

RESEARCH

Open Access



# Extremely boosted neural network for more accurate multi-stage Cyber attack prediction in cloud computing environment

Surjeet Dalal<sup>1</sup>, Poongodi Manoharan<sup>2\*</sup>, Lilhore Umesh Kumar<sup>3,4</sup>, Bijeta Seth<sup>5</sup>, Deema Mohammed alsekait<sup>6</sup>, Sarita Simaiya<sup>3,4</sup>, Mounir Hamdi<sup>2</sup> and Kaamran Raahemifar<sup>7,8,9</sup>

## Abstract

There is an increase in cyberattacks directed at the network behind firewalls. An all-inclusive approach is proposed in this assessment to deal with the problem of identifying new, complicated threats and the appropriate counter-measures. In particular, zero-day attacks and multi-step assaults, which are made up of a number of different phases, some malicious and others benign, illustrate this problem well. In this paper, we propose a highly Boosted Neural Network to detect the multi-stage attack scenario. This paper demonstrated the results of executing various machine learning algorithms and proposed an enormously boosted neural network. The accuracy level achieved in the prediction of multi-stage cyber attacks is 94.09% (Quest Model), 97.29% (Bayesian Network), and 99.09% (Neural Network). The evaluation results of the Multi-Step Cyber-Attack Dataset (MSCAD) show that the proposed Extremely Boosted Neural Network can predict the multi-stage cyber attack with 99.72% accuracy. Such accurate prediction plays a vital role in managing cyber attacks in real-time communication.

**Keywords** Zero-day attack, Multi-stage cyber attack, Neural network, Quest, Bayesian network, Intrusion detection, Security Investigation

## \*Correspondence:

Poongodi Manoharan  
dr.m.poongodi@gmail.com

<sup>1</sup> Department of Computer Science and Engineering, Amity University Haryana, Gurugram, India

<sup>2</sup> Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar

<sup>3</sup> School of Computing, University of Louisiana, Lafayette, Louisiana, USA

<sup>4</sup> Department of Computer Science and Engineering, Chandigarh University, Mohali, Punjab, India

<sup>5</sup> Department of Computer Science and Engineering, B. M. Institute of Engineering & Technology, Sonapat, Haryana, India

<sup>6</sup> Department of Computer Science and Information Technology, Applied College, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia

<sup>7</sup> College of Information Sciences and Technology, Data Science and Artificial Intelligence Program, State College, 16801 Penn State University, PA, USA

<sup>8</sup> School of Optometry and Vision Science, Faculty of Science, University of Waterloo, 200 University, N2L3G1 Waterloo, ON, Canada

<sup>9</sup> Faculty of Engineering, University of Waterloo, 200 University Ave W, Waterloo, Canada

## Introduction

Data processing has risen in various fields, including geography, engineering, business, finance, and health-care. Using cloud computing for data processing has become widely accepted. High-performance computing services are delivered through the Internet, and substantial scientific applications are run using this technique. You may use cloud computing to get three services: Infrastructure as a service, Platform as a service, and software as a service (SaaS). In the form of services, the Infrastructure as a Service (IaaS) cloud offers cloud customers access to vast computer hardware infrastructure platforms and software resources. On the other hand, users can only run an application on the Internet in a SaaS cloud; however, in a Platform as a service (PaaS) cloud, customers may utilize the existing Platform to build their application [1].

Private, public, communal, and hybrid cloud computing models exist. For businesses with similar needs, the Community cloud model is essential. When a faulty management system is used, the performance of the submitted applications/workflows is reduced. In cloud computing settings, workflow is a popular way to represent high-volume data-processing systems [2]. Graph nodes represent the computing jobs, while graph edges reflect the relationships among the graph's activities. The DAG is used to depict a workflow. The application's scientific requirements determine the DAG's size. The workflow size is modest if the scientific application is simple and straightforward. Otherwise, the amount of effort involved is enormous [3].

