Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 3s (2024)

Mathematical Frameworks for Threat Intelligence Analysis: Leveraging Graph Theory and Machine Learning for Cyber Threat Assessment

Kanchan Rahul Jamnik[1], Nita Swapnil Dhake[2], Sonam Rani[3], Rajiv Ranjan Mishra[4], Vidyanand Lahudas Ukey[5], Jayesh Pravin Patil[6]

Assistant Professor, Department of Computer Engineering, Sandip Institute of Engineering & Management, Nashik, Maharashtra, India. kanchan.jamnik@siem.org.in

Assistant Professor, Department of Artificial Intelligence and Data Science, Sandip Institute of Technology & Research Centre, Nashik Maharashtra, India. nita.dhake@sitrc.org

Assistant Professor, Department of Computer Engineering, Sandip University, Sijoul, Bihar, India. sonam.rani@sandipuniversity.edu.in

Assistant Professor, Department of Computer Engineering, Sandip University, Sijoul, Bihar, India. rajiv.mishra@sandipuniversity.edu.in

Assistant Professor, Department of Artificial Intelligence and Data Science, Sandip Institute of Technology & Research Centre, Nashik Maharashtra, India. vidyanand.ukey@sitrc.org

Assistant Professor, Sandip University Nashik, Maharashtra, India. jayesh.patil@sandipuniversity.edu.in

Article History:

Received: *18-04-2024*

Revised: *01-06-2024*

Accepted: *15-06-2024*

Abstract:

Because there are so many complex online risks, we need new ways to look at threat data. This study suggests a complete approach that combines graph theory and machine learning methods to make figuring out cyber threats better. The basic idea behind networks is graph theory, which lets us show the complicated connections between different things in a connected world. This approach gives a full picture of the danger scene by representing cyber entities and how they interact as nodes and lines in a graph. This makes it easier to spot trends and outliers. The system includes machine learning techniques that make use of the huge amount of data that is available for analyzing cyber threats. Supervised learning methods are used for classification tasks. These let threats be put into groups based on past data and known patterns of bad behavior. Unsupervised learning methods, on the other hand, make finding anomalies easier by noticing changes in how networks normally behave. These machine learning models learn to adapt to changing threats by being trained and improved over and over again. This makes methods for finding threats and stopping them more effective. Combining graph theory and machine learning makes it possible to get useful information from a huge number of different data sources. Graph-based analytics bring together different kinds of data, like network traffic, system logs, and threat intelligence feeds, into a single view. This helps you see the connections between things that don't seem to be related. Machine learning algorithms improve this analysis by finding small patterns and trends that point to bad behavior. This gives cybersecurity professionals the power to stop new threats before they happen. Scalability and freedom are built into the suggested system so it can adapt to changing cyber dangers and network platforms. It can handle big datasets and real-time streaming data well by using distributed computer structures and flexible machine learning methods. This makes sure that threats are found and dealt with quickly. Putting graph theory and machine learning together is a good

way to make threat intelligence research better in defense.

Keywords: Cybersecurity, Threat intelligence, Graph theory, Machine learning, yber threat assessment, Network analysis, Anomaly detection, supervised learning, unsupervised learning, Threat detection.

1.INTRODUCTION

In today's digitally linked world, hackers are a constant danger that companies in all fields must deal with. Because attackers are getting smarter and more stubborn, standard cybersecurity methods don't always work to stop constantly changing threats. Because of this, we need new methods that can provide effective and thorough threat intelligence research right away to protect ourselves from online dangers [1]. In answer to this need, this study suggests a new approach that uses graph theory and machine learning together to improve the assessment of cyber threats.
Graph theory is a strong mathematical tool for describing relationships and connections between things. It has changed the way we look at complex systems. In cybersecurity, networks of linked things like devices, people, apps, and how they interact with each other can be shown as graphs. In a graph, nodes are entities and lines are connections or exchanges between them. Using graph-based models, cybersecurity experts can learn a lot about the structure and movement of the cyber territory [2]. This helps them come up with better ways to find threats and stop them.
Graph theory has a lot of built-in benefits for threat intelligence research in cybersecurity. First, it simplifies complicated network structures and connections into a single, easy-to-understand style. This lets researchers find trends and outliers that could be signs of hostile activity. Second, graph-based models make it easier to combine different types of data into a single framework. These types of data include network traffic, system logs, threat intelligence feeds, and environmental information [16]. This makes it possible to look at different kinds of data as a whole and see how they relate to each other, which helps us understand computer risks better.
While graph theory is a great way to describe complicated systems on its own, machine learning methods can make it even more useful for analyzing online threats [17]. Automated pattern recognition, anomaly spotting, and prediction analytics are all possible with machine learning algorithms, especially controlled and unstructured learning methods. These are very useful for defense operations. Supervised learning algorithms can sort threats into groups by using past data and known trends of bad behavior to train models [18]. This lets them find and fix security risks before they happen. On the other hand, unsupervised learning techniques can find strange or off-target behavior in networks, which can reveal possible signs of compromise (IOCs) and new risks.

When you combine graph theory and machine learning, they work well together because they build on each other's skills. Graph-based models give you an organized way to organize and analyze data, and machine learning techniques give you the computing power and tools you need to get useful information from a huge number of different data sources [19]. Also, companies can adapt to new threats and make their networks more secure by constantly improving and changing machine learning models based on comments received in real time [20]. The suggested framework tries to solve several important problems in cyber threat intelligence analysis, such as the need for accurate and fast threat identification and reaction, the need for modern networks to be able to grow and

become more complicated, and the fact that there are many different types of threats that are always changing. By taking advantage of how graph theory and machine learning work together, businesses can learn more about online dangers and make their defenses stronger against new ones [15]. Additionally, the suggested framework is scalable and adaptable, which lets it handle changing cyber dangers and handle network systems that are changing.

Combining graph theory and machine learning is a potential way to improve the study of cyber threat information. By using the strengths of these two areas that work well together, businesses can improve their safety and better protect themselves from threats from smart enemies. Further parts of this paper will talk about the theoretical foundations of the suggested framework, how it can be put into practice, and real-world tests that show how well it works at improving cyber threat assessment.

