Empowering Computer Network Security using K-Means Reinforcement Learning Resource Allocation (KRL-RA)

S. Ayyasamy
Professor, School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology
Vellore, Tamil Nadu - 632014, India
email: ayyasamy.s@vit.ac.in

Aayan Rashid
UG scholar, School of Computer Science and Engineering (SCOPE),
Vellore Institute of Technology
Vellore, Tamil Nadu - 632014, India
email:
aayan.rashid2021@vitstudent.ac.in

Yash Singhal
UG scholar, School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology
Vellore, Tamil Nadu - 632014, India,
e-mail:
yash.singhal2021@vitstudent.ac.in

***Abstract*— In today's digital world, protecting network infrastructures from rapidly developing cyber threats is more important than ever. This research paper focuses on introducing Artificial Intelligence-based approaches to optimize network security. It discusses how AI can enhance intrusion detection capabilities, traffic engineering policies, and resource allocation strategies. Our vision is to build an innovative and effective network security architecture based on AI features that can predict attacks and enhance network performance through better resource management at high speeds. This research considers the inherent interdependence between traffic engineering and resource allocation in improving network performance. We improved the initial model of the k-mean clustering algorithm and combined it with other ML network security algorithms like Q-learning/Reinforcement Learning to give our final proposed KRL-RA model. We have further discussed various existing models of intrusion detection, network security, and resource allocation, and compared them to our improved version of the proposed KRL-RA model, which uses Machine Learning to improve its efficiency and reliability using past records. We have shown that with two different practical cases. Through iterative learning, the system obtains a better understanding of the network environment, leading to a greater intrusion detection success rate.**

Keywords— Intrusion detection, Data filtering, Resource allocation, k means clustering, Reinforcement learning, Traffic engineering

I. INTRODUCTION

A. Intrusion detection

IDS utilizes different methodologies that include signature-based detection where the known attack patterns are recognized by anomaly-based and targeting variations from established baselines of normal behavior. Cybersecurity researchers initiate the significance of emerging proficient network intrusion detection systems (IDS) for providing secure networks. While attackers endure to progress novel kinds of attacks and the network sizes endure to develop, the necessity for effectual IDS becomes significant. Additionally, IDS aims to offer confidentiality, integrity, and availability for the data communicated in networked computers by avoiding illegal access to the network [1]. These systems play an important role in early warning and provide timely response to threats of security incidents. Since cyber-criminality progresses towards more advanced threats, Intrusion Detection no longer goes about simply registering malicious activities. With the emergence of the modern cyber security realm, IDS not only detects threats but also brings about threat mitigation and integration of intelligence into a system.

B. Traffic Engineering

Combining artificial intelligence (AI) into network security traffic engineering may refer to an advanced method that aims at improving the complex process of data flow within a system. In essence, this combined approach of AI techniques aims at two objectives – efficiency and security communications while proactively detecting potential threats. AI-based network security traffic engineering, at its core, features advanced algorithms that can dynamically address the intricacies in terms of managing data within networks. Different from static, rule-based systems these AI algorithms are dynamic and adaptive to real-time conditions of the network; the evolving nature of this environment is influenced by potential cybersecurity risks. This multi-dimensional approach covers different aspects of network management. The first steps are aimed at improving network performance by appropriately dividing resources, optimizing data transmission rates, and directing important applications. Under such configuration, the network runs at optimum capacity to meet user and application requirements within a low congestion or performance bottleneck.

C. Allocation of Resource

Resource allocation in the context of AI-driven network security involves managing and using computational resources such as processing power, and memory storage that is bandwidth networks to achieve more effective countermeasures. As against the static resource allocation approaches, AI-integrated systems dynamically allocate resources according to real-time evaluations of network circumstances and security demands. Other, resource allocation is used by AI-based IDSs to enhance their performance. The flexible allocation of resources to IDS modules such as anomaly detection and signature-based analysis allows systems to react differently depending on levels of network activity. This adaptive resource allocation helps the IDS to maintain optimal efficiency during times of increased cyber threat. Resource allocation is another element of supporting real-time incidence response mechanisms that are integrated into the AI-based network security platform. This paper proposes the application of adaptive resource allocation algorithms and communication network security to improve the quality of educational video transmission [3]. Having an equilibrium state that is not biased to one side through AI-enabled resource allocation algorithms can strike a balance between the computational resources needed for security tasks and critical applications' runtimes.