Cloud computing relies heavily on virtual machines (VMs), essential components. Virtual machines (VMs) let cloud service providers make the most of their physical resources. Clients may save money on computer resources in the cloud by using virtual machines (VMs) [4]. Virtual machines (VMs) are vulnerable to various security risks as traditional web servers. Brute-force SSH assaults are reasonably straightforward to stop because of the evidence left behind by failed attempts, which may be discovered in the authorization logs [5]. "Others leave the minimal record in the system logs and are thus more challenging to identify. The co-resident assault, a covert security concern, is the topic of this research (also known as co-residence, co-residency, or co-location attack). VMs on the same server can be logically isolated using virtualization methods (i.e., co-resident VMs). As a result, apps running on different VMs shouldn't conflict with one another. Although this is possible in natural cloud systems, it's unlikely. For example, the time it takes to perform a cache read operation depends heavily on the amount of data stored in the cache. Aside from the fact that malicious individuals can construct side channels between their VM and the target VM on the same server, they can collect sensitive information from the victim.

A co-resident assault is what we term this. There are several flaws in the classic defense systems [6]. First, is the attacker, who has some time on their hands? One party can devote an extensive effort to modeling the cloud system (the protections in place) and then meticulously organize their strikes. Implementing these safeguards in practice, on the other hand, is sometimes far from optimal, giving attackers more possibilities to exploit the system." More than 9 out of 10 vulnerabilities exploited will have been known to security and IT professionals for at least a year before 2020, according to an article [7]. This is partly due to the time and difficulty of regularly fixing vulnerabilities. Modifications may put customers off to the current system configuration because they

fear a decrease in Quality of Service (QoS). Secondly and thirdly, zero-day attacks emerged due to the data the attacker amasses during the assault cycle. Internet users are becoming increasingly concerned about their online safety. Intrusion Detection Systems (IDSs) face new dangers, such as Multi-Stage Attacks (MSAs), due to these new threats. An innovative and more intelligent detection strategy is needed, as is the use of new sources of information to help overcome these new difficulties [8].

Traditional one-off network attacks differ from MSAs in that they are launched in phases and over time to preserve long-term access to the target system. The steps that make up each stage of an MSA may not all be malicious, but they all play a crucial role in its effective execution. The attacker could only complete the MSAs if they were run consecutively. In addition, because of the extended time intervals between attack phases, most existing IDS have difficulty detecting MSAs [9]. There are two types of IDS now in use: those that identify abuse and those that detect anomalies. Based on known assaults, the former has a high success rate. It cannot detect new variations of established assaults, which is a bummer. However, the latter can avoid this constraint by recognizing the differences between present and usual behavior. An anomaly detection system that uses machine learning to cope with massive data and attack detection is becoming increasingly popular in intrusion detection. However, multi-stage attacks are brutal because of the two issues listed below.

- 1) Retraining dataset windows must be manually established in all existing model re-update efforts, which means that their duration is fixed. But the length of each step in multi-stage assaults varies. Assault duration and a threshold for retraining window mismatch will significantly impact detection performance. As a result, one of the most challenging tasks is figuring out how to distinguish between the various phases of an attack.
- 2) The scanning stage, prospective stage, data theft stage, and data transfer stage are all examples of multi-stage assaults.

However, in the present research, these stages are recognized individually, which makes it impossible for any intrusion prevention system (IPS) with convenient methods to identify and respond to the various stages of an assault. As a result, how to connect the various stages is an additional issue [10]. Meanwhile, existing machine learning-based anomaly detection research has a false alarm and false negative rate of more than 10%, making it unable to deal with multi-stage assaults. We propose a

Neural Network-based approach to detecting multi-stage assaults to overcome the mentioned issues. The following is a list of our most important contributions.

- 1) There are two levels of time series and stage features built into a long-term memory network.
- 2) The stage features layer is introduced to store and calculate historical data to detect the distinct stages with varied durations in multi-stage assaults. This is followed by an analysis of the time-series characteristics layer to determine if the current data falls within an attack timeframe.
- 3) Multi-stage cyber attack dataset is used in the comparison tests. Using a variety of datasets, our method has an accuracy rate of at least 91% and a false negative rate of no more than 6.75%. The false positive and false negative rates are lowered by at least 65.83% and 65.26%, respectively, compared to the current systems.