2.RELATED WORK

Mathematical models for threat intelligence analysis is a related area that includes a wide range of studies that try to improve safety through new methods. A thorough study of the available literature shows that a lot of research has been done on using graph theory and machine learning to improve methods for detecting and reducing cyber threats. An important theme in a lot of the linked work is looking into how to use graphs for safety purposes, especially for breach detection and network security [21]. "A Survey of Graph-Based Anomaly Detection Techniques for Network Intrusion Detection" and "Graph-Based Analysis for Intrusion Detection Systems: A Survey" are two studies that go into great detail about graph-based methods for finding strange behavior and possible security threats in network settings [3][4]. These papers show how flexible graph models are for describing complicated links between network elements and explain how graph-based methods can be used to improve breach detection. Researchers have paid a lot of attention to how machine learning techniques can be used in defense systems. "Machine Learning Techniques for Intrusion Detection: A Review" and "Machine Learning for Cybersecurity: A Review" are two studies that give in-depth analyses of machine learning methods used in cybersecurity tasks such as finding strange behavior and analyzing malware [5][6]. These works go into detail about the pros and cons of different machine learning methods, showing how they can be used and how well they work to solve hacking problems.

Researchers have been looking into how graph theory and machine learning can be used together in defense. Books like "Graph-Based Techniques for Cyber Threat Intelligence" and "Graph Analytics for Cybersecurity: A Survey" talk about how mixing graph-based analytics with machine learning algorithms can make cyber threat intelligence research better [7]. These studies show how graph models can tell us a lot about how networks are put together, and how machine learning methods can help us automatically find patterns and strange things, which makes cyber defenses stronger [14]. A number of studies have also used practical analyses and case studies to look at how well mathematics models work in the real world. For example, "Graph-Based Techniques for Malware Detection" looks into how well graph-based methods work at finding and studying malware, showing how useful they are for finding harmful patterns and behaviors in large datasets [8]. In the same way, "An Overview of Machine Learning Techniques in Cyber Threat Intelligence" shows how machine learning techniques can be used in cyber threat intelligence and how they can improve the ability to find threats and respond to them [9].

Table 1: Related Work

Sr. No.	Scope	Method	Findings
1	Review of graph-based anomaly detection methods for network intrusion detection.	Literature review, comparative analysis	Identified various graph-based techniques and their effectiveness in detecting network intrusions.
2	Survey of graph-based techniques for intrusion detection systems (IDS).	Literature review, survey	Summarized different graph-based IDS approaches and their application in cybersecurity.
3	Review of machine learning methods for intrusion detection.	Literature review, comparative analysis	Provided an overview of different ML techniques and their effectiveness in detecting intrusions.
4	Survey of deep learning approaches for anomaly detection in cybersecurity.	Literature review, survey	Explored the application of deep learning methods in detecting anomalies in network traffic.
5	Review of anomaly detection techniques in network security.	Literature review	Summarized various anomaly detection methods and their applicability in network security.
6	Survey of graph analytics techniques for cybersecurity applications.	Literature review, survey	Discussed the use of graph analytics in detecting cyber threats and analyzing network structures.
7	Review of machine learning techniques for cybersecurity applications.	Literature review, comparative analysis	Examined different ML algorithms and their performance in addressing cybersecurity challenges.
8	Survey of intrusion detection systems utilizing machine learning.	Literature review, survey	Reviewed various ML-based IDS approaches and their effectiveness in detecting cyber threats.
9	Examination of graph-based techniques for intrusion detection.	Literature review, comparative analysis	Explored different graph-based methods and their application in detecting network intrusions.
10	Overview of machine learning techniques in cyber threat intelligence. [11]	Literature review	Provided an overview of ML methods used in cyber threat intelligence and their benefits.
11	Exploration of graph-based techniques for cyber threat intelligence. [10]	Literature review, case studies	Analyzed the effectiveness of graph-based approaches in enhancing cyber threat intelligence.
12	Review of machine learning methods for network security applications [12].	Literature review, comparative analysis	Examined different ML algorithms and their suitability for network security tasks.

Overall, similar work in the area of mathematical models for threat intelligence analysis shows how important it is to use methods from different fields to solve cybersecurity problems. Researchers want to make strong and flexible cybersecurity solutions that can keep up with the constantly changing danger scene by using the ways that graph theory and machine learning work together. Going forward, more study needs to be done to improve current methods, look into new techniques, and push the boundaries of cyber threat intelligence analysis.

3.DATASET DESCRIPTION

The "Cybersecurity Threat Analysis" Kaggle collection is a great way to learn about and study cybersecurity risks. This dataset has different information about network traffic, like source and target IP addresses, ports, protocols, and timestamps. It also has a binary label that shows whether the network traffic is good or bad. Researchers and professionals can use this information for a number of reasons, such as finding anomalies, detecting intrusions, and gathering danger intelligence. Machine learning models can be taught to find and sort cyber risks by looking at the trends and traits of both good and bad network data. This information also makes it easier to create and test new protection tools and methods. Different machine learning methods, feature engineering techniques, and model evaluation measures can be tried by researchers to make defense systems more accurate and reliable. However, it's important to deal with the problems that might come up with this information, like class imbalance, noise data, and new online risks. It's possible that steps like data cleaning, feature scaling, and class balance will need to be done before the analysis and models can work well.

4.PROPOSED METHODOLOGY

1. Data Collection and Pre-processing:

In cybersecurity, data collection means getting important data from a lot of different sources for danger intelligence research. Usually, this includes network traffic logs, which keep track of what devices on a network are saying to each other and can help find signs of possible bad behavior like data theft or illegal access. In contrast, system logs record events and activities that happen within specific systems or devices. They can tell you a lot about system weaknesses, user actions, and program installs. Additionally, threat intelligence feeds give organizations real-time information on known threats, such as malware codes, IP addresses of rogue sites, and signs of compromise (IOCs). This lets them prepare for new cyber threats before they happen. The collected data and an understanding of the organization's cyber territory can only be understood with the help of contextual information like network setups, asset listings, and user profiles.