II. RELATED WORK

A. *Artificial intelligence based IDS methods*

Artificial Intelligence aims to improve algorithms by implementing solution approaches that take after processes of human cognitive reasoning. The ability of humans to perform these tasks, training, and learning from examples is a fundamental inspiration. AI uses expert systems as they imitate human logic in a computer model to solve problems. However, it ought to be stated that high-performing expert systems require constant maintenance activities. The alternative systems can absorb reliable information from a set of training instances, which may include question-answer pairs or problems and their respective solution steps. Unlike mathematical or statistical data analysis, **Rule-Based Induction** is an approach that formulates rules that reveal training instances better [4]**. Classifier systems** try to learn how to classify future instances in one given set of training data. One example of a classifier is the Neural Network that mimics biological systems to be used for classification where networks trained on the dataset are biologically connected such that they can correctly categorize future examples. One more classifier example is the Decision Tree**. Decision trees** are created by cutting data into two or more categories repeatedly splitting each group, leading to segregated smaller sets of examples. Decision tree algorithms are developed to find out the best questions that lead most or all illustrations in a group into one class [5]. Table 1 depicts the building of a classification tree for network connections. Part of the Integral Component is Feature Selection, which seeks to minimize information needed for accurate predictions and improve classifier error rates. This entails examining part sets of features or knowledge sources and assessing their efficiency in sorting instruction cases.

B. *Traffic engineering and Resource allocation*

The goal of Network Traffic Engineering is to control data traffic communications within and across a network. However, this stream of traffic engineering uses the models of AI that allow for increasing efficiency because it gives intelligent answers about analysis optimization and security. This development in smart devices produced a mass explosion of generated data and variety. Such a trend requires new network solutions capable of carrying out traffic analysis. These solutions must be intelligent and scalable enough to automatically handle the vast amounts of data generated. Resource allocation strategies aim to effectively maximize performance, system utilization, and profit by considering virtualization technologies, heterogeneous resources, context awareness, and other features. In such complex scenarios, security and dependability are vital concerns that need to be considered in future computing and networking systems to provide future advanced services, such as mission-critical applications. This paper provides a comprehensive survey of existing literature that considers security and dependability for resource allocation in computing and networking systems [15].

C. *Data filtering and feature selection*

The purpose of data filtering is to reduce the amount of information directly handled by IDS. Some data may not be pertinent to the IDS and can thus be eliminated at an early stage before processing. Such an approach has the advantage of reducing storage needs and speeding up processing. However, care should be taken to avoid the tendency to discard important data in filtering [6].

In complex classification domains, some data could hinder the process of classifying it. This may be caused by the fact that some of the features might demonstrate false correlations, making it impossible to identify intrusions. Moreover, certain attributes could be repetitive because their details are included within other characteristics. Since adding additional features may increase calculation time and perhaps compromise the accuracy of an Intrusion Detection System (IDS), they should be limited. Feature selection is one of the major steps in data classification refinement, which determines a subset of features that will best classify training data. Within the framework of ID, features come from information sources used for intrusion detection and training instances originate in detected attempts at intrusions as well as normal behavior. Thus, feature selection has a vital role in identifying characteristic features of misuse or distinguishing among various types of abuses. In UNIX systems, a connection is defined by the source and destination port numbers using specific ports allocated to different services (e.g., telnet on 23). However, an attacker may mask network relationships by randomly assigning servers to different ports. Ports and services are mapped in a single machine, and an intruder can modify this port mapping [7]. Thus, there is a necessity to define what sort of connection was established without using port numbers. Our work aimed to investigate the effect of feature selection on improving network connection classification by reducing misclassification error rates and allowing fewer features as necessary for connection classification. In RGPSS, we make sequential selections from various sections of the search space. The objective is to refrain from directly selecting the first set of favorable elements taking into consideration that there may be yet some other options. This involves randomly forming a feature set and then using backward sequential as well as forward sequential selection on the state. While Backwards Sequential Selection and Random Generation are of polynomial complexity, the latter is more expensive than the former. The dimensions, classification error rates, and the number of states searched for every issue are given in Table 1 below. Based on the number of features that equals 7, one would obtain 128 feature subsets [8]. The results indicate that, in most cases, the use of a computational resource-intensive algorithm did not lead to considerable features or error improvements except for a problem concerning shell classification.