The following outlines the paper: The introduction is followed by related work. The explanation of the current state of the art and the research methods precede next. Then, it provides an overview of the model. Next, we'll look at several simulation examples. Results and discussions are provided at the end of the paper, followed by conclusions and recommendations for further research.

### Related work

This section mainly covers the review of existing cloud security research work. Research [5] expressed that work toward robotized location and recognizable proof of multi-step digital assault situations would benefit fundamentally from a technique and language for displaying such situations. The idea of assault designs was acquainted with work with the reuse of nonexclusive modules in the assault demonstrating process. CAML was utilized in a model execution of a situation acknowledgment motor that consumed first-level security cautions progressively and produced reports that distinguish multi-step assault situations found in the alarm stream.

Research [6] depicted progressed capacities for mission-driven digital situational mindfulness, given safeguard top to bottom by the Cauldron device. Cauldron consequently planned all ways of weakness through networks by connecting, totaling, normalizing, and intertwining information from various sources. It gave a refined perception of assault ways and consequently produced alleviation proposals. Adaptable demonstrating upheld multi-step examination of firewall rules and host-to-have weakness, with assault vectors inside the organization and from an external perspective. They depicted a

ready relationship given Cauldron assault charts, examining mission influence from assaults. Research [7] utilized the Hidden Markov Model (HMM) to break down and foresee the assailant's conduct, given what was gained from noticed cautions and interruptions. They utilized information mining to handle alarms to create input for the HMM to determine the expected appropriation likelihood. Their framework had the option to stream continuous Snort cautions and foresee interruptions in view of our learned standards. This framework had the option to find designs in the multi-stage assault naturally and order aggressors in view of their way of behaving. By doing this, our framework can successfully anticipate conduct and assailants and survey the risk level of various gatherings of aggressors.

Research [8] broadened a current multi-step signature language to help assault locations on standardized logs gathered from different applications and gadgets. Furthermore, the lengthy language upheld the joining of outer danger knowledge and permitted us to reference current danger pointers. With this methodology, they could make nonexclusive marks that keep them awake to date. Utilizing this language, they could distinguish different login animal power endeavors on numerous applications with only one nonexclusive mark. Research [9] portrayed a way to deal with limiting network safety gambles called Cyber Security Game (CSG), where CSG could be seen as a model-based framework for security design. CSG was a strategy supporting programming that quantitatively distinguished mission results and cantered network protection chances. It utilized a hypothetical game arrangement utilizing a game detailing that distinguished safeguard methodologies to limit the greatest digital gamble (Mini-Max), utilizing the protection strategies characterized in the protector model. This paper portrayed the methodology and the models that CSG utilizes.

Research [10] zeroed in on utilizing IDS alarms relating to unusual traffic to connect assaults identified by the IDS, recreated multi-step assault situations, and found assault chains. Because of numerous misleading up-sides in the data given by IDS, precise reproduction of the assault situation and extraction of the most

**Table 1** Summary of existing work

No.	Reference	Technique	Accuracy
1	[1]	Contextual information	58%
2	[2]	Cyber Security Game (CSG)	70%
3	[3]	multi-step attack alert correlation system	90%
4	[4]	Systematic & coherent approach	97%

fundamental assault chain was tested. Hence, they proposed a technique to reproduce multi-step assault situations in the organization, given numerous data combinations of assault time, risk evaluation, and assault hub data. The trial results demonstrated how the proposed strategy could recreate multi-step assault situations and follow them back to the first host. It could assist presiding officers with conveying safety efforts all the more successfully to guarantee the general security of the organization. Research [11] introduced Kitsune: a fitting and play NIDS that figured out how to distinguish assaults on the XXXe-enactmentXXX organization without management and in a proficient web-based way. Kitsune’scenter calculation (KitNET) utilized a gathering

of brain networks called autoencoders to separate aggregately among typical and unusual traffic designs. KitNET was upheld by a component extraction structure that productively tracks the examples of every organization channel. Their assessments showed that Kitsune recognized other assaults with an exhibition tantamount to disconnected oddity identifiers, even on a Raspberry PI. This showed the way that Kitsune can be a pragmatic and monetary NIDS.