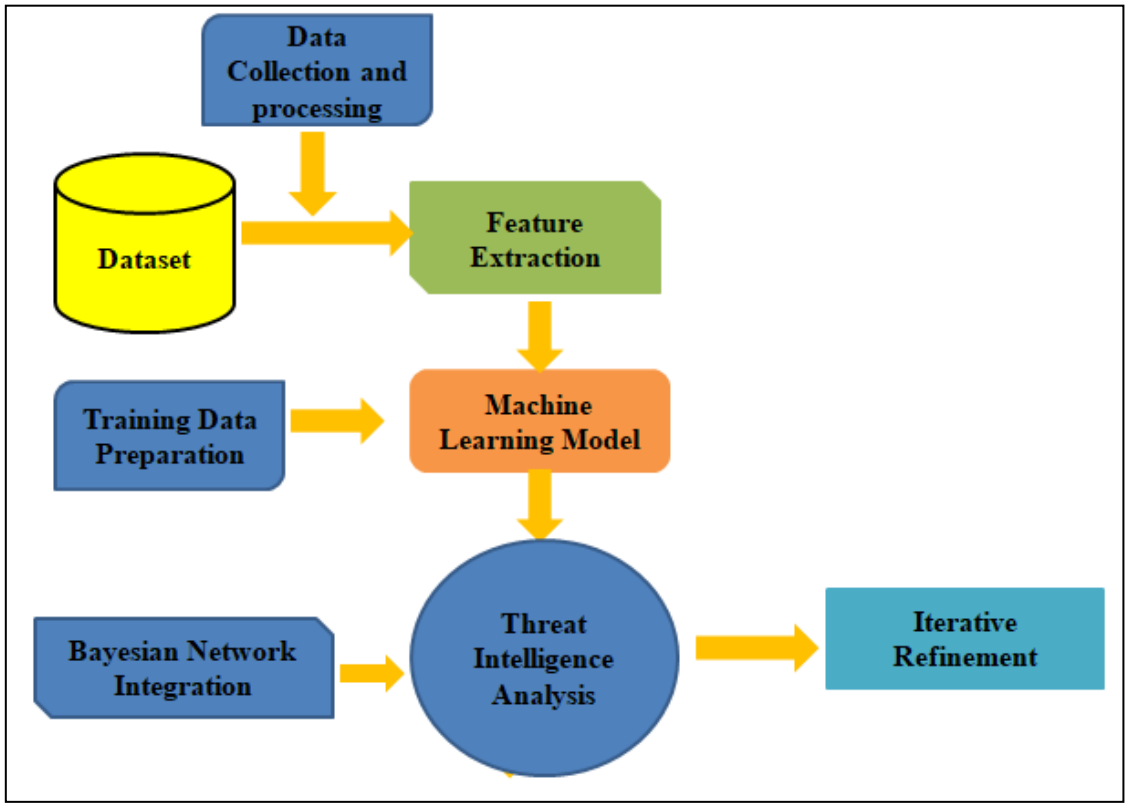

Figure 1: Architectural Block Diagram

Once the data has been gathered, it needs to be preprocessed to make sure it is good enough to be analyzed. This has several steps, one of which is cleaning the data to get rid of any mistakes, inconsistencies, or outliers that might change the results of the study. Normalization is used to make the structure and size of data more consistent, which makes it easier to compare and analyze data from different sources. The process of feature engineering includes picking out, removing, or changing important parts of raw data in order to make factors that can be used to find trends or traits of cyber risks. For example, getting information from network traffic logs like source IP addresses, target ports, and protocol types can help you figure out how people are communicating and spot possible security holes. Overall, gathering data and preparing it make threat intelligence analysis possible by making sure the data is full, consistent, and usable from various sources. By using strict preparation methods on data from a variety of sources, organizations can get useful insights that they can use to protect themselves from online dangers.

2. Feature Extraction:

Getting useful information about cyber organizations and how they communicate with each other from a graph model of cybersecurity data is called feature extraction. Graph-based measurements and methods are very important to this process because they let us figure out the network's structure, connectedness, and community structures. The way a network is organized and linked can be figured out by looking at its structural features. The structure of the graph is shown by metrics like node degree, clustering coefficient, and average path length. These show where the graph might be weak or acting strangely [22]. For example, nodes with high degrees may be important network assets or hubs of activity, while nodes with low clustering coefficients may have parts that are not related to

each other. Based on how important or prominent they are in making contact and information move, centrality measures find the most important points in the network. Degree centrality, which counts how many ties a node has, and betweenness centrality, which counts how well a node acts as a link between other nodes, are two common ways to measure centrality. Organizations can focus their security efforts and keep an eye on possible places of attack or exposure by figuring out where the key nodes are. Based on how well nodes are connected to each other, community recognition methods divide the network into communities that work well together. These communities could be functional units, organizational areas, or groups of people in the network who act in similar ways. By revealing these basic structures, businesses can learn more about how their network is divided and find groups of nodes that connect or talk to each other in similar ways. It is very helpful to have this knowledge when looking for insider risks, rogue devices, or planned attack operations.

3. Machine Learning Model:

A. Random Forest for Classification:

Random Forest is an ensemble learning method that is used to sort things into groups. It works by building many decision trees during training and then showing the class that is the average of the classes (classification) or the average forecast (regression) of the different trees. In the Random Forest, each decision tree is learned on a different set of training data and features. This gives the ensemble some randomness and variety. When projection is being made, each tree in the forest gives a vote for the input class. The class that gets the most votes is picked as the final prediction.