Feature	Feature type
Index	int
Expert system warning	float
Time in seconds	int
# of packets from source	int
# of packets from destination	int
# of data bytes from source	int
# of data bytes from destination	int

Table 1: NSM features

Problem	Algorithm	Number of States	Size of Best Feature Set	Error Rate
All	Beam	53	6	0.011266%
All	BSS	29	6	0.011266%
All	RGSS	206	6	0.011266%
SMTP	Beam	53	5	0.007231%
SMTP	BSS	29	5	0.007231%
SMTP	RGSS	190	5	0.007231%
Login	Beam	38	4	0.001177%
Login	BSS	29	4	0.001177%
Login	RGSS	188	4	0.001177%
Shell	Beam	38	4	0.002018%
Shell	BSS	29	4	0.002018%
Shell	RGSS	178	4	0.001009%

Table 2: Number of states and best feature set size along with error rates

We demonstrated the effectiveness of feature selection using small-scale cases, and now we plan to apply our approach to intrusion detection. Network Security Monitoring (NSM) provides comprehensive data that leads to what was used in our experiment and can be set up with a collection of strings to be analyzed in connections [10]. This capability proves valuable to find important files or commands that wander the network. However, managing the potential hundreds of interesting strings for inspection is difficult.

D. IDS using Decision trees

Supervised classification and regression involve the decision tree technique. It starts with something inverted, which means that each node either a choice or an attribute test branching differently represents outcomes of testing, and terminal nodes holding final decisions/target values are located decision trees form the basis for advanced ensemble methods such as Random Forest and Gradient Boosting where accuracy should be enhanced by using data from several steps Decision Trees [13].

The algorithm of decision tree is as follows:
1. Selecting the Best Feature:
The algorithm compares several features and selects the one with the highest accuracy for data segmentation. Other measures, such as entropy and information gain or Gini impurity for classification and average squared error in regression are used to identify the best-cut value.
2. Splitting the Data:
The chosen feature is applied to partition the dataset into several subsets. Each line represents different routes of the decision tree.
3. Repeating the Process:
It repeats this process for every subset, finding the best feature at each set and subdividing it.
4. Stopping Criteria:
It continues until the threshold is obtained, for instance, when some depth in a tree or no samples at each node is determined after a certain level of purity has been reached.
5. Assigning Labels or Values:
Regarding regression where a tree is constructed, the labels would be assigned for leaves based on classification or predicted values.

Performance evaluation of proposed classifier with six decision tree classifiers.

Classifiers	Detection percentage	Error percentage
Decision Stump	79.73	20.27
C4.5	92.1	7.9
Naïve Bayes	92.27	7.73
Random Forest	89.21	10.79
Random Tree	88.98	11.02
REP Tree	89.11	10.89
Proposed	98.38	1.62

Fig. 1: Performance evaluation of different models

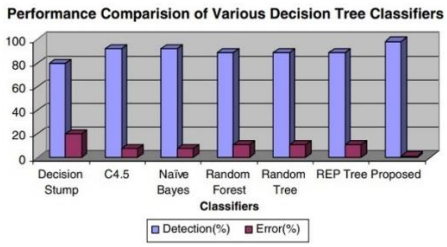

Fig. 2: Comparison in bar chart

E. Comparison of various traffic engineering models

The optimization of data flow and network performance can be achieved through various methods in network traffic engineering (NTE) [15]. A summary of some current methods and their estimated success percentages is provided as follows:

1. Quality of Service (QoS) Management [16]:
Method: The prioritization and management of various network traffic types to have better quality service for essential applications. Success Percentage: Voice/video performance (latency, jitter, and packet loss) is up to 95% better than unmanaged traffic.

2. Traffic Engineering with Multiprotocol Label Switching (MPLS):
Method: MPLS allows for efficient routing and traffic control by utilizing labels to switch as per the pre-defined paths. Success Percentage: As much as 85% reduced congestion and increased overall bandwidth use by up to 15%.

3. Dynamic Routing Protocols (including OSPF and BGP):
Method: The traffic is efficiently distributed because dynamic routing protocols change depending on what happens in a network. Success Percentage: Up to 80% optimization in routing traffic and an average of ten percent reduction in traveling time per packet.

