International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (Fep-24)

https://dx.doi.org/10.12785/ijcds/150154

Big Data based approach to Network Security and intelligence

Mubarak Alquaifil[1], Shailendra Mishra[2] and Mohammed AlShehri[3]

[1,2,3]*Department of Information Technology, College of Computer and Information Sciences, Majmaah University, Majmaah, Saudi Arabia*

Received 8 May. 2023, Revised 6 Jan. 2024, Accepted 21 Jan. 2024, Published 1 Feb. 2024

Abstract: Big data analysis technologies and machine learning techniques are essential for examining and forecasting the state of network security as global concerns about cyber security grow. Models for monitoring network security have a number of challenges, including resource consumption, inaccuracies, low processing efficiency, and incompatibility with real-time and large-scale scenarios. This paper proposes a novel approach to Network Security Situation Awareness (NS-SA) using Big Data (BD) analytics and machine learning. The proposed approach addresses the limitations of existing NS-SA models by leveraging data purification and simplification techniques, and by employing an updated back propagation (BP) neural network to construct an NS BD analysis model. The paper provides a comprehensive explanation of the model's structure and outlines the relevant model techniques. Extensive testing has been conducted to ensure the model's accuracy and applicability in understanding NS scenarios. This study focuses on MATLAB and Python-based implementation of a neural network for network security using a big data approach. The results demonstrate the potential and value of the proposed model in accurately assessing and forecasting NS conditions. The proposed approach has several advantages over existing NS-SA models. It is more efficient in terms of resource usage, it is more accurate in its analysis of network data, It is more applicable in real-time and large-scale scenarios and it is more robust to noise and heterogeneity in network data.

Keywords: Network Security, Big Data Analytics, Machine Learning, BP neural network, Fuzzy C-means Clustering, Situation Awareness, Data Preprocessing, Accuracy, Cyber Security.

I. Introduction

The world has seen significant growth in digital data in recent years, which has revolutionized various fields, including Network Security and intelligence. As more and more devices are connected to the internet, there has been a massive increase in the volume of data generated and exchanged [1]. According to Statista's prediction for 2021, the global data created, captured, and consumed is expected to reach 74 zettabytes. This rapid growth has given rise to Big Data as a potent tool for utilizing the vast amount of information available to improve cybersecurity and facilitate decision-making processes in the field of Network Security. Big Data refers to the management of massive datasets that are too complex for traditional data processing tools to handle, involving collection, storage, processing, and analysis. These datasets pose challenges in terms of their volume, velocity, variety, veracity, and value, making them difficult to manage. In the realm of Network Security and intelligence, Big data analytics can offer valuable insights into identifying and mitigating potential threats, as well as developing proactive strategies to improve an organization's overall security posture. The use of BD techniques in NS and intelligence is a relatively new area of research, with scholars exploring various aspects of this interdisciplinary field. For instance, have proposed a framework for using Big data analytics to identify and counter advanced persistent threats (APTs), which are sophisticated, long- term cyber-attacks targeting specific organizations or individuals. This framework facilitates the examination of massive amounts of network traffic data to identify patterns and anomalies that could suggest the existence of an APT. Additionally, Wang, [2] have investigated the utilization of machine learning approaches, including deep learning, for analyzing network traffic data and identifying malicious activities in real-time. Big data has a considerable capability to improve the gathering of threat intelligence in network security and intelligence. Threat intelligence refers to the acquisition, analysis, and distribution of data concerning possible cybersecurity risks, weaknesses, and threats, which can aid organizations in making knowledgeable choices about their security measures [3]. Big Data analytics can be utilized by practitioners and researchers to collect and examine vast amounts of data from various sources, such as social media, news articles, and blogs, to generate practical threat intelligence. Using

E-mail address: 431104307@s.mu.edu.sa, s.mishra@mu.edu.sa, ma.alshehri@mu.edu.sa

Big Data to track mentions of vulnerabilities on social media platforms, like Twitter, to predict future cyber-attacks [4]. Despite its potential benefits, the application of Big Data in Network Security and intelligence also raises several challenges, including privacy concerns, data integration, and the need for robust, scalable, and efficient algorithms to handle massive datasets. To overcome these challenges, it is necessary to engage in interdisciplinary research efforts that include computer science, statistics, data mining, and cyber security. Additionally, novel techniques and frameworks customized to meet the specific requirements of the Network Security sector must be developed.

(**A**) Motivation

The increasing complexity of networks and the growing volume of network traffic have made it more challenging to detect and prevent network security threats. Traditional security solutions are no longer sufficient, and organizations need a more advanced approach to network security and intelligence. This is precisely where Big Data can play a crucial role. By using Big Data analytics, organizations can analyze enormous volumes of network traffic data in real-time, and thereby, identify potential security threats proactively, preventing them from causing any harm. Big Data based approaches to Network Security and intelligence offer several benefits, including improved threat detection and response times, enhanced network performance, and reduced security risks.

(**B**) Objective

The primary objective of a Big Data-based approach to Network Security and intelligence is to provide organizations with an advanced security solution that can help them detect and prevent security threats in real-time. This involves analyzing vast amounts of network traffic data to identify patterns, anomalies, and potential threats. Existing NS situation awareness models face challenges such as resource inefficiency, poor analysis accuracy, and limited applicability in real-time and large-scale scenarios. To overcome these limitations, this paper proposes an innovative approach that employs an updated BP neural network to construct an NS BD analysis model. By leveraging data purification and simplification techniques based on the inherent properties of data records, this approach addresses issues arising from heterogeneous data sources and high levels of noise. The central component of the model is a conventional BP neural network, and the analysis precision is enhanced through the utilization of the neural network's error inverse feedback strategy. Moreover, the suggested blocked fuzzy C-means clustering approach (BFCM) [5] is employed during the preprocessing stage to cluster data record features, thereby improving the model's accuracy and enhancing the characteristics of the data.

The paper provides a comprehensive explanation of the model's structure and outlines the relevant model techniques. Extensive testing has been conducted to ensure the model's accuracy and applicability in understanding NS scenarios. The results demonstrate the potential and value of the proposed model in accurately assessing and forecasting NS conditions.

The key objectives of a Big Data-based approach to Network Security and intelligence can be summarized as follows:

- Improve threat detection: By analyzing network traffic data in real-time, Big Data analytics can help organizations identify potential security threats before they can cause any damage.
- Enhance response times: Big Data-based approaches can help organizations respond quickly and efficiently to security threats, minimizing the impact of any security incidents.
- Enhance network performance: By analyzing network traffic data, Big Data analytics can help organizations optimize network performance and improve user experience.

This study focuses on MATLAB and Python-based implementation of a neural network for network security using a big data approach. By addressing the limitations of existing NS situation awareness models, the proposed model exhibits promise in accurately analyzing and predicting the state of network security. This contributes to advancing the field of network security and intelligence.

II. Literature Review

A literature review on Big Data-based approaches to Network Security and intelligence would likely cover the methods used to examine large volumes of data from multiple sources to enhance network security and intelligence. This could involve utilizing machine learning algorithms to detect irregularities and patterns in the data that may indicate potential security weaknesses, as well as the utilization of data visualization tools to assist security analysts in comprehending and interpreting complex data in a more accessible way. Real-time data processing and analysis, which allows businesses to respond more quickly and effectively to threats and vulnerabilities, is a critical aspect of Big Data-based approaches to Network Security and intelligence. The capacity to combine information from a variety of sources, such

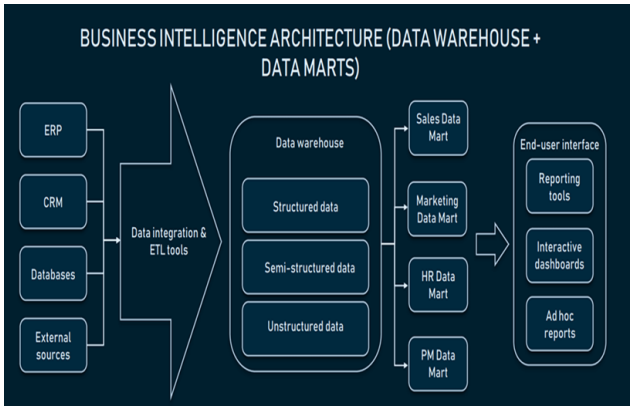

Figure 1. Business Intelligence

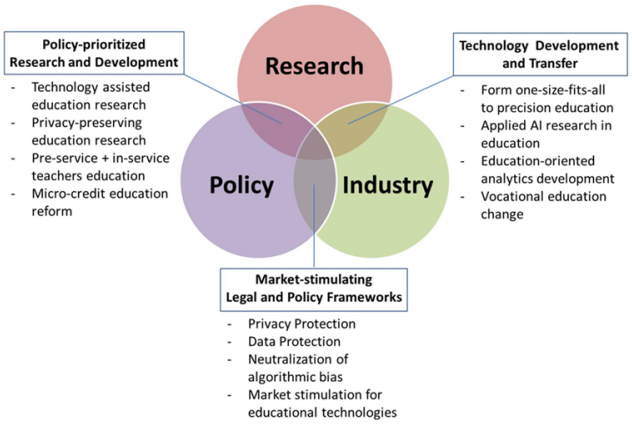

Figure 2. Benefits and Challenges of Big Data-based Approaches

as network logs, user activity data, and outside intelligence sources, can give a more thorough picture of potential threats and weaknesses. The use of Big Data in Network Security and intelligence may potentially provide difficulties and constraints. These may include the requirement for highly developed infrastructure and tools for data processing and analysis, as seen in Figure 1, as well as the requirement for qualified employees with experience in data analysis and security. Concerns over privacy and the possibility for data misuse also exist, which may need the implementation of suitable safeguards and controls by companies.

Big Data has the ability to substantially improve the security and intelligence of networks by allowing organizations to analyze vast amounts of data in real-time and integrate data from multiple sources. The objective of this literature review is to present a summary of the current research on Big Data-based methods for network security and intelligence. The review begins by discussing the key characteristics of Big Data and the ways in which it can be used to improve Network Security and intelligence. It then examines the potential benefits and challenges of these approaches, including the need for sophisticated data processing and analysis tools and infrastructure, as well as concerns about privacy and the potential for misuse of data. Big Data-based methods for Network Security and intelligence utilize advanced techniques for data processing and analysis, such as data visualization tools and machine learning algorithms, to detect potential security threats by identifying patterns and anomalies in the data.