Research [12] proposed a comprehensive framework to test complicated, innovative risks and major countermeasures. In particular, zero-day attacks, which were not publicly disclosed, and multi-step attacks, which were constructed from a few discrete breakthroughs,

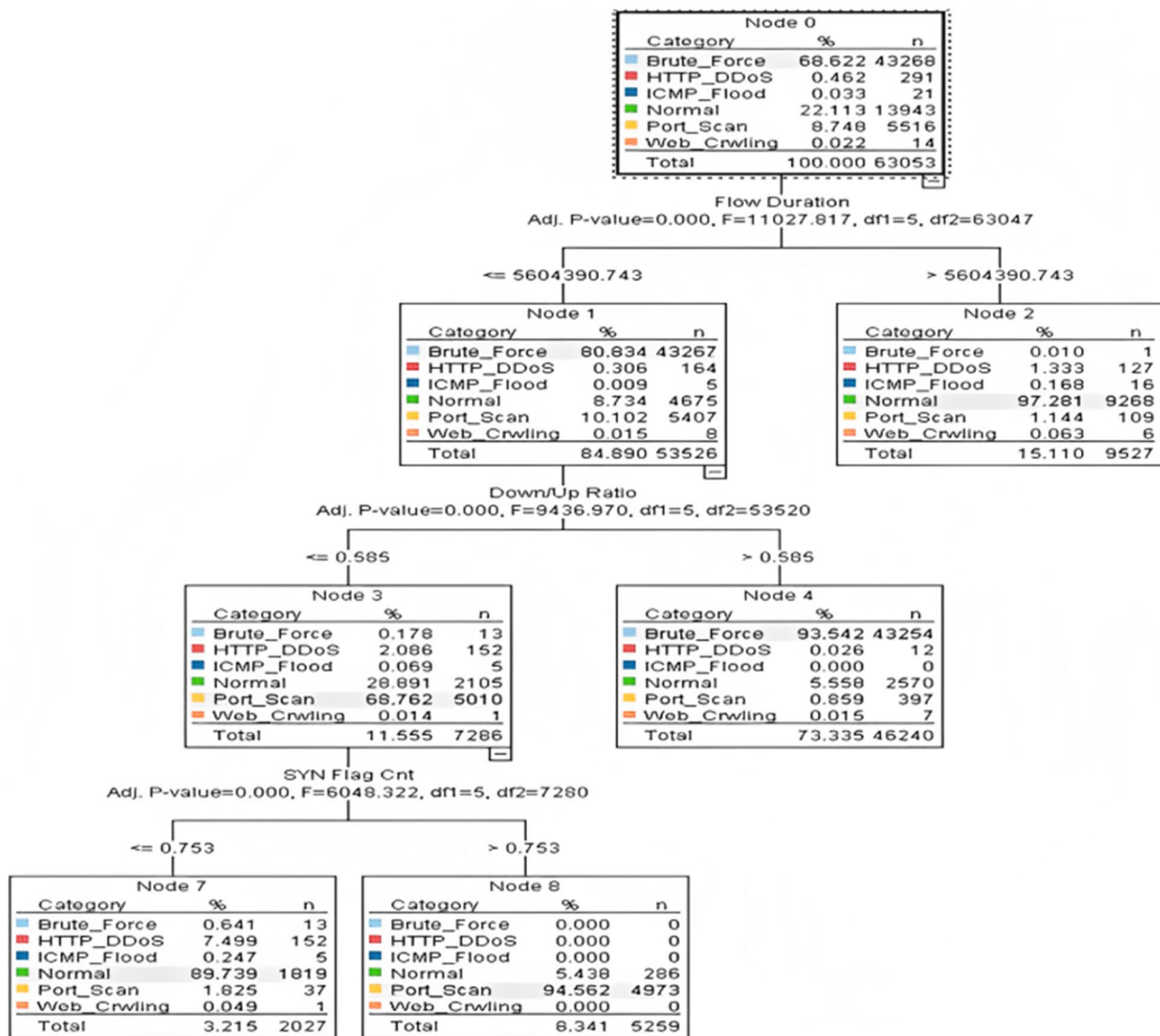


Fig. 1 Quest model

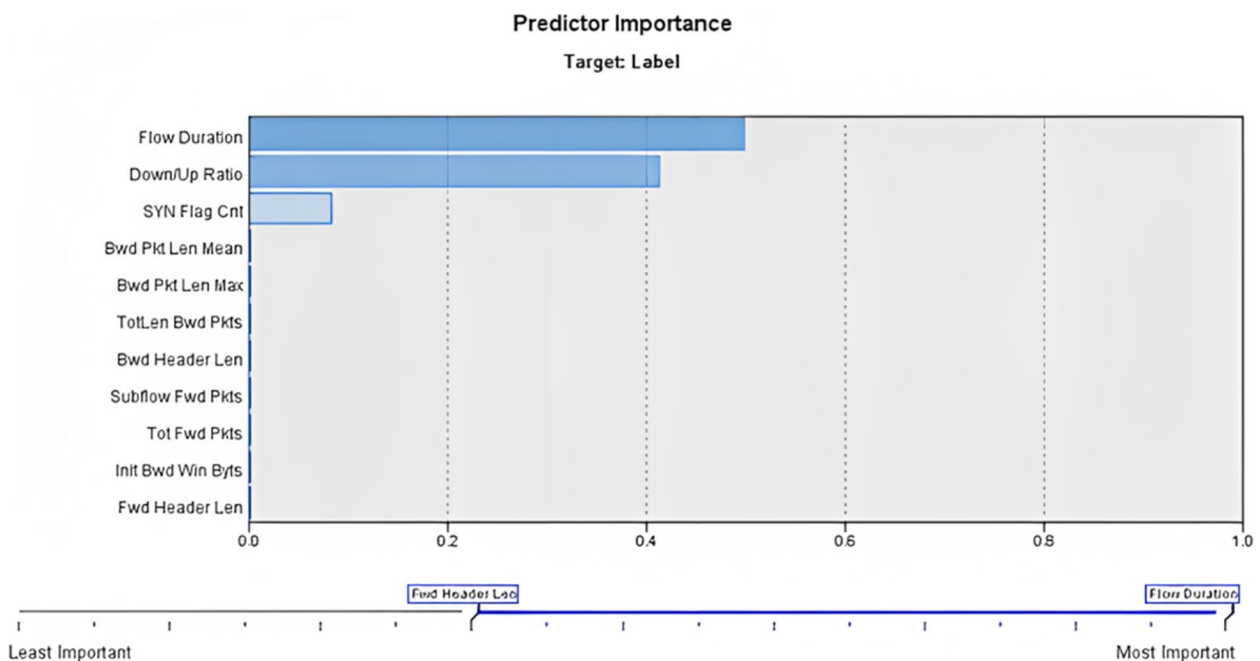
some harmful and others benign, illustrated this problem well [13]. Artificial intelligence (AI) was developed to track these attacks in the artificial brainpower arena. Rule-based and exception-recognition-based setups were among the measurable approaches. Incorporating social anomaly detection and event succession tracking into AI was a natural progression. Interrupt recognition is frequently performed online, and security examinations conducted unconnected both use artificial awareness.

Research [14] introduced an original ID that takes advantage of logical data as Pattern-of-Life (PoL) and data connected with master judgment on the organization’s conduct. This IDS zeroed in on distinguishing an MSA continuously, without a past preparation process. The fundamental objective of the MSA was to make a Point of Entry (PoE) to an objective machine, which could be utilized as a component of an APT assault. Our outcomes check that the utilization of context-oriented data works on the productivity of our IDS by improving the identification pace of MSAs by 58%. Research [15] introduced a methodology that gathered and corresponded cross-space digital danger data to recognize multi-stage digital assaults in energy data frameworks. To give an excellent premise to relevantly evaluate and comprehend what is happening to savvy lattices in the event of facilitated digital assaults, they required a precise and reasonable way to deal with distinguishing digital episodes.

They researched the materialness and execution of the introduced connection approach. They examined the outcomes to feature difficulties in space explicit discovery components.