4. Load Balancing:
Method: Load balancing network traffic across many servers to avoid overload and constant resource usage. Success Percentage: Implementation of load balancing enhances application availability and responsiveness.

5. SDN (Software-Defined Networking):
Method: SDN enables on-demand dynamic configuration of network resources and traffic flows according to real-time conditions.
Success Percentage: Up to 90% increase in application availability and up to 20% decrease in average server response time.

6. Network Slicing:
Method: Using virtual network slices having dedicated resources for services or applications.
Success Percentage: Maximum 75% decrease in network configuration time and an increase of up to 15 percent in resource efficiency.

7. Machine Learning for Anomaly Detection:
Method: Machine learning algorithms to detect the anomaly in network behaviour and potential security threats. Success Percentage: Enhancement as high as 90% in quality of service for some applications and an increase in revenue by up to 20%.

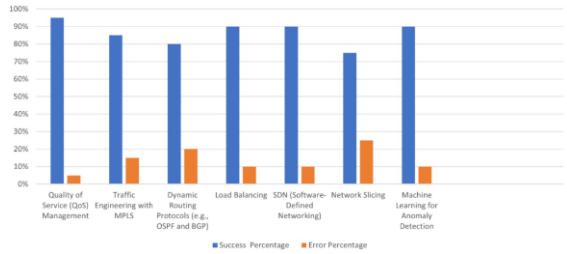

Fig. 3: Comparison of different models with their success and error percentages

III. PROPOSED WORK

A. Initial proposed model: k-means clustering

Clustering finds hidden patterns and highlights critical attributes from data for detection. Moreover, clustering can serve as a reduction operation because it preserves the cluster properties instead of raw data. There is a significant relationship between learning and data compression for researchers who are concerned with Artificial Intelligence. Generally speaking, conceptual generalization entails finding a more compact representation of the collection objects and one could use hierarchical clustering algorithms for inductive generalizations. Clustering in terms of statistics requires calculating the probability for each example to be a member of a certain cluster. Exemplar algorithms create a general representative for each cluster during the clustering process. Distance clustering has to use some distance measure for it to be able to identify cluster membership. However, conceptual clustering

requires that an object must meet both necessary and sufficient conditions to belong in a cluster.

In this particular case, we apply the K-means algorithm [11]. The objective of the K-means algorithm is an iterative procedure to cluster a dataset into 'k' predefined distinct, non-overlapping clusters. All the data points should be from only one cluster. It tries to ensure that the intra-cluster objects have maximum similarity and minimum distance between clusters. It is derived from the process of assigning data points to clusters minimizing a specific function which calculates as sums squared distances between each point and cluster centroid (the arithmetic mean of all values associated with any given cluster). The objective is to minimize intra-cluster variations and improve the uniformity of data points grouped into one cluster.

The operational steps of the k-means algorithm are as follows:
1. K defines the number of clusters desired.
2. Set up the centroid in a way that is different from randomization simply by shuffling one's dataset and randomly picking K data points as centroids without replacements.
3. Continue iteratively till convergence of the centroids, a condition that coincides with an equilibrium allocation of data points to clusters.
4. Determine the sum of squares for distances between points and centroids.
5. Find the nearest centroid for each data point.
6. Compute new centroids for the clusters by averaging all data points belonging to each cluster.

K-means clustering pseudocode:

```
Function GenerateRandom3DDataPoints(num_points):
    Set random seed for reproducibility
    Generate and return 'num_points' random 3D data points

Function KMeansClusteringWithEncircledPoints(data_points, k_values):
    Create a figure with subplots for each value of k
    For each k in k_values:
        Initialize KMeans model with 'k' clusters
        Fit the model to the data points
        Get cluster labels and centroids from the model

        Scatter plot data points with cluster labels, colored by cluster
        Scatter plot centroids in red

        For each cluster label:
            Get points belonging to the current cluster
            Calculate the cluster center and maximum distance to a point
            Create a sphere to encircle the points
            Plot the sphere with the color of the cluster

        Set labels and titles for the subplot
        Show a legend for centroids
    Display the 3D plots

# Main Script
data_points = GenerateRandom3DDataPoints(12)
k_values = [2, 3, 4]
```

B. Final proposed model: KRL-RA

We propose a model in this research paper that integrates Artificial Intelligence methods with the Intrusion Detection System (IDS) method of K-Mean Clustering and utilizes the Traffic Engineering principles of Reinforcement Learning or Q-Learning and allotting the Resources correctly and efficiently by the integration of the two models combined to make the proposed model thus increasing the success percentage of the model. The KRL-RA model or the K-Mean Reinforcement Learning Resource Allocation algorithm effectively does this process in the context of network security for identifying and mitigating anomalies. We explain this proposed model using a sample real-time scenario of network security and attempted breach/intrusion of data packets along with the response system using AI and ML.