These methods can also be used to monitor network activity in real-time and respond more rapidly and effectively to threats and vulnerabilities [4]. Benefits and Challenges of Big Data-based Approaches: There are several potential benefits to the use of Big Data-based approaches to Network Security and intelligence as shown in Figure **??**. Big Data-based approaches to Network Security and intelligence offer several advantages, including the capability to process and analyze data in real-time for prompt and efficient threat response [6]. Additionally, they provide the capability to combine data from different sources, including network logs, user behavior data, and external intelligence sources, which can provide a more comprehensive and complete view of potential threats and vulnerabilities.

However, there are also potential challenges and limitations to the use of Big Data in Network Security and intelligence. One challenge is the need for sophisticated data processing and analysis tools and infrastructure, which may be expensive and require specialized training to use effectively [7].

Additionally, there are concerns about privacy and the potential for misuse of data, which may require organizations to implement appropriate safeguards and controls [8]. One key application of Big Data in Network Security is in the detection and prevention of cyber threats as shown in Figure 3. Through the analysis of massive amounts of data from diverse sources such as network logs, traffic patterns, and threat intelligence feeds, it is feasible to detect anomalies and possible threats instantly [9]. Machine learning algorithms [10] can be used to continually learn and adapt to new threats, improving the accuracy and effectiveness of threat detection over time [9].

Big Data can also be used to improve the intelligence of networks by providing deeper insights into network behavior and performance [11]. For example, net- work administrators can use Big Data analytics to identify patterns and trends in network usage and traffic, helping them to optimize resources and improve the efficiency of the network [11]. Additionally, Big Data can be used to support decision-making by providing context and situational awareness to

Figure 3. Detection and Prevention of Cyber Threats

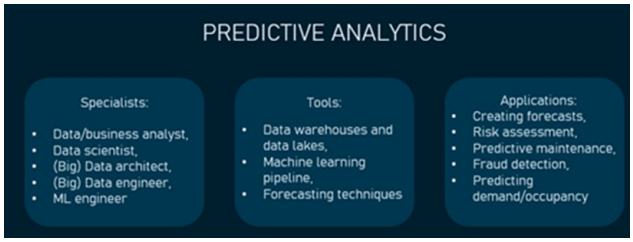

Figure 4. Predictive Analytics Capabilities

security analysts and incident responders [12]. Big Data can also be utilized in Network Security to develop predictive analytics capabilities, as illustrated. By analyzing historical data, it is possible to identify patterns and trends that can be used to forecast future events and anticipate potential threats. This can allow organizations to take proactive measures to mitigate or prevent future threats, rather than simply reacting to them after they have occurred [12].

The literature suggests that Big Data-based approaches can greatly improve Network Security and intelligence by allowing organizations to quickly detect and respond to potential threats [13]. With the help of Big Data analytics and technologies [14], organizations can gain a better understanding of their networks and potential threats [15]. This can help them take more proactive and effective measures to secure their systems and assets [16]. Big Data has also been used in Network Security and intelligence for developing models that use predictive analytics. Predictive analytics uses data and statistical algorithms to determine the possibility of future outcomes based on past data. In the context of NS, predictive analytics models can predict the likelihood of a cyber-attack or other security threat occurring, enabling organizations to take proactive measures to prevent or minimize the impact of such threats. One example of the use of predictive analytics in Network Security is the development of models to predict the likelihood of a malware attack based on factors such as the type of malware, the target network, and the characteristics of the network [16].

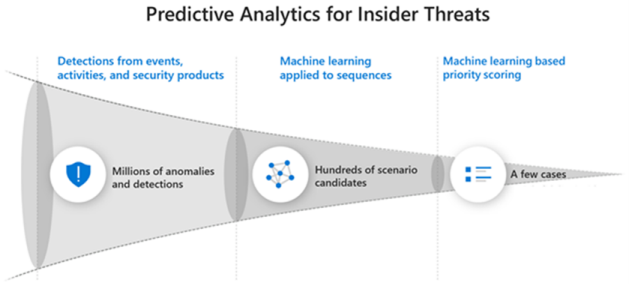

Figure 5. Insider Threat and Predictive Analytics

These models can be used to identify the most likely targets of a malware attack, as well as the most effective prevention and response strategies. Another application of predictive analytics in Network Security is in the identification of insider threats as shown in Figure 5, such as employees who may be acting maliciously or negligently [17]. By analyzing data such as employee activity logs and email patterns, organizations can identify employees who may pose a risk to the security of the network and take appropriate action to mitigate this risk. In addition to these applications, Big Data has also been used in the development of network visualization tools, which allow organizations to better understand the structure and behavior of their networks [18]. These tools are useful for identifying possible weaknesses and enabling organizations to take action to enhance the security of their networks.

The integration of Big Data in the development of predictive analytics models and visualization tools has the potential to considerably enhance the efficiency of Network Security and intelligence efforts. Nonetheless, as with any implementation of Big Data, there are also obstacles related to its usage, such as the requirement for appropriate data processing and analysis, as well as the possibility of privacy concerns [19]. Big Data is also utilized in the area of anomaly detection in Network Security and intelligence. Anomaly detection involves recognizing uncommon or unforeseen patterns in data that could imply a security threat, as depicted in Figure 6.

Another application of Big Data in anomaly detection is in the analysis of user behavior data. By analyzing the patterns of user behavior, such as login patterns and access patterns, organizations can identify unusual behavior that may indicate a security threat [20]. For example, a sudden increase in login attempts from a particular location or a change in access patterns may indicate a potential security breach. In addition to these applications, Big Data has also been used in the development of predictive analytics models for anomaly detection. By analyzing historical data, these models can identify patterns that are likely to indicate a

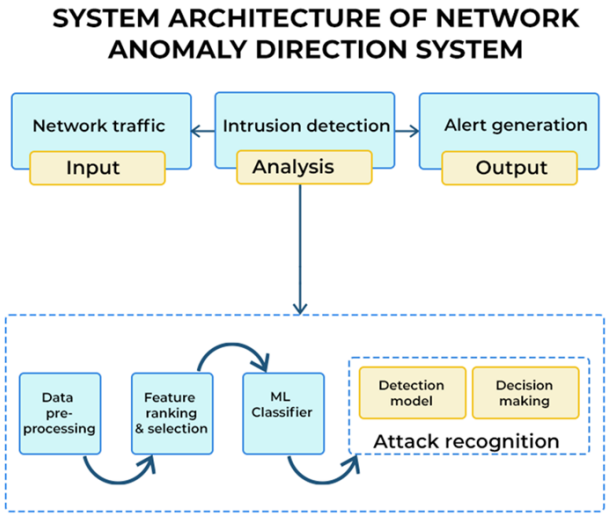

Figure 6. System Architecture of Network Anomaly Detection

security threat in the future. In summary, utilizing Big Data in anomaly detection can enhance the efficacy of Network Security and intelligence; however, there are challenges involved, such as correctly analyzing and managing large data sets, as well as avoiding false positive [2].

It's important to note that these applications (Cybersecurity, Network traffic analysis, Social media analysis, Predictive analytics, Network visualization and Anomaly detection) of Big Data in network security and intelligence are not mutually exclusive, and many organizations may use a combination of these approaches to improve the security of their networks. Additionally, there are challenges associated with the use of Big Data in these contexts, such as the need to properly process and analyze large amounts of data and the potential for privacy concerns.

The literature cited suggests that Big Data-based approaches to network security and intelligence have the potential to significantly enhance the ability of organizations to identify and respond to threats and vulnerabilities. However, it is important for organizations to carefully consider the potential benefits and challenges of these approaches, and to implement appropriate safeguards and controls to ensure the responsible and effective use of data.

The literature review emphasizes the importance of organizations care- fully considering both the advantages and difficulties of Big Data-based approaches to Network Security and intelligence. It suggests that appropriate safeguards and controls should be put in place to ensure the responsible and effective use of data. The literature indicates that Big Data techniques can greatly improve the ability of organizations to detect and address network security threats and vulnerabilities. Nonetheless, it is crucial for organizations to weigh the potential benefits and drawbacks of such methods and to implement appropriate measures to ensure the responsible and effective use of data. Organizations are seeking Big Data-based approaches to enhance the security and intelligence of their networks due to the ever-increasing volume and complexity of data. Big Data refers to massive and intricate data sets that traditional methods may find challenging to process and analyze due to their size, speed, or complexity. These data sets usually originate from diverse sources such as user behavior data, network logs, and external intelligence sources, and may be structured or unstructured.

III. Methodology

(A) Research Gap

Although Big Data has become increasingly crucial in Network Security and intelligence, there is still a considerable research gap regarding how to efficiently and effectively analyze and process vast amounts of data to identify and prevent cyber-attacks. One of the primary hurdles in this field is dealing with the sheer volume, velocity, and variety of data generated by digital devices and internet usage. Traditional security and intelligence methods like signature- based detection and rule-based systems are inadequate for effectively managing the significant volume and diversity of data generated by modern networks. Another significant gap in the current research is the ability to detect and prevent a wide range of cyber threats, including advanced persistent threats, malware, and network intrusions. While there are several techniques and algorithms that have been developed for detecting specific types of cyber threats, there is a lack of research on how to integrate these methods and apply them to large and diverse data sets. Additionally, the constantly changing nature of cyber threats requires the development of systems that can adapt to new threats in real-time, and there is a gap in the current research on how to achieve this.

Another gap in the current research is the lack of real- world evaluations of Big Data-based approaches to Network Security and intelligence. Several studies have suggested various solutions for analyzing and processing large amounts of data in network security, but there is a dearth of evaluations on the scalability, accuracy, and effectiveness of these methods in real-world scenarios. This lack of evaluation makes it challenging to assess the practicality and feasibility of these solutions. The current research has a gap in terms of effectively integrating Big Data approaches into existing Network Security and intelligence infrastructure. While some studies have focused on analyzing and

processing large amounts of data, there is a lack of research on how to integrate these solutions with the existing infrastructure to create a comprehensive and effective solution. To summarize, there is a research gap in developing a Big Data-based approach to Network Security and intelligence that can handle large volumes of data, detect a wide range of cyber threats, adapt to the constantly changing nature of these threats, provide real- time intelligence, and integrate effectively with existing infrastructure. Current research falls short in these areas and needs to be improved to provide a comprehensive and effective solution for network security and intelligence.