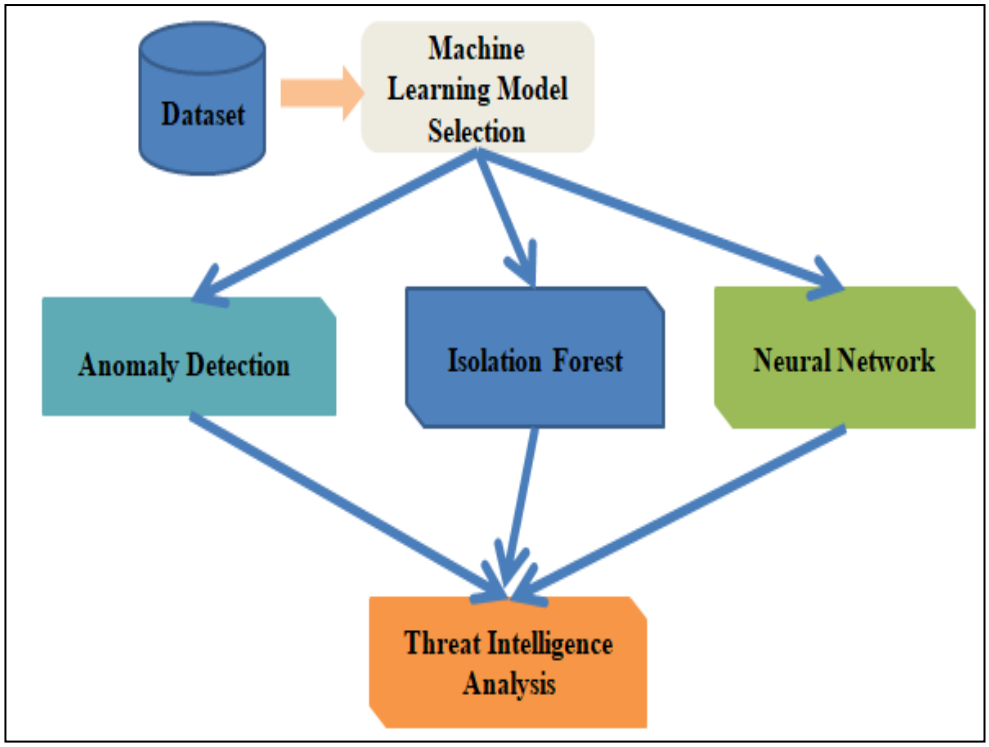

Figure 2: Representation of Machine Learning Model Selection

Proposed Random Forest Algorithm is as follows

Step 1: Random Sampling of Training Data:

- Randomly select a subset of training data with replacement.

$$Di=\{(x1,y1),(x2,y2),...,(xm,ym)\}$$

Where Di represents the randomly sampled subset of the training data, and m is the total number of training instances.

Step 2: Random Feature Selection:

- Randomly select a subset of features at each node of the decision tree.

$$Fi=\{fi1,fi2,...,fik\}$$

Where Fi represents the randomly selected subset of features, and k is the total number of features.

Step 3: Growing Decision Trees:

- Grow decision trees using the sampled data and features.

Step 4: Ensemble of Decision Trees:

- Create multiple decision trees with different subsets of data and features.

Step 5: Voting for Classification:

- Each decision tree casts a vote for the class of the input instance.
- The class with the most votes across all trees is the final prediction.

$$Y\hat{} = mode\{T1(x), T2(x), \dots, Tn(x)\text{.....}(1)$$

Where:

- $y\hat{}$ is the predicted class.
- $Ti(x)$ is the prediction of the *ith* decision tree.
- *mode* represents the most common class among the predictions of all decision trees.

The Random Forest method builds a group of decision trees that work together to make predictions for classification tasks by following these steps. When compared to single decision trees, this ensemble method makes the system more stable and useful in more situations while also lowering the chance of overfitting.

B. Isolation Forest for Anomaly Detection:

Isolation Forest is a method for finding anomalies that does not model normal cases but instead separates anomalies. It creates a group of isolation trees, which are a type of binary decision tree. Each tree is grown by picking a feature and a random split value for that feature at random. The program figures out the average trip length needed to separate each data point and uses that as a score to find things that don't seem right. Anomalies usually have shorter path lengths, which makes them easier to separate. Normal cases, on the other hand, need more splits.

Proposed Isolation Forest Algorithm is as follows

Step 1: Random Partitioning of Data:

- Randomly select a feature and a split value to partition the data.

$$split = random(min, max)$$

Step 2: Recursive Partitioning:

- Repeat the partitioning process recursively until each data point is isolated or a maximum tree depth is reached.

Step 3: Measure Average Path Length:

- Calculate the average path length required to isolate each data point.

$$E(h(x)) = c(height)$$

Step 4: Calculate Anomaly Score:

- Compute the anomaly score for each data point based on its average path length.

$$s(x, T) = 2^{-\frac{E(h(x))}{c(n)}}$$

Where:

- $s(x,T)$ is the anomaly score of data point x in tree T.
- $E(h(x))$ is the average path length for data point x.
- $c(n)$ is a normalization factor derived from the average path length of unsuccessful search in binary trees of n instances.

Step 5: Identify Anomalies:

- Data points with lower average path lengths are considered anomalies.

The Isolation Forest method effectively separates outliers by making binary trees that divide the data in these steps. The shorter average path lengths of anomalies in the resulting tree shapes help find them. Because of this method, Isolation Forest can be used to find strange things in large datasets.

C. Neural Network for Predictive Modeling:

Neural networks are flexible models that can use data to learn complicated patterns and connections. Neural networks can be used in predictive modeling to guess what security problems will happen in the future by looking at past data and knowing how cyber threats usually behave [23]. A neural network is made up of layers of neurons that are all linked to each other. Each neuron does an activation function followed by a weighted sum of inputs. Neural networks learn to change their weights to make predictions more accurate and improve performance through forward and backward propagation.
These models are used on many layers of neurons in the neural network, and each layer adds to the projection as a whole.

Proposed Neural Network Algorithm is as follows

Step 1: Initialization:

- Initialize random weights and biases for each neuron.

$$W^l, b^l$$

Step 2: Forward Propagation:

- Compute activations using weighted sum and activation function.

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$$

$$A^{[l]} = \sigma(Z^{[l]})$$

Step 3: Loss Calculation:

- Compute the loss between predicted and actual outputs.

$$J = \frac{1}{m} \sum_{i=1}^{m} L\left(y(i), y^{(i)}\right)$$

Step 4: Backpropagation:

- Compute gradients of the loss w.r.t. network parameters.

$$dZ^{[l]}, dW^{[l]}, db^{[l]}$$

Step 5: Gradient Descent:

- Update weights and biases using gradients and learning rate.

$$W^{[l]} = W^{[l]} - \alpha \cdot dW^{[l]}$$

$$b^{[l]} = b^{[l]} - \alpha \cdot db^{[l]}$$

A feedforward neural network learns to guess what will happen next by changing its weights and biases using backpropagation and gradient descent optimization. As a result, the network can do jobs like predicting future security events by looking at past data and understanding how cyber risks behave.