Case Scenario: Network Security Anomaly Detection and Response:

Problem Statement: The proposed system tries to deal with the trouble of adapting to dynamic hazard environments whilst optimizing useful resource intake for protection protocols by correctly identifying and responding to anomalies in network customer conduct. Two key components of the device are K-means clustering and reinforcement mastering (Q-Learning). The K-Means Clustering factor collects information on network customers, in conjunction with their competencies, supply and excursion spot IP addresses, port numbers, packet sizes, and protocols. The desire for pertinent functions consists of payload sizes, protocol kinds, and node-to-node communication frequencies. Following that, the community website online traffic is simply divided into clusters relying on utilization patterns and the use of K-Means Clustering. After that, periodical updates are made to the cluster centroids to bear in mind the ones changing styles. The Reinforcement Learning (Q-Learning) component uses the modern-day cluster of community customers to generate actions that advocate safe reaction alternatives and ascending states. A system of praise is primarily based on replicating the efficacy of hazard discount and shielding measures. The most reliable protecting response laws are assessed using an operational and cutting-edge Q-desk, which keeps sturdy stability among research and the use of validated, effective strategies. In the reinforcement studying paradigm, each K-method cluster has to be mapped to a country. If you desire to mix the additives, moves provide plausible substitutes for shielding responses.

Using K-approach clusters, Q-values are initialized, providing a basis for the reinforcement of gaining knowledge of a set of rules. The K-means getting to know price is dynamically adjusted consistent with the fee of trade in community site visitors' styles, and periodical version updating methodically updating every K-fashion cluster as neatly as the Q-desk to dynamically tailor to evolving community site visitor's styles. Resource allocation entails the usage of Q-learning policy to make selections on protection responses on each cluster, wherein a version of Q-learning is implemented in actual time to further improve the reinforcement getting-to-know model through empirically tracking community visitors' styles and stochastically fine-tuning protection responses according to the learned principles with stochastic gradient descent algorithm.

Real-life situation example 1:

Step 1. Situation description:
The first step is to generate synthetic data for the traffic from a network with two features: Packet length and protocol type. The protocol is coded to allow easier analysis (1 means FTP protocol, 2 means port 80 HTTP protocol, and 3 means other protocols) before further testing in the sandbox.

Packet	Feature 1 (Size)	Feature 2 (Protocol)
1	450	1
2	600	2
3	300	1
4	750	3
5	400	2
6	550	1
7	200	3
8	800	2

Table 3: Description of sample feature size and protocol

Step 2. K-means Clustering:
Using a twofold K-means clustering, we classify files based on two clusters. Entries in Cluster 1 are considered as normal traffic consisting of packets 1, 3, 5, and 6; while entries in Cluster 2 are represented as possibly malicious consisting of packets 2, 4, 7, and 8 that are useful in identifying anomalies and protecting network safety.

Step 3. Q-learning for Resource Allocation:

Q- Learning Algorithm:
Step 1. Several resource allocation moves are probably noted beneath based totally on the load stage inside the network.
Step 2. Use what became surely received to update the Q-values.
Step 3. Iteratively refine the Q-values.
Step 4. Integration and Feedback Loop: - The K-method clustering is informed on an attack every time the Q -Q-learning machine detects it (for example, Packet2; Packet4, Packet7, and additionally packet8). The dynamics of Q-getting to know stem from the fact that it distributes resources based totally on the intrusion feedback information to sell both safety and also optimization in community standard overall performance.

The state matrix shows network load (Low, Medium, High), and the action matrix reflects resource-allocation levels (Low, Medium, High). The reinforcement learning algorithms can make decisions while observing the reward and Q-values. This is how the ML model basically recognizes from past activities.