Research [16] proposed a methodology for assault mining and location that performed errands of caution relationship, misleading positive end, assault mining, and assault expectation. To speed up the quest for the separated caution grouping information to mine assault designs, the Prefix Span calculation was additionally refreshed in the store system. The refreshed Prefix Span expanded the handling proficiency and accomplished an improved outcome than the first one in tests. With the Bayesian hypothesis, the changing likelihood for the grouping design string was determined, and the alert progress likelihood table was built to draw the assault diagram. At last, long-momentary memory organization and word-vector strategies were utilized to perform an online forecast. Consequences of mathematical tests show that the strategy proposed in this paper had severe strength areas for an incentive for assault discovery and expectation.

Research [17] proposed MAAC, a multi-step assault ready connection framework, which decreased rehashed cautions and consolidated multi-step assault ways in light of ready semantics and assault stages. Progressed digital assaults incorporated numerous stages to accomplish a definitive objective. Conventional interruption



**Fig. 2** Predictor importances in quest model



identification frameworks, for example, endpoint security the board instruments, firewalls, and other checking devices, produced numerous cautions during the assault. These cautions included assault hints and numerous bogus up-sides inconsequential to assaults. The assessment consequences of this present reality datasets demonstrated that MAAC could successfully decrease the cautions by 90% and track down assault ways from countless alarms.

Research [18] concentrated on the occasion set off multi-step model prescient control for the discrete-time nonlinear framework over correspondence networks affected by parcel dropouts and digital assaults. In the first place, it was equipped for deciding if the tested sign should be conveyed to the questionable organization and was intended to streamline correspondence assets. Second, two Bernoulli processes were acquainted with addressing the arbitrarily happening parcel dropouts in the questionable

organization and the haphazardly happening misdirection assaults on the actuator side from the enemies. Also, the outcomes on the recursive plausibility and shut circle solidness connected with the arranged framework were accomplished, which unequivocally think about the outside aggravation and info requirement. At last, re-enactment probes of the mass-spring-damping framework were completed to delineate the judiciousness and adequacy of the control methodology. After studying various existing works, Table 1 depicts the summary of existing work as given below.

Table 1 depicts the accuracy level of different techniques in predicting cyber attacks. As shown in this table, the maximum level of accuracy is 97%. This fact motivates the authors to propose a new machine learning model for predicting Multi-stage Cyber attack in Cloud Environment more accurately to make the cloud applications more secure in real time.

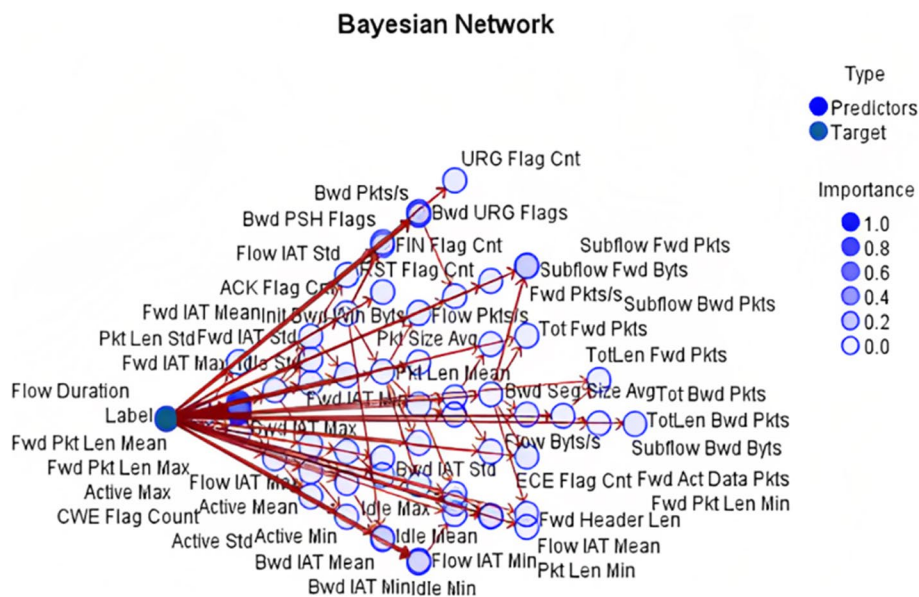


Fig. 3 Bayesian network

**Conditional Probabilities of Label**

Probability					
Brute_Force	HTTP_DDoS	ICMP_Flood	Normal	Port_Scan	Web_Crwing
0.69	0.00	0.00	0.22	0.09	0.00