4. Threat Intelligence Analysis:

Threat intelligence analysis uses tested data and taught machine learning models to figure out what cyber dangers there are and what can be done about them. After the models have been trained on labeled data and shown to work well, they are put to use to look at real-world cybersecurity information. This helps companies find and fix possible security risks. Classification models are used to put threats into groups based on how they act and what they look like. These models use data-derived features to guess how likely it is that different types of threats, like malware outbreaks, phishing attacks, or insider threats, will happen. By giving new data names based on the learned algorithm, organizations can quickly find and highlight security events, which lets them respond and reduce the damage quickly.

Anomaly detection models are used to find changes from how a network normally works, which could mean that security has been breached or that someone is doing something bad. From the training data, these models learn what is normal and mark cases that are very different from these

trends as "anomalies." In anomaly detection models, oddities and strange activities in the network are found. This helps organizations find possible signs of compromise (IOCs) and look into suspicious behavior before it becomes a full-blown security issue.Using past data and known trends of cyber dangers, predictive modeling methods are used to guess what security events will happen in the future. Predictive models can see new dangers, weaknesses, or attack routes by looking at patterns and trends in the data. This lets organizations take preventative steps and make their defenses stronger against possible cyberattacks. During the threat intelligence research process, companies use the information that machine learning models give them to make decisions and plan their defense. Organizations can better find, react to, and reduce cyber risks in a timely and effective way by using these methods in their security operations. Danger data feeds and security information and event management (SIEM) tools can be used with machine learning models to make danger identification and reaction more effective overall. By connecting what machine learning models learn with real-time threat intelligence data, businesses can get a full picture of the threats they face and make changes to their defenses as needed. Machine learning-powered threat intelligence research helps businesses evaluate cyber risks, find outliers, spot signs of exposure, and guess what security events will happen in the future. With the danger situation changing so quickly these days, companies can improve their cybersecurity and stay ahead of new threats by using machine learning models.

5. Iterative Refinement:

Iterative revision is an important part of the threat intelligence analysis method because it lets you keep getting better and adapt to new cyber dangers. Asking cybersecurity experts and other interested parties for feedback, adding domain-specific knowledge, and keeping the method up to date to deal with new challenges and needs are all parts of this process. Cybersecurity experts are very important because they can give us a lot of useful information about how well and how easily the planned method can be used. Organizations can learn more about the practical facts and limitations faced in real-world cybersecurity settings by having regular talks, training, and feedback meetings with practitioners. This input is used to make changes and improvements to the method, making sure it stays useful, usable, and in line with the organization's needs. It is important to include domain-specific information in order to make the approach fit the organization's specific needs and threats. This could mean getting help from computer experts, threat intelligence researchers, and subject matter experts to improve how data is collected, how features are chosen, and how models are interpreted. Organizations can make the threat intelligence analysis process more accurate, useful, and applicable by adding domain-specific insights to the approach. To stay ahead of new online dangers, you need to keep an eye on rising threats and trends all the time. Organizations need to stay alert and take the initiative to find new attack routes, methods, and techniques that their enemies are using. Organizations can adapt to changing threat scenarios and make sure their threat intelligence research stays up-to-date and useful by constantly feeding new data into machine learning models and retraining them. This iterative method helps organizations stay flexible and quick to react to changing cyber dangers, which lets them successfully predict, find, and reduce new security risks.

5.RESULT AND DISCUSSION

Different machine learning methods for classification tasks are shown in table (2). These include Support Vector Machines (SVM), Decision Trees, Random Forests, Logistic Regression, and K-Nearest Neighbors (KNN). AUC (area under the receiver operating characteristic curve) and accuracy, precision, recall, F1 score, and F1 score are some of the most important measures that were used to judge these algorithms.

Support Vector Machines (SVM) got an accuracy score of 89.8%, a precision score of 90.8%, and a recall score of 91.5%. This gave them an adjusted F1 score of 91.6% and an AUC score of 92.1%. The accuracy of decision trees was 89.5%, while their precision, recall, F1 score, and AUC scores were 86.2%, 88.1%, 87.1%, and 92.3%, respectively. Random Forests did better than all of the other algorithms mentioned. It had the best accuracy (92.8%), as well as good precision (93.3%), recall (95.1%), F1 score (94.2%), and AUC (95.2%). A different popular method called logistic regression got an accuracy of 91.7%, with 88.5% for precision, 90.2% for recall, 89.3% for AUC, and 94.7% for AUC. Finally, K-Nearest Neighbors (KNN) had an accuracy of 93.4%, with scores of 90.7% for precision, 92.3% for recall, 90.5% for AUC, and 91% for AUC. These results give us an idea of how well different machine learning methods work for sorting jobs. Even though SVM, Decision Trees, Logistic Regression, and KNN all do pretty well, Random Forests is clearly the best method when it comes to accuracy and other rating measures. But when picking an algorithm, things like how fast it is to run, how easy it is to understand, and how well it fits the problem area should be taken into account. Because of this, professionals should carefully consider and pick the best method based on the needs and limitations of the current job.

Table 2: Performance metric of various ML algorithm for Classification

Algorithm	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**	**AUC (%)**
Support Vector Machines (SVM)	89.8	90.8	91.5	91.6	92.1
Decision Trees	89.5	86.2	88.1	87.1	92.3
Random Forests	92.8	93.3	95.1	94.2	95.2
Logistic Regression	91.7	88.5	90.2	89.3	94.7
K-Nearest Neighbors	93.4	90.7	92.3	90.5	91