Packet	Network Load	Feature 1 (Size)	Feature 2 (Protocol)
1	Low	450	1
2	Medium	600	2
3	Low	300	1
4	High	750	3
5	Medium	400	2
6	Low	550	1
7	High	200	3
8	Medium	800	2

Table 4: State matrix for example 1

Network Load	Action (Resource Allocation)
Low	Low
Medium	Medium
High	High

Table 5: Action matrix for example 1

Packet	Network Load	Action	Intrusion Detected	Reward
1	Low	Low	No	1
2	Medium	Medium	Yes	-1
3	Low	Low	No	1
4	High	High	Yes	-1
5	Medium	Medium	No	1
6	Low	Low	No	1
7	High	High	Yes	-1
8	Medium	Medium	Yes	-1

Table 6: Simplified Reward matrix

Network Load	Action (Resource Allocation)
Low	0
Medium	0
High	0

Table 7: Q-values initialized with zeroes

Real-life situation example 2:

In this case study, we will analyze the IP addresses of the devices on a network with source and destination for data packets sent over networks.

Step 1. Situation description:
The descriptions of the simulated data are below and they include:
1. Total Transferred Bytes
2. Pkts
3. IP source
4. IP destination

These descriptions include the amount of information that has been sent, the number of messages recorded, the sending and receiving of packets, and the source and destination of the information. This allows us to study in a faked environment and this will be the basis of creating a reality simulator.

Flow	Bytes Transferred	Packets	Source IP	Destination IP
1	500	10	192.168.1.1	203.0.113.1
2	1200	25	192.168.1.2	203.0.113.1
3	300	5	192.168.1.3	203.0.113.2
4	1500	30	192.168.1.1	203.0.113.2

Table 8: Description of sample packets with their source and destination IP addresses

Step 2. K-means Clustering:
Starting with feature selection, the data analysis procedure reveals that the Total Bytes Transferred (Feature 1) and the Number of Packets (Feature 2) are important in distinguishing different clusters. Further standardization also ensures data consistency. This is simplified by using K=2 K-means clustering to divide network flows into two groups. In Cluster One, flows 1 and 3 are regarded as Normal Traffic due to their distinct patterns of packet count and bytes transmitted, while Flows 2 and 4 in Cluster Two have been labeled as Potentially Malicious based on their packet and byte transport metrics. As a result, this method helps with separating normal activities from abnormal ones, thus, improving overall network security operations including detection of anomalies.

Flow	Bytes Transferred	Packets	Cluster
1	500	10	1
2	1200	25	2
3	300	5	1
4	1500	30	2

Table 9: k means clustering on example 2

Step 3. Reinforcement Learning:
In the reinforcement learning setup, state representation consists of considering Feature 1 (Total Bytes Transferred) and the current network load (Low, Medium, High). The action space is defined with Low, Medium, and High options for resource allocation to match the dynamic behavior of network demands. A system of incentives is created in which rewards are given for detecting invasion and using fewer resources. It should be noted that Q-learning techniques are used when flows fall into Cluster 2 which may indicate dangerous traffic. In such situations, the algorithm might also connect huge resources thus facilitating deeper examination and tailored reactions.

Step 4. Dynamic Feedback Loop:
Depending on the information from the K-means clustering, the Q-learning reward system is adapted for intrusion detection-learning and adjusts its resource allocation decisions dynamically based on current IDS conditions and external conditions. By Q-learning Integration, Q-learning is applied to find the optimum resource allocation strategy in comparison to the discovered clusters and the states.

Step 5. Decision based on Resource Allocation:
The state of the network can determine Q-learning's decision to assign standard resources to a flow in Cluster 1 (normal traffic). The following shows the allocation of resources:

Flow	Cluster	Network Load	Action (Resource Allocation)
1	1	Low	Standard
2	2	Medium	High
3	1	Low	Standard
4	2	Medium	High

Table 10: Action matrix for the clusters

In this instance, aid distribution is dynamically adjusted using Q-mastering in response to K-manner-recognized clusters. Cluster 1 allocates well-known assets to recurring visitors, whereas Cluster 2

allocates high resources to probably dangerous site visitors. This incorporated approach improves community safety and aids efficiency by way of allowing adaptive resource allocation based on real-time intrusion detection consequences. This is a simplified example; the function space, clustering, and Q-learning parameters would be more complicated in an actual-world scenario. Furthermore, the goals and desires of the system must be carefully taken into consideration earlier than integrating those algorithms into practice.