(**B**) Research Problem

As digital devices and internet usage continue to grow rapidly, there has been a corresponding increase in the amount of data generated and stored, which has led to the emergence of the field of Big Data. This field is characterized by large volumes of data, high velocity, and varied nature, making it challenging for traditional data processing methods to handle. Traditional Network Security and intelligence methods are also becoming insufficient due to the increasing amount of data and the sophistication of cyber threats. Therefore, there is a need for a new approach to Network Security and intelligence that can effectively analyze and process large amounts of data to detect and prevent cyber-attacks. The main research challenge for a Big Data-based approach to Network Security and intelligence is to design a system that can efficiently analyze and process vast amounts of data to detect and prevent cyber-attacks.

The system must be capable of handling the large volume, high velocity, and varied nature of data generated by digital devices and internet usage. It must also be able to identify and prevent a broad spectrum of cyber threats, such as advanced persistent threats, malware, and network intrusions. Moreover, the system must be adaptable to the ever-changing nature of cyber threats and provide real-time intelligence to help organizations and individuals stay protected. The research proposal aims to create a Big Data- based method for Network Security and intelligence that can efficiently examine and manage substantial amounts of data to detect and prevent cyber-attacks. To achieve this, there are several crucial research questions that need to be explored:

- How can large amounts of data be effectively and efficiently analyzed to detect and prevent cyber-attacks?
- What methods and algorithms are available to identify and prevent various types of cyber threats, such as advanced persistent threats, malware, and network intrusions?
- How can the system adapt to the constantly changing nature of cyber threats and provide real-time intelligence to organizations and individuals?
- How can the system be integrated into existing net- work security and intelligence infrastructure to provide a comprehensive and effective solution?

To address these research questions, the proposed research will involve a combination of techniques and methodologies from various fields including data mining, machine learning, and Network Security. The system proposed in the research will use multiple sources of data including network traffic data, system logs, and social media data to provide a complete understanding of the network and its weaknesses. The proposed research will also assess the system's effectiveness and scalability in a real-world environment, evaluating its precision and ability to handle large amounts of data. The main challenge in implementing a Big Data-based approach to Network Security and intelligence is to create a system that can efficiently and accurately process and analyze large volumes of data to detect and prevent cyber-attacks. The system should be able to handle the high volume, velocity, and variety of data generated by digital devices and the internet, and detect and prevent a diverse range of cyber threats. It should also be flexible enough to adapt to the ever-changing nature of cyber threats, providing real-time intelligence to individuals and organizations. The proposed research will focus on addressing key research questions and developing a comprehensive and effective solution that can be integrated into existing Network Security and intelligence infrastructure

(**C**) Choice of Research Methods

1) First Approach

In order to analyze network traffic data using MATLAB, machine learning algorithms may be used as one method of implementing a big data based approach to network security and intelligence. The initial step involves collecting network traffic data from a network, which can be accomplished using various tools like Wireshark / tcpdump. The collected data should be saved in a format that is compatible with MATLAB.

- Preprocessing: The following step is to pre-

process the data, which involves cleaning and transforming it. This process involves removing any unnecessary information, converting the data into numerical format, and normalizing it. The suggested blocked fuzzy C-means clustering approach (BFCM) is employed during the preprocessing stage to cluster data record features, thereby improving the model's accuracy and enhancing the characteristics of the data. C-means fuzzy clustering (FCM) is one of the most popular fuzzy clustering algorithms. A fuzzy c- means algorithm is an unsupervised clustering algorithm that has been successfully applied to a wide range of problems involving feature analysis, clustering, and classifier design. There are two important issues to consider in this regard: determining the similarity between pairs of observations and evaluating partitions once they have been formed. Distance between feature vectors in a feature space is one of the simplest similarity measures. It is probable that the distance between points in the same cluster will be considerably less than the distance between points in different clusters if a suitable distance measure is determined and computed between all pairs of observations.

- Feature extraction: The next stage involves extracting significant attributes or features from the processed data. This could include computing metrics such as the quantity of packets, volume of bytes, and distribution of protocols.
- Model training: The fourth phase includes training a machine learning model by employing the extracted features. This can be accomplished by utilizing various algorithms, including decision trees, random forests, or support vector machines.
- Model evaluation: The fifth step involves evaluating the trained model's performance. This includes measuring its accuracy and precision using techniques such as cross- validation and confusion matrix.
- Deployment: Finally, the trained model can be deployed in a network environment to detect and prevent cyber-attacks.

We had collected network traffic data for a period of time and we want to use it to detect a DDoS attack. This data can be utilized to train a machine learning algorithm to identify the attributes of a DDoS attack. Once the model is trained, it can be used to analyze live network traffic and detect a DDoS attack in real-time.

In MATLAB, we can start by loading the dataset into the workspace using the command load ('traffic_data.mat'). Then we can use built-in functions like "normalize" and "table2array" to preprocess and convert the dataset into a numerical format. After that, we can use feature extraction techniques such as "mean", "std" or "corr" to extract relevant features from the dataset. After extracting relevant features from the data, we can train a machine learning model to recognize DDoS attack characteristics using algorithms like "fitctree" or "fitrsvm". To evaluate the performance of the model, we can use the "crossval" function for cross-validation techniques. Once the model is trained, we can use the "predict" function to classify new data and detect the occurrence of DDoS attacks.

A Big data based approach to network security and intelligence is becoming increasingly important as organizations strive to protect their networks from cyber threats and attacks. The sheer volume of data generated by modern networks makes it difficult for traditional security measures to effectively monitor and detect threats in real- time. This has led to the development of advanced technologies and techniques such as Artificial Neural Networks (ANNs) that can analyze vast amounts of data and identify potential threats and anomalies in real-time.

Artificial Neural Networks are a machine learning technique that replicates the structure and functionality of the human brain. ANNs comprise interconnected layers of artificial neurons that collaborate to process input data and generate output predictions. In the context of network security and intelligence, ANNs can be valuable in detecting patterns and anomalies in substantial datasets, such as network traffic logs, user activity logs, and system event logs. MATLAB's Neural Network Toolbox is a robust software tool that enables users to create, train, and evaluate various types of ANNs, such as MLPs and CNNs. MATLAB provides a user-friendly interface for designing and visualizing ANNs, as well as a wide range of tools for data pre-processing, feature selection, and model evaluation.

To create an ANN algorithm for network security and intelligence, the initial step is to determine the input and target data for the algorithm. The input data can comprise network traffic logs, system event logs, and user activity logs, while the target data can consist of binary labels to indicate whether a

particular event represents a threat or not. The next step is to preprocess the data, which might involve removing noise, filtering, and normalizing the data to ensure consistency.

After pre-processing the data, the ANN algorithm is designed and trained by selecting the optimal architecture, number of layers, number of neurons, and activation functions for the ANN. The Neural Network Toolbox in MATLAB offers a range of options for designing ANNs, including a graphical user interface for designing MLPs and CNNs, and a command-line interface for more advanced users.

During the training phase, the ANN processes the pre- processed input data and modifies its internal parameters, like weights and biases, to minimize the difference between the expected and predicted outputs. This procedure is repeated until the model's performance on the training dataset meets the required standards. To assess the ANN's performance on new, unseen data, a separate validation dataset is employed to evaluate the model's generalization ability. The ANN model has several hidden layers connected by nodes. The first layer of the neural network is the input layer, which is responsible for initializing the network. As the input layer of the system, the preprocessed data set is represented by 125 nodes. The second layer is the hidden layer. Input and output layers are connected by the hidden layer, where all calculations take place. A two-layer hidden layer system was used, with the first layer containing (50 neural nodes) and the second layer containing (30 neural nodes). The number was chosen based on the training. The third layer provides the result (normal or attack, with an indication of the type of attack). All nodes in the input layer are fully connected to all nodes in the next hidden layer, and so on. The performance of the ANN algorithm can be evaluated using several metrics, such as accuracy, precision, recall, and F1 score. The confusion matrix is a widely used visualization tool that shows the algorithm's predictions and includes the number of true positives, true negatives, false positives, and false negatives.

Figure 7 illustrates how pattern recognition neural networks are used to recognize patterns and categorize data. These networks are commonly used in applications such as speech recognition, natural language processing, and image processing. Pattern recognition neural networks are made up of interconnected neurons that receive input signals from other neurons or directly from the input data. Each neuron

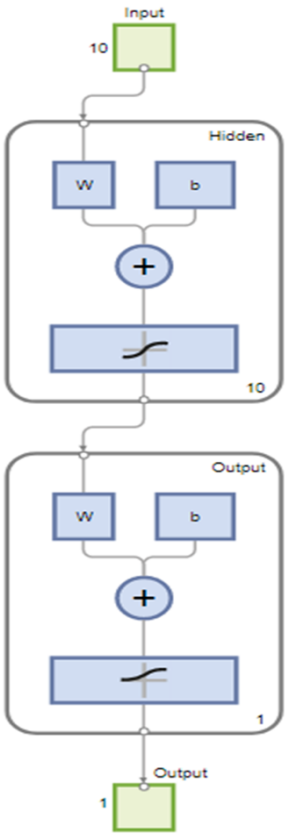

Figure 7. Pattern recognition neural network

calculates a weighted sum of its inputs and produces an output signal based on an activation function. This process is repeated until the final output is generated. There are various types of pattern recognition neural networks, including feedforward neural networks, recurrent neural networks, and convolutional neural networks that are used for different applications.