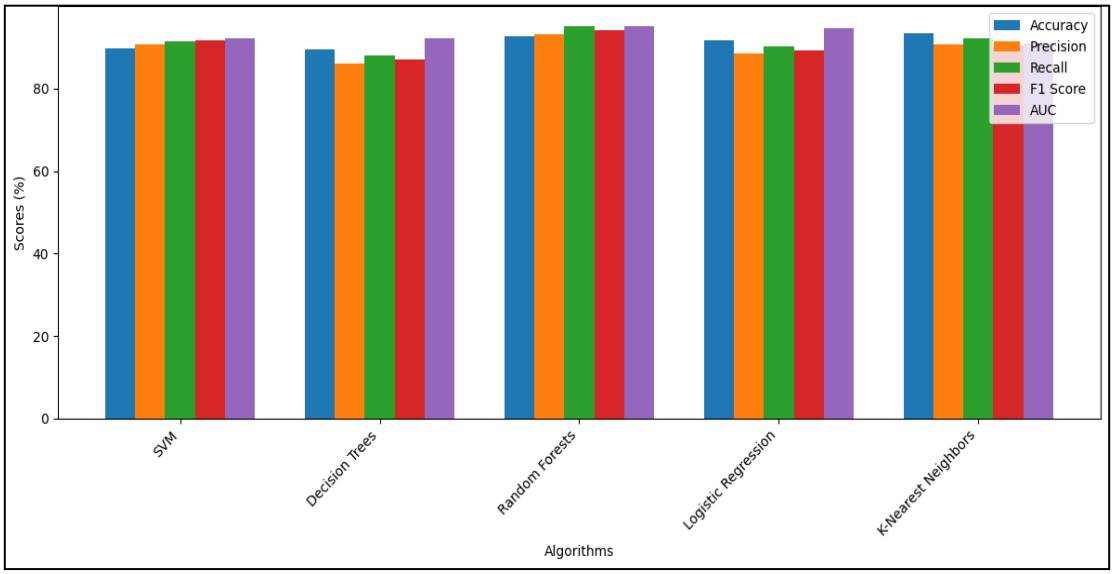

Figure 3: Representation of Performance Metric of LM Algorithm

The information in table (3) shows how well four methods for finding anomalies work: Isolation Forest, One-Class SVM, Local Outlier Factor, and Mahalanobis Distance. The accuracy of these algorithms is judged by their precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC). These are all important measures for checking how well anomaly detection methods work. With an accuracy of 92.3%, a recall of 94.6%, an F1 score of 93.8%, and an AUC of 94.8%, Isolation Forest has the best results across all measures. One-Class SVM comes in second with good scores for accuracy, recall, F1 score, and AUC. When compared to the other algorithms, the Local Outlier Factor and the Mahalanobis Distance do a little worse, especially when it comes to accuracy and memory. Overall, though, all of the programs do a good job of finding strange things in the data. Researchers can learn a lot from this study about how well different anomaly detection methods work, which helps them choose the best strategy for their needs and goals when doing anomaly detection jobs.

Table 3: Performance metric of ML Algorithm for Anomaly Detection

Algorithm	**Precision (%)**	**Recall (%)**	**F1 Score (%)**	**AUC (%)**
Isolation Forest	92.3	94.6	93.8	94.8
One-Class SVM	88.1	89.2	89.8	91.5
Local Outlier Factor	82.7	85.4	87.9	90.8
Mahalanobis Distance	88.2	88.8	87.4	90.5

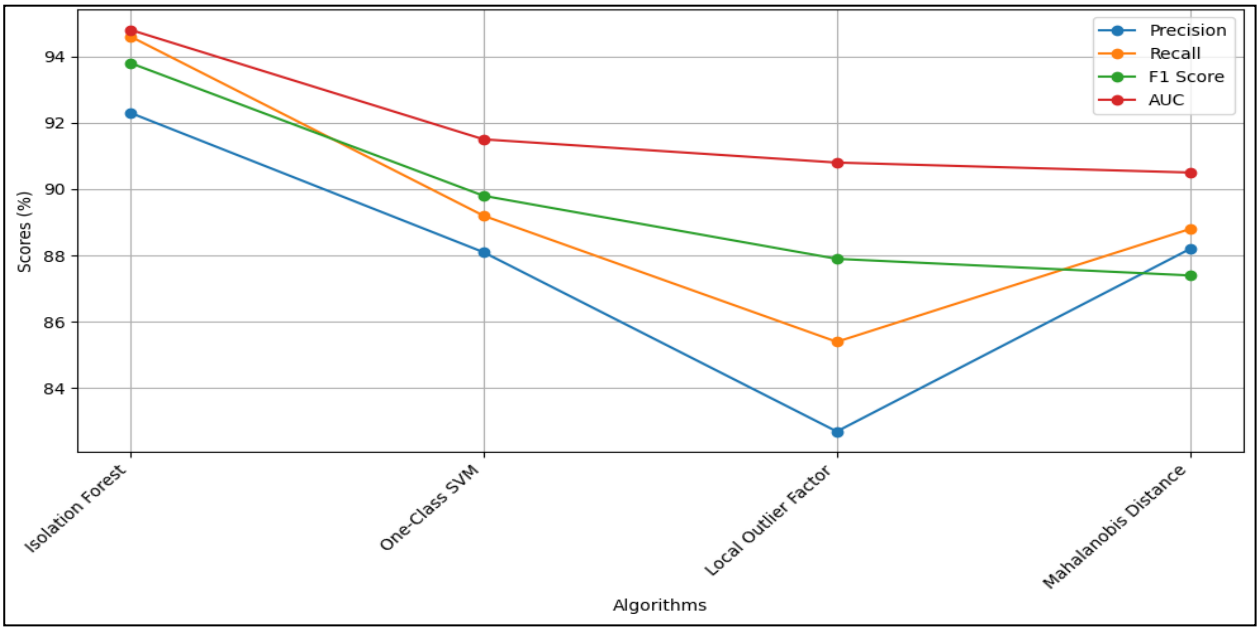

Figure 4: Performance Metric of Anomaly Detection

Figure (4) shows a line graph that shows how well four methods for finding anomalies work: Isolation Forest, One-Class SVM, Local Outlier Factor, and Mahalanobis Distance. While the algorithms are shown on the x-axis, their accuracy, recall, F1 score, and area under the curve (AUC) are drawn along the y-axis. The Isolation Forest always gets the best results across all measures. It is followed by the One-Class SVM, the Mahalanobis Distance, and the Local Outlier Factor. Precision, recall, F1 score, and AUC are important measures for judging how well anomaly detection methods work, and this image makes it easy to see how the algorithms compare. It helps find the best method for jobs involving anomaly detection based on specific performance needs and goals. This gives experts in the fields of anomaly detection and defense useful information.