Fig. 4 Conditional probabilities of level target variable

### Conditional Probabilities of URG Flag Cnt

Parents		Probability	
Bwd URG Flags	Label	<= 0.5	> 0.5
<= 0.5	Brute_Force	1.00	0.00
<= 0.5	HTTP_DDoS	1.00	0.00
<= 0.5	ICMP_Flood	1.00	0.00
<= 0.5	Normal	1.00	0.00
<= 0.5	Port_Scan	1.00	0.00
<= 0.5	Web_Crwing	1.00	0.00
> 0.5	Port_Scan	0.00	1.00

Fig. 5 Conditional probabilities of URG flag Cnt input variable

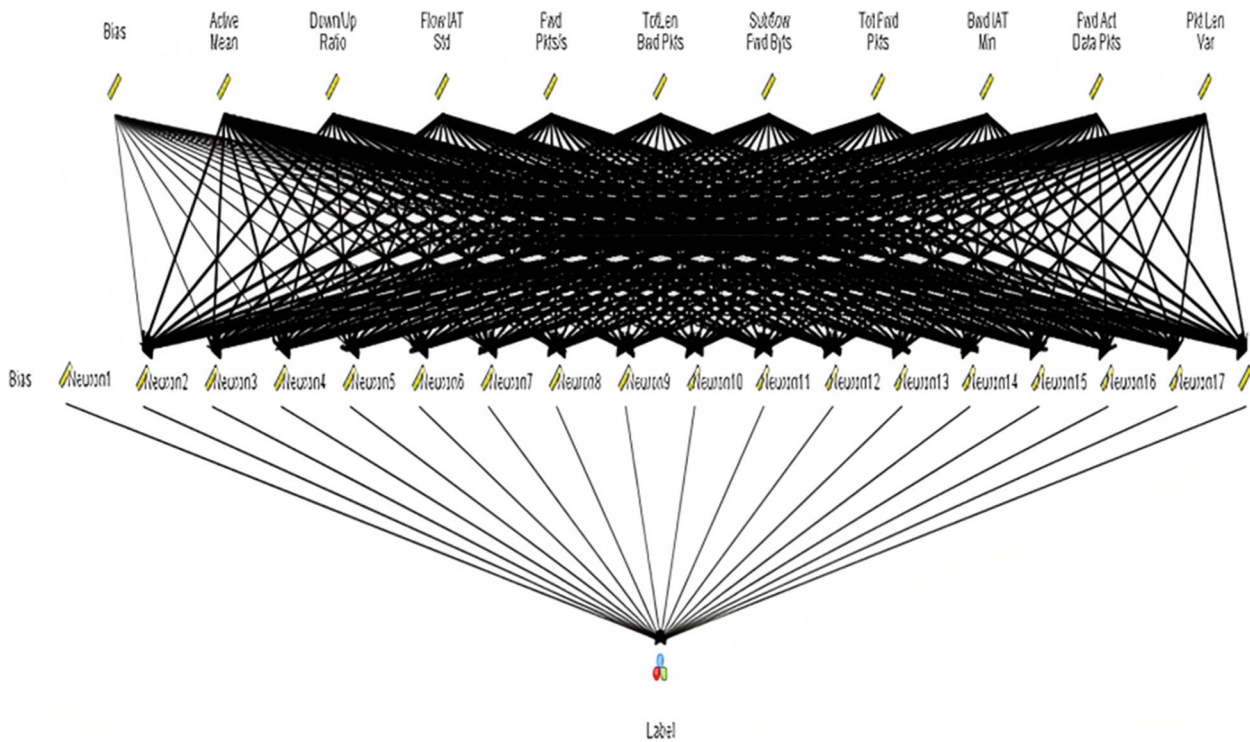


Fig. 6 Neural network for the current problem

**Table 2** Description of neural network for the current problem

PointType	X	Y	V4
scale	0	63.0303	Bias
scale	0	62.0606	Active Mean
scale	0	61.0909	Down/Up Ratio
scale	0	60.1212	Flow IAT Std
scale	0	59.1515	FwdPkts/s
scale	0	58.1818	TotLenBwdPkts
scale	0	57.2121	SubflowFwdByts
scale	0	56.2424	Tot FwdPkts
scale	0	55.2727	Bwd IAT Min
scale	0	54.303	Fwd Act Data Pkts
scale	0	53.3333	Pkt Len Var
scale	0	52.3636	Active Std
scale	0	51.3939	Bwd IAT Std
scale	0	50.4242	FwdPkt Len Std
scale	0	49.4545	FwdPkt Len Max
scale	0	48.4848	SubflowBwdByts
scale	0	47.5152	Fwd Header Len
scale	0	46.5455	SubflowBwdPkts
scale	0	45.5758	TotLenFwdPkts
scale	0	44.6061	CWE Flag Count
scale	0	43.6364	Flow IAT Min
scale	0	42.6667	Pkt Size Avg
scale	0	41.697	Flow IAT Mean
scale	0	40.7273	FwdPkt Len Min
scale	0	39.7576	FwdPkt Len Mean
scale	0	38.7879	FIN Flag Cnt
scale	0	37.8182	Fwd IAT Std
scale	0	36.8485	Active Max
scale	0	35.8788	Bwd PSH Flags
scale	0	34.9091	Flow Pkts/s
scale	0	33.9394	Flow IAT Max
scale	0	32.9697	Idle Min
scale	0	32	BwdPkt Len Min
scale	0	31.0303	SubflowFwdPkts
scale	0	30.0606	Idle Std
scale	0	29.0909	RST Flag Cnt
scale	0	28.1212	Bwd Header Len
scale	0	27.1515	InitBwd Win Byts
scale	0	26.1818	Pkt Len Min
scale	0	25.2121	Idle Mean
scale	0	24.2424	Bwd IAT Tot
scale	0	23.2727	Fwd IAT Min
scale	0	22.303	Bwd IAT Max
scale	0	21.3333	Active Min
scale	0	20.3636	Bwd IAT Mean
scale	0	19.3939	Fwd IAT Max
scale	0	18.4242	PSH Flag Cnt
scale	0	17.4545	Flow Byts/s
scale	0	16.4848	BwdPkts/s



**Table 2** (continued)

PointType	X	Y	V4
scale	0	15.5152	ECE Flag Cnt
scale	0	14.5455	Idle Max
scale	0	13.5758	FwdSeg Size Avg
scale	0	12.6061	Pkt Len Max
scale	0	11.6364	Tot BwdPkts