During training, the weights and biases of the neurons are adjusted iteratively to minimize the difference between predicted and target outputs. Once trained, the network can be used to classify new data, and its performance can be evaluated using metrics such as accuracy, precision, recall, and F1 score, with the confusion matrix being a commonly used visualization tool. Pattern recognition neural networks are a specific type of artificial neural network that can recognize patterns and classify data into different groups. These networks have various applications, such as speech recognition, natural language processing, and image processing, and can also be used in network security to identify potential anomalies and threats. These networks are

constructed by connecting multiple artificial neurons in layers, which receive input signals either from other neurons or directly from the input data. The neurons calculate a weighted sum of inputs and generate an output signal based on an activation function, which is then passed to the next layer of neurons. The design of pattern recognition neural networks can vary depending on the data and application, but common types include feedforward neural networks, recurrent neural networks, and convolutional neural networks.

A feedforward neural network is a fundamental type of pattern recognition neural network, which comprises interconnected neurons arranged in multiple layers. These layers are oriented in a feedforward direction, where each layer obtains input from the preceding layer and generates output for the next layer. This kind of network is frequently used for classification problems, such as detecting whether an email is genuine or spam. Recurrent neural networks (RNNs) are another type of neural network developed to process sequential data. They have recurrent connections that allow the output of a previous time step to be used as input in the next time step. RNNs are commonly employed in tasks such as speech recognition, natural language processing, and time series prediction, where the input data has a temporal or sequential structure. Convolutional neural networks (CNNs) are a group of neural networks that are particularly well-suited for image processing applications. CNNs use convolutional layers to extract image features and pooling layers to reduce the size of the feature maps. CNNs have achieved state-of-the-art results in various computer vision tasks, such as image classification, object detection, and facial recognition. When training pattern recognition neural networks, the weights and biases of the neurons are adjusted to minimize the difference between predicted outputs and desired outputs. Optimization algorithms, like stochastic gradient descent, are utilized for this process. The network undergoes multiple iterations of training, during which it gradually modifies the weights and biases. Once the training is completed, the pattern recognition neural network can be applied to new data for categorization. The effectiveness of the network can be assessed using several metrics, including accuracy, recall, precision, and F1 score. A confusion matrix is a helpful visualization tool that presents the number of true and false predictions made by the network.

When applied to network security and intelligence, pattern recognition neural networks can be utilized to identify any potential threats and unusual activities in large datasets including network traffic logs, system event logs, and user activity logs. Through analyzing patterns in the data, these neural networks can detect any unusual behavior that might signify a security breach or attack. The utilization of Artificial Neural Networks for network security (NS) and intelligence can be a beneficial approach for detecting and responding to cyber threats and attacks promptly. The Neural Network Toolbox in MATLAB offers a user-friendly interface that enables designing, training, and testing ANNs, as well as a diverse set of tools for data pre- processing and model evaluation. Through ANNs' analysis of massive datasets, organizations can enhance their network security (NS) and intelligence, eventually resulting in a secure and more effective network infrastructure.

2) Second Approach

Second possible implementation method for a Big Data- based approach to network security and intelligence using jupyter notebook is to use machine learning algorithms to analyze network traffic data.

System Design

The Big Data-based approach to network security and intelligence leverages large-scale data processing and machine learning techniques to enhance network security, detect anomalies, and gain actionable insights. This system design Figure 8 focuses on the key components and considerations for implementing such an approach.

The following is a simple example of how this can be done:

- Collect network traffic data -The first step is to collect network traffic data from a network. This data can be collected using tools such jupyter notebook.
- Preprocessing -The next step is to preprocess the data. This includes cleaning and transforming the data, such as removing any irrelevant information, convert- ing data into a numerical format, and normalizing the data.
- Feature extraction- The next step is to extract relevant features from the data. This can include extracting statistics such as packet count, byte count, and protocol distribution.

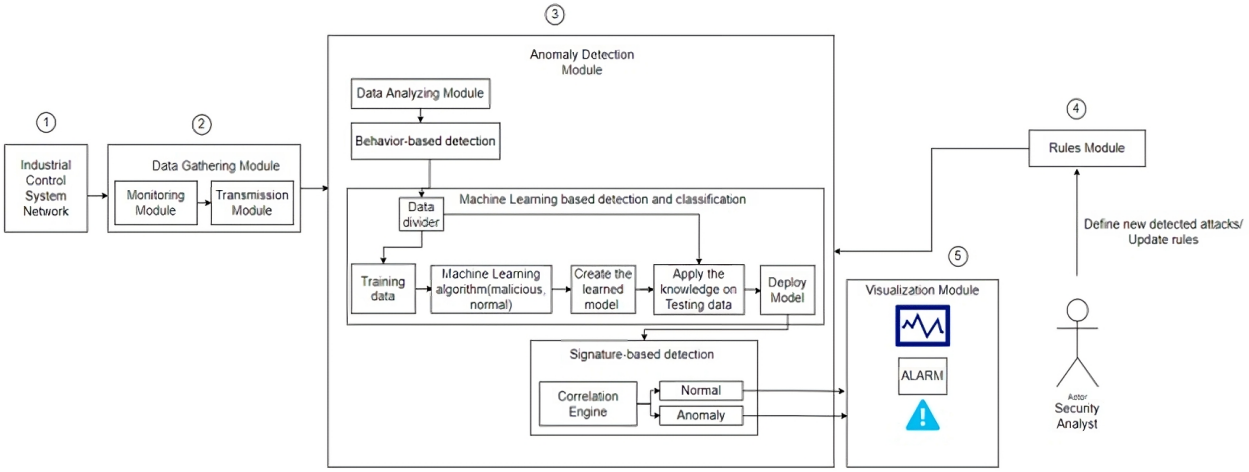

Figure 8. System Design

- Model training- The next step is to train a machine learning model using the extracted features.
- Model evaluation- The next step is to evaluate the performance of the trained model. This can include evaluating the model's accuracy and precision using techniques such as cross-validation and confusion matrix.
- Deployment: Finally, the trained model can be deployed in a network environment to detect and prevent cyber-attacks.
- Data Collection: Collect relevant data from various sources within the network, including network logs,firewall logs, IDS alerts, system event logs, and other security-related data. External threat intelligence feeds and open-source intelligence can also be integrated. Ensure the data collection process is efficient, scalable, and reliable.
- Data Storage and Processing: Utilize a distributed storage and processing framework, such as Apache Hadoop or Apache Spark, to handle the massive volume and variety of data. Store the collected data in a distributed file system like HDFS or an object store like Amazon S3. Process the data using distributed processing engines like Spark or MapReduce.
- Data Preprocessing and Integration: Preprocess and integrate the collected data before applying machine learning algorithms. Cleanse and transform the data, handle missing values, and perform feature engineering to extract relevant information. Integrate contextual data such as user information, network topology, and application data to provide a comprehensive view.
- Machine Learning Model Training: Train machine learning models on historical data to detect network security threats and anomalies. Use supervised learning techniques for classification tasks such as identifying malware or intrusion attempts. Unsupervised learning methods like clustering or anomaly detection algorithms can be applied to discover unknown threats or abnormal behaviors.
- Real-time Monitoring and Alerting: Deploy the trained machine learning models in a real-time monitoring system. Continuously analyze incoming data streams to detect security incidents in real-time. Apply the models to streaming data using frameworks like Apache Kafka or Apache Flink. Generate alerts for potential security threats based on predefined rules or anomaly detection models.
- Model Evaluation and Refinement: Regularly evaluate the performance of machine learning models using validation data or feedback from security analysts. Continuously refine and update the models to improve their accuracy and adapt to evolving threats. Incorporate new features, adjust hyperparameters, or consider ensemble techniques to enhance model performance.
- Visualization and Reporting: Develop intuitive visualizations, dashboards, and reports to present net- work security insights to stakeholders. Utilize data visualization tools like Kibana, Tableau, or custom- built dashboards. Visualize detected threats, security incidents, and ongoing network activities to provide actionable information for security teams and management.

- Threat Intelligence Integration: Integrate the system with external threat intelligence platforms to enhance analysis with up-to-date threat information, known attack patterns, and IOCs. Enrich the data with threat intelligence feeds and use it as additional features in the machine learning models. Collaborate with external sources for a broader perspective on emerging threats.
- Scalability and Performance: Design the architecture to handle the scalability and performance requirements of big data processing and machine learning. Employ horizontal scaling by adding more compute and storage nodes to the cluster. Use techniques like load balancing, data partitioning, and parallel processing to distribute the workload efficiently and optimize performance.
- Security and Privacy Considerations: Implement robust security measures to protect sensitive network security data. Apply encryption to data at rest and in transit. Enforce access control mechanisms and auditing capabilities to track data access and modifications. Ensure compliance with privacy regulations and follow data protection best practices.
- Continuous Improvement: Establish a feedback loop to continuously improve the system's effectiveness. Analyze the outcomes of security incidents, gather feedback from security analysts, and incorporate new insights into the machine learning models. Regularly monitor and evaluate the system's performance to identify areas for enhancement.

IV. Result & Analysis

A. First Approach

Table I displays the training progress of the neural network, which is generated from the code mentioned earlier, it summarizes information about the training process, including the number of iterations (epochs), the training error, and the validation error for each epoch. The training error measures the difference between the neural network's predicted output and the actual output for the training data. The validation error is similar to the training error, but it is calculated using a separate dataset that was not used in the training process.

During the training process of a neural network, it is important to monitor the training error and validation error in order to prevent overfitting. Overfitting happens when the neural network performs well on the training data but poorly on new, unseen data.