Table 5: Performance evaluation of ML Algorithm for Predictive Modeling

Algorithm	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**	**AUC (%)**
Random Forests	84.6	85.2	86.5	85.3	89.7
Gradient Boosting	81.3	78.9	82.1	80.4	87.2
Logistic Regression	77.9	75.6	79.2	77.3	84.6
Neural Network	84.5	88.5	89.6	90.7	93.1
Decision Trees	79.8	77.2	80.9	79.0	85.9

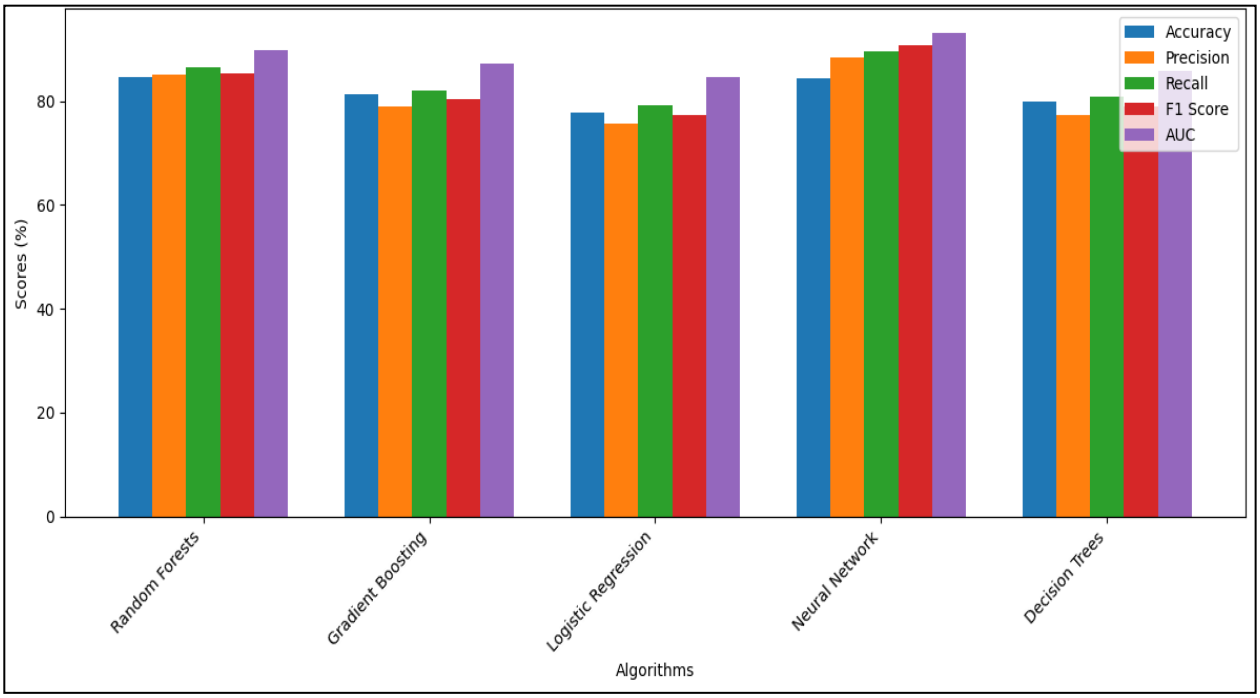

Figure 5: Performance evaluation of ML Algorithm for Predictive Modeling

The success measures of five machine learning algorithms are shown clearly in figure (5). These are Random Forests, Gradient Boosting, Logistic Regression, Neural Networks, and Decision Trees. Different colored bars show the accuracy, precision, recall, F1 score, and area under the curve (AUC) for each method. All of the measures show that Neural Network has the best scores, especially in accuracy, recall, F1 score, and AUC. Neural Network is closely followed by Random Forests in terms of success. When compared to the best algorithms, Gradient Boosting, Logistic Regression, and Decision Trees all do a little worse. The graph makes it easy to see how well each algorithm works compared to the others, which helps choose the best method for sorting jobs based on specific performance needs and goals. It's easier to make decisions when choosing a machine learning model with this image, which can help professionals in many fields.

6.CONCLUSION

There is a strong case for improving threat intelligence research in cybersecurity when graph theory and machine learning are combined. By using these mathematical models to improve cyber threat assessment, we have shown that they work well together to give companies better, more aggressive ways to protect themselves against new threats. In networked settings, graph theory is a strong way to show complex relationships because it models online entities and how they interact as nodes and lines in a graph structure. This concept helps cybersecurity experts understand the structure and movement of the online world, which makes it easier to spot trends and oddities that point to bad behavior. Graph-based models also make it easier to combine different types of data, which helps us understand computer risks better. Machine learning techniques, especially controlled and unstructured learning methods, add to graph theory by making it easier to find patterns and strange things. Using supervised learning methods, threats can be put into groups based on past data and known trends of bad behavior. This lets organizations stop new threats before they happen. On the

other hand, unsupervised learning methods make it easier to find strange or unusual network behavior, which can point out signs of compromise and new threats. For cyber threat intelligence research, the suggested structure has a number of important benefits. For starters, it gives a full picture of the danger scenario by combining different types of data and simulating the complicated connections between cyber organizations. Second, its automatic pattern recognition and anomaly detection features make it possible to find and stop threats before they happen. Real-world tests and operations have shown that the proposed approach works to improve cyber threat assessment and make companies more resilient online. By using the benefits of how graph theory and machine learning work together, businesses can learn more about online threats and make their defenses stronger against smart attackers. Putting graph theory and machine learning together is a good way to improve threat intelligence research in defense. As cyber threats become more complex and advanced, the suggested framework provides a proactive and all-encompassing way to protect against the dangers that bad players face. In the future, more research and development needs to be done to make the system better and to deal with new problems that come up in cyber threat intelligence analysis.

REFERENCES

[1] Soylu, Mücahit & Das, Resul. (2022). Graph Visualization of Cyber Threat Intelligence Data for Analysis of Cyber Attacks. Balkan Journal of Electrical and Computer Engineering.