IV. Result analysis

This technique, which takes under consideration numerous information and trials are made for the AI to evolve and maintain getting higher on every new release, uses the analyze-from-mistake idea of Reinforcement mastering along with the Q-Learning capabilities to permit AI to boom the fulfillment percent of the proposed model on each correction it made. The result of this optimization attempt has an accuracy fulfillment percentage rating of around ninety-four percent (~94%) based totally on repeated pattern trying out, which similarly improves the fulfillment percentage of best employing k-method clustering in intrusion detection (approximately ninety-two%) and indicates the extremely good usefulness of this approach.

V. Conclusion

This research paper presents a new approach with the help of artificial intelligence techniques that are aimed at improving network security. In a general manner, Q-learning with reinforcement learning K-means clustering is used to perfect resource allocation strategies and increase penetration detection accuracy. The proposed system performs network data analysis by employing K-mean clustering. This is because clusters enable capturing patterns in the data enabling a better understanding of normal and anomalous network behavior. For example, with the aid of reinforcement learning Q-learning resource allocation becomes more adaptive. Metrics such as accuracy, precision, and recall. Future research areas towards improving the performance can be regarding parameter tuning & and optimization of the system. Its scalability is also evaluated based on the complexity and size of network infrastructures. Research also needs to be done on the practical implications of setting AI-based strategy dynamics systems and benchmarking them against still lingering network security models.

VI. References

[1] Aljehane, N. O., Mengash, H. A., Eltahir, M. M., Alotaibi, F. A., Aljameel, S. S., Yafoz, A., ... & Assiri, M. (2024). Golden jackal optimization algorithm with deep learning assisted intrusion detection system for network security. Alexandria Engineering Journal, 86, 415-424.

[2] Z. Shu et al., "Traffic engineering in software-defined networking: Measurement and management," in IEEE Access, vol. 4, pp. 3246-3256, 2016, doi: 10.1109/ACCESS.2016.2582748.

[3] Guangzhi, W. (2021). Application of adaptive resource allocation algorithm and communication network security in improving educational video transmission quality. *Alexandria Engineering Journal*, *60*(5), 4231-4241.

[4] W. Buntine, R. Caruana. “Introduction to IND and Recursive Partitioning." IND Documentation, NASA Ames Research Center, 1991.

[5] S. Weiss, C Kulikowski. Computer Systems That Learn. Morgan Kauman, California, 1991.

[6] H. Debar, M Becker, D. Siboni. “A Neural Network Component for an Intrusion Detection System." Proceedings, IEEE Symposium on Research in Computer Security and Privacy, 1992.

[7] T. Heberlein, Private Communication, March 1994.

[8] Frank, J. (1994). Artificial Intelligence and Intrusion Detection: Current and Future Directions.

[9] L. Heberlein, G. Dias, K. Levitt, B. Mukherjee, J Wood, D. Wolber. “A Network Security Monitor." Proceedings, IEEE Symposium on Research in Computer Security and Privacy, 1990.

[10] W. Buntine, R. Caruna. “Introduction to IND and Recursive Partitioning." IND Documentation, NASA Ames Research Center, 1991.

[11] L. Lankewicz, M. Benard. “Real-time Anomaly Detection Using a Non-Parametric Pattern Recognition Approach." Proceedings 7th Annual Computer Security Applications Conference, 1991.

[12] kaggle kernels pull prashant111/k-means-clustering-with-python

[13] M. Kumar, M. Hanumanthappa and T. V. S. Kumar, "Intrusion Detection System using decision tree algorithm," 2012 IEEE 14th International Conference on Communication Technology, Chengdu, China, 2012, pp. 629-634, doi: 10.1109/ICCT.2012.6511281.

[14] M. M. I. Khan and G. Nencioni, "Resource Allocation in Networking and Computing Systems: A Security and Dependability Perspective," in IEEE Access, vol. 11, pp. 89433-89454, 2023, doi: 10.1109/ACCESS.2023.3306534.

[15] R. Singh, N. Srivastava and A. Kumar, "Machine Learning Techniques for Anomaly Detection in Network Traffic," 2021 Sixth International Conference on Image Information Processing (ICIIP), Shimla, India, 2021, pp. 261-266, doi: 10.1109/ICIIP53038.2021.9702647.

[16] Gain More for Less: The Surprising Benefits of QoS Management in Constrained NDN Networks ACM ICN 2019, Macau.