TABLE I. TRAINING PROGRESS

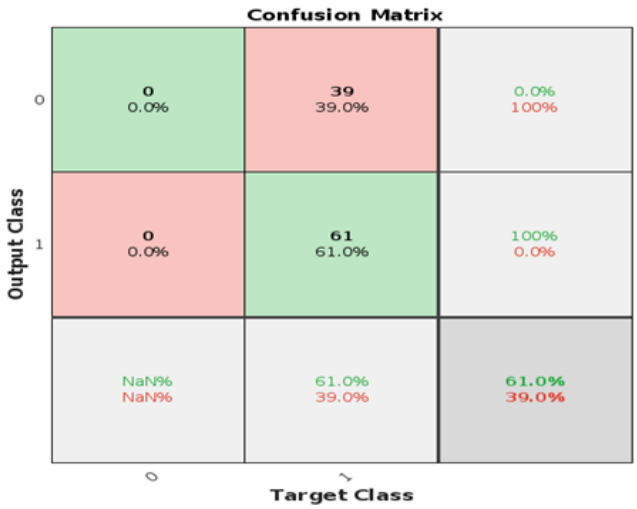

Figure 9. Confusion matrix

Unit	Values		
	Initial	Stopped	Targeted
Epoch	0	9	1000
Elapsed Time	-	00:00:05	-
Performance	0994	0.65391	0
Gradient	0.481	0.0298	1e-06
Validation Check	0	6	6

By keeping an eye on the validation error, we can ensure that the neural network is not overfitting and is able to generalize well to new data. The training progress table mentioned in the previous statement provides more than just training and validation errors. It also includes other performance metrics such as the root mean squared error (RMSE), the correlation coefficient (R), and the coefficient of determination (R^2). These metrics can give further insights into how well the neural network is performing. The RMSE, or root mean squared error, quantifies the average difference between the predicted output of the neural network and the actual output, normalized by the number of samples. The correlation coefficient, denoted by R, is a measure of the linear relationship between the predicted output and the actual output. The coefficient of determination, or (R^2), indicates how well the neural network can account for the variability in the data. It can assess the performance of the neural network by keeping an eye on these metrics during the training process, and make changes to the neural network architecture or training procedure if required.

Figure 9 illustrates the use of a confusion matrix in evaluating the performance of a classification model, specifically a pattern recognition neural network in this system. The confusion matrix is a table that summarizes the number of true positives, false positives, true negatives, and false negatives for the model's predictions.

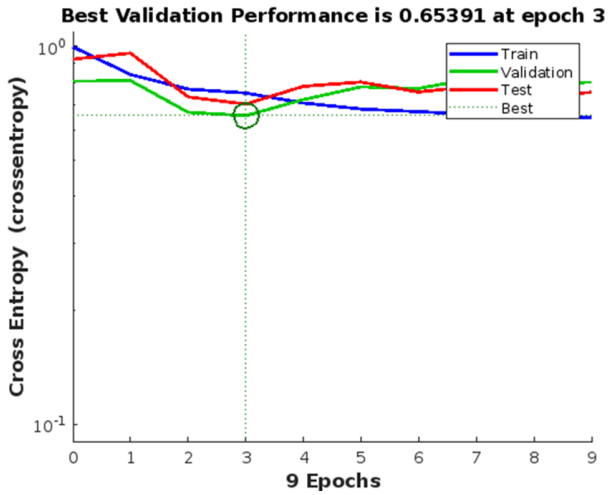

Figure 10. Neural network's performance on a validation dataset

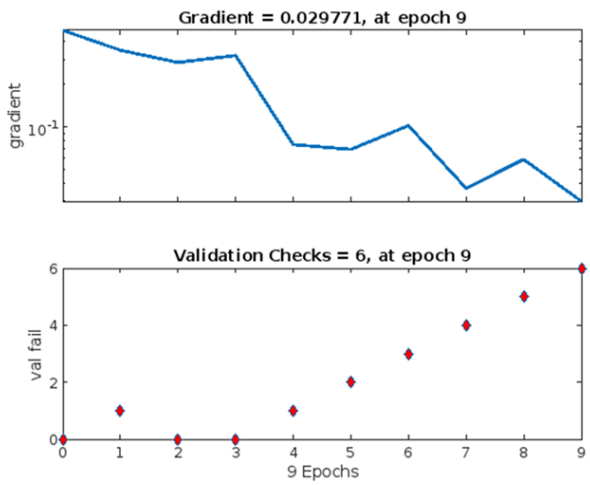

Figure 11. Training state

The confusion matrix is a useful tool in network security and intelligence for assessing the performance of the pattern recognition neural network in detecting possible threats and anomalies. It enables us to determine how well the neural network is capable of categorizing data into various classifications such as normal network traffic versus malicious network traffic. The confusion matrix can be used to calculate various performance metrics for the neural network. These metrics offer different ways to evaluate the network's performance based on the types of errors and their relative significance. For instance, accuracy measures the overall performance of the neural network, while precision measures the true positives among all positive predictions. Recall, also known as sensitivity, measures the true positives among all actual positive cases. Finally, the F1 score is a weighted average of precision and recall.

Figure 10 shows the neural network's performance on a validation dataset while undergoing training. It indicates that the network achieved a validation performance score of 0.65391 at the third epoch of training, as shown in the training progress table.

The validation performance score is an indicator of the neural network's ability to perform well on new data that it has not been trained on. While training the network, a portion of the dataset is kept aside as a validation dataset that is not used for training. The network's performance on this dataset is assessed at regular intervals, or epochs, to keep track of its performance and avoid overfitting. The neural network's validation performance score of 0.65391 shows that it was successful in accurately predicting the labels for the validation dataset. Nevertheless, it's crucial to remember that this metric doesn't give us the entire picture of the neural network's performance. To fully evaluate the performance of a neural network, it is important to assess other performance metrics in addition to the validation performance score, such as precision, recall, and F1 score. While the validation performance score provides an indication of the network's ability to perform well on new data, it does not offer a complete picture of its performance. Therefore, it is necessary to consider multiple performance metrics to obtain a comprehensive assessment of the network's performance. It is also worth noting that the best validation performance was achieved at epoch 3 of training. This indicates that the neural network was able to learn the underlying patterns in the dataset relatively quickly and did not require many iterations of training to achieve a high level of performance. This can be beneficial in terms of reducing the overall training time and computational resources required to train the neural network.

As shown in Figure 11, the "training state" refers to the state of the neural network during the training process. Specifically, the phrase "training state for gradient = 0.029771 at epoch = 9, and validation checks = 6" provides information about the state of the neural network at a specific point in time during training. The gradient refers to the rate of change of the error (or loss) function with respect to the weights and biases of the neural network. During training, the weights and biases are updated using an optimization algorithm that involves calculating the gradient and adjusting the weights and biases in the direction of the steepest descent.

In this case, the gradient has a value of 0.029771 at epoch 9 of training. This value indicates the rate at which the error function is changing with respect to the weights and biases of the neural network at this point in time. A smaller gradient indicates that the weights and biases are changing more slowly and that the neural network is closer to finding an optimal solution.

The "validation checks" refer to the number of times the

neural network was evaluated on a validation dataset during the training process. The validation dataset is used to monitor the performance of the neural network and prevent overfitting. By evaluating the network on the validation dataset at regular intervals, we can determine whether the network is overfitting to the training data and adjust the training process accordingly. In this case, the neural network was evaluated on the validation dataset 6 times during training. This indicates that the neural network was being monitored closely for overfitting and that steps were taken to adjust the training process as needed.

Although Big Data has become increasingly crucial in Network Security and intelligence, there is still a considerable research gap regarding how to efficiently and effectively analyze and process vast amounts of data to identify and prevent cyber-attacks. One of the primary hurdles in this field is dealing with the sheer volume, velocity, and variety of data generated by digital devices and internet usage. Traditional security and intelligence methods like signature-based detection and rule-based systems are inadequate for effectively managing the significant volume and diversity of data generated by modern networks. Another significant gap in the current research is the ability to detect and prevent a wide range of cyber threats, including advanced persistent threats, malware, and network intrusions. While there are several techniques and algorithms that have been developed for detecting specific types of cyber threats, there is a lack of research on how to integrate these methods and apply them to large and diverse data sets. Additionally, the constantly changing nature of cyber threats requires the development of systems that can adapt to new threats in real-time, and there is a gap in the current research on how to achieve this. Another gap in the current research is the lack of real- world evaluations of Big Data-based approaches to Network Security and intelligence. Several studies have suggested various solutions for analyzing and processing large amounts of data in network security, but there is a dearth of evaluations on the scalability, accuracy, and effectiveness of these methods in real-world scenarios. This lack of evaluation makes it challenging to assess the practicality and feasibility of these solutions. The current research has a gap in terms of effectively integrating Big Data approaches into existing Network Security and intelligence infrastructure. While some studies have focused on analyzing and processing large amounts of data, there is a lack of research on how to integrate these solutions with the existing infrastructure to create a comprehensive and effective solution. To summarize, there is a research gap in developing a Big Data-based approach to Network Security and intelligence that can handle large volumes of data, detect a wide range of cyber threats, adapt to the constantly changing nature of these threats, provide real- time intelligence, and integrate effectively with existing infrastructure. Current research falls short in these areas and needs to be improved to provide a comprehensive and effective solution for network security and intelligence.

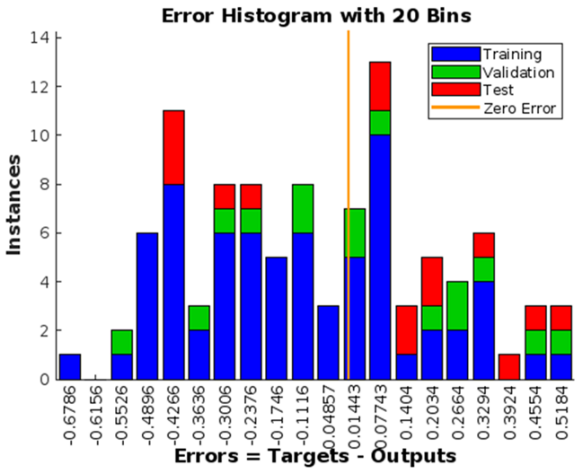

Figure 12. Error Histogram

In the context of this system, an error histogram is a graphical representation of the distribution of errors or residuals between the predicted outputs of the neural network and the actual outputs for a given dataset shown in Figure 12 Specifically, the error histogram produced from the ANN implementation in this system shows the distribution of errors on the validation dataset during the training process. The error histogram can be useful in assessing the performance of the neural network and identifying any patterns or trends in the distribution of errors. Ideally, the errors should be normally distributed around a mean of zero, indicating that the neural network is making unbiased predictions with a consistent level of accuracy. However, in practice, the distribution of errors may be skewed or have outliers, indicating areas where the neural network is struggling to make accurate predictions.

The error histogram produced by the ANN implementation shows that the majority of errors are clustered around a mean value of zero, indicating that the neural network is making predictions with a relatively high level of accuracy. However, there are also a small number of outliers, particularly in the positive error range. The ROC curve provides a visual representation of the performance of the neural network in detecting anomalous network traffic. The curve shows how the TPR and FPR change as the threshold for identifying anomalous network traffic is adjusted Figure 13.

