variant of decision tree algorithm. In *2015 International Conference on Nascent Technologies in the Engineering Field (ICNTE)* (pp. 1-5). IEEE.

[39] Rhodes B.C., Mahaffey J.A. and Cannady D.J. (2000) "Multiple Self-Organizing Maps for Intrusion Detection," in National Information Systems Security Conference, Baltimore.

[40] Roesch M (1999) Snort: Lightweight intrusion detection for networks. In: LISA. Vol. 99. USENIX, Seattle, WA, USA. pp 229–238

[41] Rustam, Z., & Ariantari, N. P. A. A. (2018, October). Comparison between support vector machine and fuzzy Kernel C-Means as classifiers for intrusion detection system using chi-square feature selection. In *AIP Conference Proceedings* (Vol. 2023, No. 1). AIP Publishing.

[42] Shakil P., and Farid D. (2014) "Feature Selection and Intrusion Classification in NSL-KDD Cup 99 Dataset Employing SVMs," *in Proceedings of the 8th International Conference on Software, Knowledge, Information Management and Applications*, Dhaka, pp. 1-6.

[43] Sharifi, A. M., Amirgholipour, S. K., & Pourebrahimi, A. (2015). Intrusion detection based on joint of k-means and knn. *Journal of Convergence Information Technology*, *10*(5), 42.

[44] Sourcefire (2015) Snort, Home Page. http://www.snort.org/. Accessed 2015-1-10

[45] Suthaharan S, Panchagnula T (2012) Relevance feature selection with data cleaning for intrusion detection system. In: Southeastcon, 2012 Proceedings of IEEE. IEEE, Orlando, FL, USA. pp 1–6

[46] Tejedor J, Macias-Guarasa J, Martins HF, Pastor-Graells J, Corredera P, et al. (2017) Machine learning methods for pipeline surveillance systems based on distributed acoustic sensing: A review. Applied Sciences 7: 841.

[47] Terzi, D. S., Terzi, R., & Sagiroglu, S. (2017, October). Big data analytics for network anomaly detection from netflow data. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 592-597). IEEE.

[48] Terzi, D. S., Terzi, R., & Sagiroglu, S. (2017, October). Big data analytics for network anomaly detection from netflow data. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 592-597). IEEE.

[49] Virvilis-Kollitiris, N. (2015). Detecting advanced persistent threats through deception techniques. *Ph. D. dissertation, Information Security and Critical Infrastructure Protection (INFOSEC) Laboratory*.

[50] Vishwakarma, S., Sharma, V., & Tiwari, A. (2017). An intrusion detection system using KNN-ACO algorithm. *Int J Comput Appl*, *171*(10), 18-23.

[51] What is Microsoft Azure and Why Use It? Available online: https://www.sumologic.com/resource/whitepaper/what-is-microsoft-azure-and-why-use-it/ (accessed on 5 May 2018).Microsoft. Azure Storage Blobs. Available online: https://azure.microsoft.com/en-us/services/storage/ blobs/ (accessed on 8 December 2018).

[52] Zikopoulos, P., Deroos, D., Parasuraman, K., Deutsch, T., Giles, J., & Corrigan, D. (2012). *Harness the power of big data The IBM big data platform*. McGraw Hill Professional.

[53] Zubair, N. (2016) *Pro Spark Streaming the Zen of Real-Time Analytics Using Apache Spark*; Apress: Berkeley, CA, USA.

[54] Zuech, R., Khoshgoftaar, T. M., & Wald, R. (2015). Intrusion detection and big heterogeneous data: a survey. *Journal of Big Data*, *2*, 1-41.