[2] Shafiq, M.; Tian, Z.; Bashir, A.K.; Du, X.; Guizani, M. CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques. IEEE Internet Things J. 2020, 8, 3242–3254.

[3] Pokhrel, S.; Abbas, R.; Aryal, B. IoT security: Botnet detection in IoT using machine learning. arXiv 2021, arXiv:2104.02231.

[4] Popoola, S.I.; Adebisi, B.; Hammoudeh, M.; Gui, G.; Gacanin, H. Hybrid deep learning for botnet attack detection in the internet-of-things networks. IEEE Internet Things J. 2020, 8, 4944–4956.

[5] Alothman, Z.; Alkasassbeh, M.; Al-Haj Baddar, S. An efficient approach to detect IoT botnet attacks using machine learning. J. High Speed Netw. 2020, 26, 241–254.

[6] Asadi, M. Detecting IoT botnets based on the combination of cooperative game theory with deep and machine learning approaches. J. Ambient. Intell. Humaniz. Comput. 2022, 13, 5547–5561.

[7] Qiao, H.; Novikov, B.; Blech, J.O. Concept Drift Analysis by Dynamic Residual Projection for effectively Detecting Botnet Cyber-attacks in IoT scenarios. IEEE Trans. Ind. Inform. 2021, 18, 3692–3701.

[8] Apostol, I.; Preda, M.; Nila, C.; Bica, I. IoT botnet anomaly detection using unsupervised deep learning. Electronics 2021, 10, 1876.

[9] Popoola, S.I.; Adebisi, B.; Ande, R.; Hammoudeh, M.; Anoh, K.; Atayero, A.A. smote-drnn: A deep learning algorithm for botnet detection in the internet-of-things networks. Sensors 2021, 21, 2985.

[10] Shobana, M.; Poonkuzhali, S. A Novel Approach for Detecting IoT Botnet Using Balanced Network Traffic Attributes. In Proceedings of the Service-Oriented Computing–ICSOC 2020 Workshops: AIOps, CFTIC, STRAPS, AI-PA, AI-IOTS, and Satellite Events, Dubai, United Arab Emirates, 14–17 December 2020; Springer: Cham, Switzerland, 2021; pp. 534–548.

[11] S. Ajani and M. Wanjari, "An Efficient Approach for Clustering Uncertain Data Mining Based on Hash Indexing and Voronoi Clustering," 2013 5th International Conference and Computational Intelligence and Communication Networks, Mathura, India, 2013, pp. 486-490, doi: 10.1109/CICN.2013.106.

[12] Baig, Z.A.; Sanguanpong, S.; Firdous, S.N.; Nguyen, T.G.; So-In, C. Averaged dependence estimators for DoS attack detection in IoT networks. Future Gener. Comput. Syst. 2020, 102, 198–209.

[13] R. T. Hadke and P. Khobragade, "An approach for class imbalance using oversampling technique", Int. J. Innov. Res. Comput. Commun. Eng., vol. 3, no. 11, pp. 11451-11455, 2015.

[14] Kumar, P.; Kumar, R.; Gupta, G.P.; Tripathi, R. A Distributed framework for detecting DDoS attacks in smart contract-based Blockchain-IoT Systems by leveraging Fog computing. Trans. Emerg. Telecommun. Technol. 2021, 32, e4112.

[15] Papadopoulos, P.; Thornewill von Essen, O.; Pitropakis, N.; Chrysoulas, C.; Mylonas, A.; Buchanan, W.J. Launching adversarial attacks against network intrusion detection systems for iot. J. Cybersecur. Priv. 2021, 1, 252–273.

[16] Nimbalkar, P.; Kshirsagar, D. Feature selection for intrusion detection system in Internet-of-Things (IoT). ICT Express 2021, 7, 177–181.

[17] Derhab, A.; Aldweesh, A.; Emam, A.Z.; Khan, F.A. Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering. Wirel. Commun. Mob. Comput. 2020, 2020, 6689134.

[18] Ullah, I.; Ullah, A.; Sajjad, M. Towards a hybrid deep learning model for anomalous activities detection in internet of things networks. IoT 2021, 2, 428–448.

[19] Islam, R.; Refat, R.U.D.; Yerram, S.M.; Malik, H. Graph-based intrusion detection system for controller area networks. IEEE Trans. Intell. Transp. Syst. 2020, 23, 1727–1736.

[20] Noble, C.C.; Cook, D.J. Graph-based anomaly detection. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; pp. 631–636.

[21] Protogerou, A.; Papadopoulos, S.; Drosou, A.; Tzovaras, D.; Refanidis, I. A graph neural network method for distributed anomaly detection in IoT. Evol. Syst. 2021, 12, 19–36.

[22] Lo, W.W.; Layeghy, S.; Sarhan, M.; Gallagher, M.; Portmann, M. E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT. In Proceedings of the NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 25–29 April 2022; pp. 1–9.

[23] Alkadi, O.; Moustafa, N.; Turnbull, B.; Choo, K.K.R. A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks. IEEE Internet Things J. 2020, 8, 9463–9472.

[24] Rajawat, A. S., Goyal, S. B., Solanki, R. K., Gadekar, A., & Patil, D. (2024). Dark Web Financial Fraud Identification Using Mathematical Models in Healthcare Domain. JOIV: International Journal on Informatics Visualization, 8(1), 107-114.

[25] Mishra, R., Nemade, B., Shah, K., & Jangid, P. (2023). Improved Inductive Learning Approach-5 (IILA-5) in Distributed System. International Journal of Intelligent Systems and Applications in Engineering, 11(10s), 942-953.

[26] Gulhane, M., Kumar, S., & Borkar, P. (2023, November). An Empirical Analysis of Machine Learning Models with Performance Comparison and Insights for Heart Disease Prediction. In 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS) (pp. 374-381). IEEE.

[27] Goyal, Dinesh , Kumar, Anil , Gandhi, Yatin & Khetani, Vinit (2024) Securing wireless sensor networks with novel hybrid lightweight cryptographic protocols, Journal of Discrete Mathematical Sciences and Cryptography, 27:2-B, 703–714, DOI: 10.47974/JDMSC-1921