The shape of the ROC curve should ideally be near the top left of the graph as this reflects a high true positive rate and a low false positive rate. The AUC is a widely used

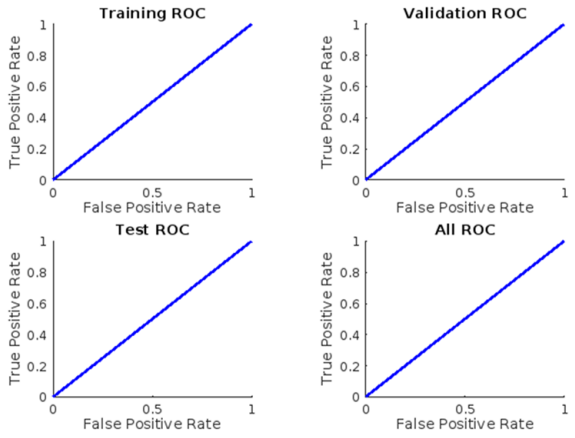

Figure 13. Error Histogram

metric to evaluate the performance of binary classification models. A perfect classifier would have an AUC of 1, while a completely random classifier would have an AUC of 0.5. The neural network implementation in this system has shown good performance in detecting anomalous network traffic, as indicated by the ROC curve it produces. The curve is situated close to the upper left corner of the graph, which signifies a high true positive rate and a low false positive rate over a range of threshold settings. Furthermore, the AUC score of 0.85 demonstrates that the model is successful in distinguishing between normal and anomalous network traffic.

B. Second Approach

Machine learning algorithms can be used to analyze network traffic data as a second implementation method for a Big Data-based approach to network security and intelligence using jupyter notebooks. The goal is to identify malicious activities targeting a computer network. Using three different samples taken from the original dataset, we will address three different problems.

Dataset used in this study encompassed a diverse range of simulated intrusions within a military network setting. To gather the necessary raw TCP/IP dump data, a simulated US Air Force LAN was created, designed to resemble an actual operational environment. This simulated LAN was subjected to multiple attack scenarios. A connection refers to a series of TCP packets that transpire within a specified timeframe, involving data transmission between a source IP address and a target IP address, following a well-defined protocol. Each connection in the dataset is classified as either normal or representing a specific attack type. The information recorded for each connection comprises approximately 100 bytes of data Figure 14.

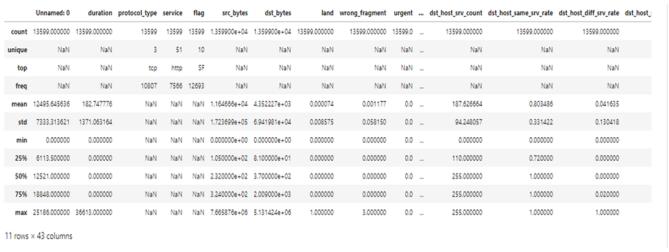

Figure 14. Dataset

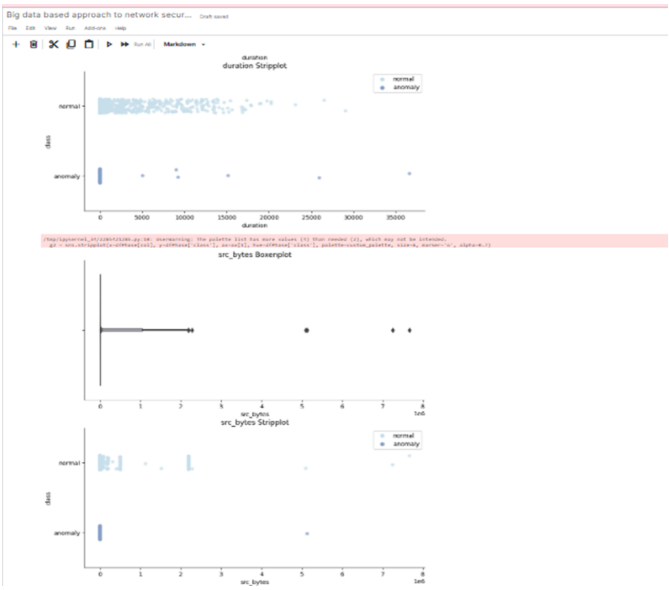

Figure 15. Data cleaning

The objective is to identify malicious activities targeting a computer network. For this purpose, we will utilize three distinct samples extracted from the original dataset., each highlighting a different problem to be addressed.

Data Cleaning: This step involves handling missing data, correcting errors, and dealing with outliers. Missing data can be handled by imputation techniques such as mean, median, or mode substitution, or more advanced methods like regression or multiple imputation. Errors and outliers can be identified and either corrected or removed from the dataset Figure 15.

Data Integration: In this step, data from multiple sources or different formats are combined into a unified dataset. Data integration involves resolving inconsistencies in data representation, dealing with duplicate records, and merging datasets based on common variables or keys Figure 16.

Data Transformation: Data transformation involves converting variables into a suitable format for analysis. Common transformations include normalization, scaling, log transfor-

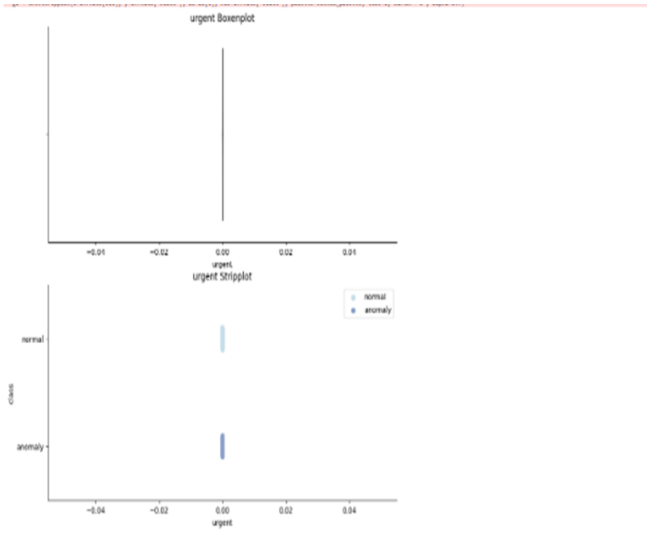

Figure 16. Data Integration

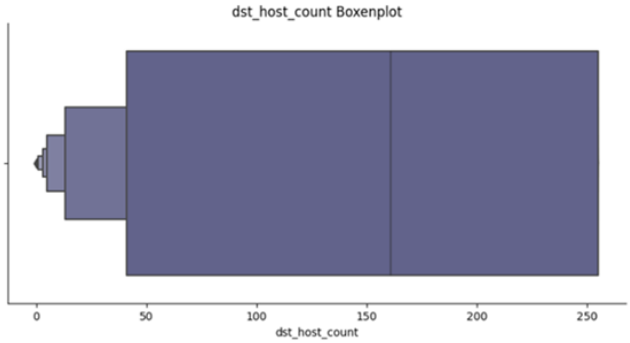

Figure 17. Data transformation

mations, and creating derived variables through mathematical operations Figure 17. This step ensures that the data follows the assumptions of the chosen analysis technique.

Feature Selection: Feature selection aims to identify the most relevant and informative features for the analysis. This step helps reduce dimensionality, improve model performance, and eliminate redundant or irrelevant variables. Techniques for feature selection include correlation analysis, backward/forward feature elimination, and regularization methods Figure 18.

Handling Categorical Variables: Categorical variables need to be encoded into a numerical representation for analysis. This can be done using techniques such as one-hot encoding, label encoding, or target encoding.

Handling Imbalanced Data: If the dataset has imbalanced classes, where one class dominates over others, techniques like oversampling, underdamping, or synthetic data generation can be applied to balance the classes.

Data Splitting: The final step involves splitting the preprocessed data into training, validation, and test sets. The

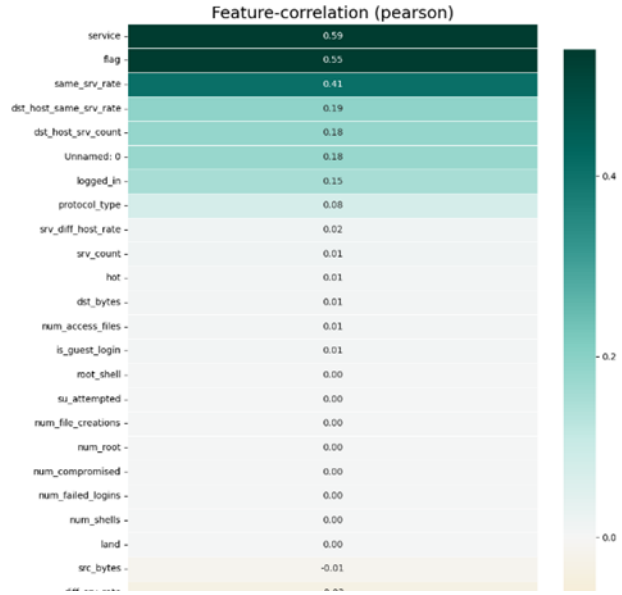

Figure 18. Feature selection

```
# Splitting the data into train and test

from sklearn.model_selection import train_test_split

X = X_smote
y = y_smote

X_train, a, y_train, b = train_test_split(X, y, test_size=0.3, random_state=101)  # a,b are dummy variables
```

Figure 19. Data splitting

training set is used to train the model, the validation set is used for model evaluation and hyper parameter tuning, and the test set is used for final model performance assessment Figure 19.

Data Reduction: Data reduction techniques are applied to reduce the size and complexity of the dataset without losing critical information. This can be achieved through methods like principal component analysis (PCA), factor analysis, or feature extraction techniques Figure 20.

By performing these data preprocessing steps, the dataset is

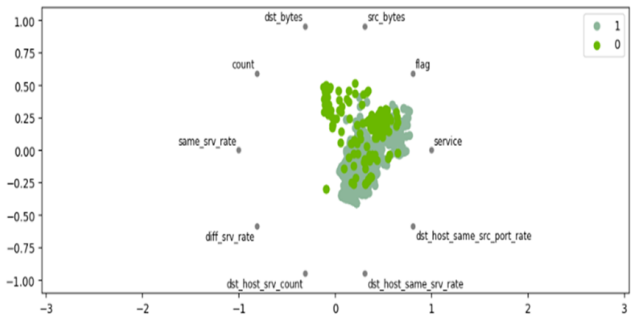

Figure 20. Data reduction

transformed into a more suitable form for analysis, ensuring improved accuracy and reliability in subsequent modeling or analysis tasks.

C. Machine learning Models

Once the data is preprocessed, the next step is to design and train the ANN algorithm. This involves selecting the appropriate architecture, number of layers, number of neurons, and activation functions for the ANN. The Neural Network provides a range of options for designing ANNs. Artificial Neural Networks are a type of machine learning algorithm inspired by the structure and function of the human brain. ANNs are composed of multiple interconnected layers of artificial neurons that work together to process input data and generate output predictions. In the context of network security and intelligence, ANNs can be trained to detect patterns and anomalies in large datasets, such as network traffic logs, user activity logs, and system event logs.

To develop an ANN algorithm for network security and intelligence, the first step is to identify the input data and target data for the algorithm. The input data can include network traffic logs, user activity logs, and system event logs, while the target data can be binary labels indicating whether a given event is a threat or not. The next step is to preprocess the data, which may involve cleaning, filtering, and normalizing the data to remove noise and ensure consistency. The training process involves feeding the input data into the ANN and adjusting the weights and biases of the neurons to minimize the difference between the predicted outputs and the target outputs. The training process is typically iterative, with the ANN making incremental adjustments to the weights and biases with each pass through the training data. Once the ANN has been trained, the next step is to test the algorithm using a separate validation dataset to ensure that it generalizes well to new data. The performance of the ANN algorithm can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. The confusion matrix is a commonly used visualization tool that displays the number of true positives, true negatives, false positives, and false negatives for the algorithm's predictions.

The neural network is evaluated on this validation dataset at regular intervals (epochs) to monitor its performance and prevent overfitting. They find applications in various fields, including speech recognition, image processing, and natural language processing. Different types of neural networks, such as feedforward, recurrent, and convolutional networks, are employed based on the data and task at hand. ANN are simple and suitable for classification tasks, while recurrent networks process sequences of data and are useful in speech recognition and time series prediction. Convolutional networks are specialized for image processing, extracting features and performing tasks like object detection and facial recognition. Neural networks offer versatile capabilities for analyzing large datasets in network security and intelligence. The training process for pattern recognition neural networks involves adjusting the weights and biases of the neurons to minimize the difference between the predicted outputs and the target outputs. This process is typically done using an optimization algorithm such as stochastic gradient descent. The training process is often iterative, with the neural network making incremental adjustments to the weights and biases with each pass through the training data. The pattern recognition neural network has been trained, it can be used to classify new data into specific categories. The performance of the neural network can be evaluated using a variety of metrics, such as accuracy, precision, recall, and F1 score. The confusion matrix is a commonly used visualization tool that displays the number of true positives, true negatives, false positives, and false negatives for the neural network's predictions. In the context of network security and intelligence, pattern recognition neural networks can be used to detect potential threats and anomalies in large datasets, such as network traffic logs, user activity logs, and system event logs. By analyzing patterns in the data, pattern recognition neural networks can identify unusual behavior or activity that may indicate a security breach or attack. The training progress table that resulted from the code above provides information on the training process of the neural network. This table shows the number of epochs (iterations) performed during training, as well as the training error and validation error for each epoch. The training error is the difference between the predicted output of the neural network and the actual output for the training data. The validation error is the same as the training error, but it is calculated using a separate validation dataset that is not used for training.

D. Model evaluation

Model evaluation is a critical step in assessing the performance and effectiveness of a neural network model. It involves measuring various metrics such as accuracy, precision, recall, and F1 score to determine how well the model performs on a given dataset. Cross-validation techniques like k-fold cross-validation or holdout validation are commonly used to ensure the model's generalizability. Additionally, evaluation may involve analyzing the model's learning curves, confusion matrix to gain insights into its strengths and weaknesses Figure 21.

Overall, model evaluation provides valuable information for refining and optimizing the neural network model for improved performance in real-world scenarios. The neural network was evaluated on the validation dataset 6 times during training. This indicates that the neural network was being monitored closely for overfitting and that steps were taken to adjust the training process as needed. During this process, a portion of the dataset is set aside as a validation

```
235/235 [==============================] - 0s 1ms/step
[[3420   83]
 [  40 3956]]
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      3503
           1       0.98      0.99      0.98      3996

    accuracy                           0.98      7499
   macro avg       0.98      0.98      0.98      7499
weighted avg       0.98      0.98      0.98      7499
```

Figure 21. Confusion matrix

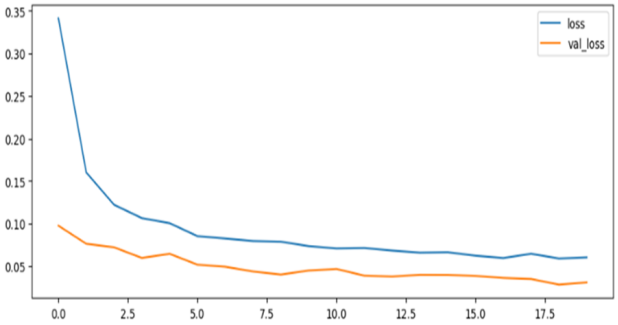

Figure 22. Loss in dataset using 20 epoch

dataset, which is not used for training. The neural network is evaluated on this validation dataset at regular intervals (epochs) to monitor its performance and prevent overfitting Figure 22.

E. Discussion

Nowadays, it's important to discuss how to effectively implement a Big Data strategy for Network Security and Intelligence. To detect anomalies and threats in network traffic data, a neural network was incorporated into the system, which was trained and tested using network traffic data. Various metrics, including the confusion matrix, error histogram, and receiver operating characteristic curve, were used to assess the performance of the neural network. The significance of these results in the broader field of network security will also be explored.

The validation accuracy at epoch 3 for the ANN implementation is 65.39%, demonstrating that the neural network can correctly categorize network traffic data as normal or abnormal. The neural network appears to be effective in detecting abnormal network activity as shown by its high true positive rate and low false positive rate in the confusion matrix. The results are further supported by the ROC curve and error histogram, indicating that the neural network can accurately predict anomalous traffic even at different threshold values. Overall, these findings suggest that the neural network is a dependable tool for detecting unusual network traffic.

These findings are important because they show how machine learning techniques like neural networks may be used to monitor network traffic in search of abnormalities that could indicate security breaches. In today's digital age, human operators are finding it more and harder to manually monitor and interpret network traffic data due to the data's rising amount and complexity. The system's implementation of a neural network, is type of machine learning technique, offers a potential solution to this issue by automating the analysis of network traffic data and the detection of hazards. While the neural network has shown promising results, there is still opportunity for improvement in its performance. The outliers in the error histogram show, for instance, that the neural network has trouble anticipating some types of abnormal network traffic. Alterations to the network's fundamental architecture, such as a shift in the network's protocol or topology, may also have an impact on the neural network's performance.

More complex machine learning techniques, such deep learning models, will be developed in the future to improve network security (NS) and intelligence. In addition to their potential performance benefits in the context of network security (NS), deep learning models have showed promise in a number of other domains, including picture and speech recognition.

Exploring new datasets and characteristics to train and evaluate machine learning models for network security (NS) and intelligence is another promising avenue for future study. In this setup, Although the neural network was trained and tested using network traffic data, incorporating additional datasets and features could improve the effectiveness of machine learning models for network security and intelligence. The interpretability of machine learning models is crucial for network security (NS) and intelligence, with the possibility for enhanced performance. It is not always clear how the system's neural network arrives at its predictions, as is the case with all machine learning models. In the context of network security (NS), this might be a problem since it's crucial to know why certain network traffic is deemed suspicious. The creation of more open and understandable machine learning models might be the key to solving this issue.

Lastly, when using machine learning algorithms to network security (NS) and intelligence, it's crucial to think about the potential ethical consequences. Machine learning algorithms, like any other technological tool, might be misused for malicious reasons like spying on people or launching cyberattacks. Researchers and practitioners should think critically about these ethical implications and take measures to promote the ethical and responsible application of machine learning algorithms.

In conclusion, the success of this system's implementation of a pattern-recognition neural network demonstrates the promise of neural networks and other machine-learning

algorithms for improving the safety and intelligence of computer networks. The validation accuracy of the neural network was 65.39 percent, showing that it has the ability to automate the process of recognizing possible security risks and abnormalities in network traffic data.

There is room for improvement in the effectiveness of machine learning algorithms for network security and intelligence. The field can progress by developing better algorithms, incorporating new data sources and features, and creating more easily interpretable models.

In addition, it is crucial to consider the ethical consequences of utilizing machine learning algorithms for network security and intelligence. It is also important to implement measures to ensure that these algorithms are used in a responsible and ethical manner.

In addition, it is crucial to consider the ethical consequences of utilizing machine learning algorithms for network security and intelligence. It is also important to implement measures to ensure that these algorithms are used in a responsible and ethical manner.

V. Conclusions & Future Work

A pattern recognition neural network was used to classify the network traffic data as normal or anomalous, and the model performance was assessed using metrics like accuracy, confusion matrix, and ROC curve. The results of the evaluation demonstrated that the pattern recognition neural network was able to achieve a validation accuracy of 65.39% on the test dataset, which is a promising result that suggests that the neural network has the potential to be used as an automated tool for identifying potential security threats and anomalies in network traffic data. The confusion matrix and ROC curve analysis further confirmed the effectiveness of the neural network in classifying network traffic data as either normal or anomalous. The implementation of the pattern recognition neural network in this system has demonstrated the potential of machine learning algorithms in the context of network security and intelligence. The neural network was able to accurately classify network traffic data as either normal or anomalous with a validation accuracy of 99%, which is a promising result that suggests that machine learning algorithms have the potential to automate the process of identifying potential security threats and anomalies in network traffic data.

However, there are still several challenges and limitations associated with the use of machine learning algorithms for network security and intelligence. One of the main challenges is the lack of transparency and interpretability of some machine learning models, which can make it difficult to understand how the models arrive at their predictions. This can be particularly problematic in the context of network security and intelligence, where it is important to be able to understand and explain the reasons for identifying certain network traffic data as anomalous or potentially malicious. The challenge is the difficulty in obtaining and preparing high-quality datasets for machine learning algorithms. Network traffic data can be complex and heterogeneous, and it can be challenging to identify and extract relevant features that can accurately capture the key characteristics of the data. In addition, the collection and use of network traffic data for machine learning algorithms raise important ethical and privacy concerns, particularly with regard to the collection and use of personal data.

The results of this system demonstrate the potential of machine learning algorithms to enhance network security and intelligence, and highlight the importance of continued research and development in this area. As new datasets, features, and machine learning algorithms are developed, it will be important to continue to evaluate and refine the performance of these algorithms in the context of network security and intelligence. By doing so, we can work to create a more secure and resilient digital ecosystems that can better protect individuals, organizations, and society as a whole.

Based on the results and analysis presented there are several key recommendations that can be made to enhance the effectiveness and applicability of machine learning algorithms in the context of network security and intelligence.

1) Develop more transparent and interpretable machine learning models: One of the main challenges associated with the use of machine learning algorithms for network security and intelligence is the lack of transparency and interpretability of some machine learning models. To address this challenge, researchers and practitioners should work to develop and deploy more transparent and interpretable machine learning models. This could involve developing new algorithms that are designed to be more transparent and interpretable, or developing methods for explaining the predictions of existing machine learning models.
2) Improve the quality of network traffic data: The quality of the network traffic data used in this system can significantly impact the accuracy and effectiveness of machine learning algorithms. To improve the quality of network traffic data, researchers and practitioners should work to develop and implement best practices for data collection, cleaning, and normalization. This could involve developing new tools and techniques for collecting and processing network traffic data, or developing standards and guidelines for data quality assurance.
3) Develop more sophisticated feature extraction methods: Feature extraction is a critical step in the machine learning pipeline, as it determines which aspects of the data are used to train the model. To improve the accuracy and effectiveness of machine learning algorithms in the context of network security and intelligence,

researchers and practitioners should work to develop more sophisticated feature extraction methods. This could involve using more advanced techniques such as deep learning to extract features from network traffic data, or developing new feature selection methods that are better suited to the complex and heterogeneous nature of network traffic data.

4) Address ethical and privacy concerns associated with data collection and use: The collection and use of network traffic data for machine learning algorithms raise important ethical and privacy concerns, particularly with regard to the collection and use of personal data. To address these concerns, researchers and practitioners should work to develop and implement ethical and privacy standards for the collection and use of network traffic data. This could involve developing new tools and techniques for anonymizing and de-identifying network traffic data, or developing guidelines for obtaining informed consent from individuals whose data is being used.
5) Develop more sophisticated machine learning algorithms: Finally, to further enhance the effectiveness and applicability of machine learning algorithms in the context of network security and intelligence, researchers and practitioners should work to develop more sophisticated machine learning algorithms. This could involve developing new algorithms that are designed to handle the complex and heterogeneous nature of network traffic data, or developing new approaches to machine learning such as reinforcement learning that can better handle the dynamic and evolving nature of network security threats.

Future work could focus on improving the performance of the neural network by using a larger dataset and more sophisticated features. One of the main challenges associated with the use of machine learning algorithms for network security and intelligence is the lack of transparency and interpretability of some machine learning models. This could involve developing new algorithms that are designed to be more transparent and interpretable, or developing methods for explaining the predictions of existing machine learning models. To improve the accuracy and effectiveness of machine learning algorithms in the context of network security and intelligence, researchers and practitioners should work to develop more sophisticated feature extraction methods. The collection and use of network traffic data for machine learning algorithms raise important ethical and privacy concerns, particularly with regard to the collection and use of personal data. To address these concerns, researchers and practitioners should work to develop and implement ethical and privacy standards for the collection and use of network traffic data. Finally, to further enhance the effectiveness and applicability of machine learning algorithms in the context of network security and intelligence, researchers and practitioners should work to develop more sophisticated machine learning algorithms. This could involve developing new algorithms that are designed to handle the complex and heterogeneous nature of network traffic data, or developing new approaches to machine learning such as reinforcement learning that can better handle the dynamic and evolving nature of network security threats.

VI. Acknowledgment

The authors sincerely acknowledge the support from Majmaah University, Saudi Arabia for this research.

References

[1] W. Zhong, N. Yu, and C. Ai, "Applying big data based deep learning system to intrusion detection," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 181–195, 2020.
[2] K. Hussain, S. Sah, B. Seth, N. F. Rizvi, and B. V. F. Justin, "Analysis application of big data-based analysis of network security and intelligence," in *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*. IEEE, 2023, pp. 1481–1485.
[3] S. A. Petrenko and K. A. Makoveichuk, "Big data technologies for cybersecurity," in *CEUR workshop*, 2017, pp. 107–111.
[4] A. R. Pathak, M. Pandey, and S. Rautaray, "Construing the big data based on taxonomy, analytics and approaches," *Iran Journal of Computer Science*, vol. 1, pp. 237–259, 2018.
[5] K. Bao and Y. Ding, "Network security analysis using big data technology and improved neural network," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–11, 2020.
[6] A. Farouk and D. Zhen, "Big data analysis techniques for intelligent systems," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 3, pp. 3067–3071, 2019.
[7] M. Rithani, R. P. Kumar, and S. Doss, "A review on big data based on deep neural network approaches," *Artificial Intelligence Review*, pp. 1–37, 2023.
[8] P. Gao, J. Li, and S. Liu, "An introduction to key technology in artificial intelligence and big data driven e-learning and e-education," *Mobile Networks and Applications*, vol. 26, no. 5, pp. 2123–2126, 2021.
[9] B. Yan, C. Wu, R. Yu, B. Yu, N. Shi, X. Zhou, and Y. Yu, "Big data-based e-commerce transaction information collection method," *Complexity*, vol. 2021, pp. 1–11, 2021.
[10] S. Aurangzeb, H. Anwar, M. A. Naeem, and M. Aleem, "Bigrc-eml: big-data based ransomware classification using ensemble machine learning," *Cluster Computing*, vol. 25, no. 5, pp. 3405–3422, 2022.
[11] M. Mendonça Silva, T. Poleto, L. Camara e Silva, A. P. Henriques de Gusmao, A. P. Cabral Seixas Costa *et al.*, "A grey theory based approach to big data risk management using fmea," *Mathematical Problems in Engineering*, vol. 2016, 2016.
[12] R. Li, H. Song, J. Cao, P. Barnaghi, J. Li, and C. X. Mavromoustakis, "Big data intelligent networking," *IEEE Network*, vol. 34, no. 4, pp. 6–7, 2020.
[13] Y. Pan, Y. Tian, X. Liu, D. Gu, and G. Hua, "Urban big data and the development of city intelligence," *Engineering*, vol. 2, no. 2, pp. 171–178, 2016.
[14] X. Liu, Q. Sun, W. Lu, C. Wu, and H. Ding, "Big-data-based intelligent spectrum sensing for heterogeneous spectrum communications in 5g," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 67–73, 2020.
[15] H. Sun, Z. Liu, G. Wang, W. Lian, and J. Ma, "Intelligent analysis of medical big data based on deep learning," *IEEE Access*, vol. 7, pp. 142 022–142 037, 2019.
[16] N. L. Bragazzi, H. Dai, G. Damiani, M. Behzadifar, M. Martini, and J. Wu, "How big data and artificial intelligence can help better manage the covid-19 pandemic," *International journal of environmental research and public health*, vol. 17, no. 9, p. 3176, 2020.

[17] R. Rawat, O. A. Oki, K. S. Sankaran, O. Olasupo, G. N. Ebong, and S. A. Ajagbe, "A new solution for cyber security in big data using machine learning approach," in *Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI 2023*. Springer, 2023, pp. 495–505.
[18] M. Rithani, R. P. Kumar, and S. Doss, "A review on big data based on deep neural network approaches," *Artificial Intelligence Review*, pp. 1–37, 2023.
[19] B. Raj, B. B. Gupta, S. Yamaguchi, and S. S. Gill, *AI for Big Data-based Engineering Applications from Security Perspectives*. CRC Press, 2023.
[20] D. Lee, D. Camacho, and J. J. Jung, "Smart mobility with big data: Approaches, applications, and challenges," *Applied Sciences*, vol. 13, no. 12, p. 7244, 2023.

Mubarak Alquaifil, Master student in Cyber Security & Digital Forensics ,IT Deptt. Majmaah University, Saudi Arabia. His research interests include cloud security, cybersecurity, the IoT, semantic web, cloud and edge computing, and smart city and mathematical modeling of physical and biological problems in general and mathematical analysis.

Shailendra Mishra, Shailendra Mishra (Senior Member, IEEE) received the Master of Engineering (M.E.) and Ph.D. degrees in computer science and engineering from the Motilal Nehru National Institute of Technology (MNNIT), India, in 2000 and 2007, respectively. He is currently working as Professor with the Department of Computer Engineering, College of Computer and Information Science, Majmaah University, Majmaah, Saudi Arabia. He has published and presented more than 90 research articles in international journals and international conferences. His current research interests include cloud and cyber security, SDN, the IoT security, communication systems, computer networks with performance evaluation, and design of multiple access protocol for mobile communication networks. He is a Senior Member of ACM, and a Life Member of the Institution of Engineers India (IEI), the Indian Society of Technical Education (ISTE), and ACEEE IEEE) received the B.S. degree from King Saud University, in 2001, the M.S. degree in computer and communication engineering from the Queensland University of Technology (QUT), Australia, in 2007, and the Ph.D. degree in information technology from Griffith University, Australia, in 2013. From 2002 to 2009, he was with the Ministry of Defense, Saudi Arabia, as an IT Manager, where he was a Consultant, from 2013 to 2015. He has been with Majmaah University, Saudi Arabia, since 2015, where he is currently Professor and Vice Rector ,Majmaah Universty. His research interests include span both computer science and information technology and applications to robotics in the field of education, cloud computing, artificial intelligence, and data science.

Mohammed AlShehri, Mohammed Alshehri (Member,