scale	0	10.6667	Pkt Len Std
scale	0	9.697	ACK Flag Cnt
scale	0	8.7273	Flow Duration
scale	0	7.7576	BwdSeg Size Avg
scale	0	6.7879	Fwd IAT Mean
scale	0	5.8182	SYN Flag Cnt
scale	0	4.8485	BwdPkt Len Std
scale	0	3.8788	Pkt Len Mean
scale	0	2.9091	BwdPkt Len Mean
scale	0	1.9394	BwdPkt Len Max
scale	0	0.9697	Fwd IAT Tot
scale	1	60.6316	Bias
scale	1	57.2632	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	53.8947	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	50.5263	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	47.1579	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	43.7895	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	40.4211	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	37.0526	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	33.6842	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	30.3158	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	26.9474	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	23.5789	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	20.2105	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	16.8421	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	13.4737	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	10.1053	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	6.7368	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
scale	1	3.3684	Hidden layer activation: Hyperbolic tangent Output layer activation: Softmax
set	2	32	Label

## Primitives

### Quest model

QUEST stands for “Quick, Unbiased, and Efficient Statistical Tree” and is a decision tree construction algorithm that uses binary classification. The primary goal of its creation was to speed up the processing of massive C&R Tree studies involving a considerable number of variables or a large number of instances [19, 20]. The second motivation for QUEST was to mitigate the preference that classification tree techniques have for inputs

that provide more splits, such as continuous (numeric range) input fields or those with multiple categories [13]. To assess the value of a node’s input fields, QUEST employs a series of rules based on statistical significance tests. It is possible that each input to a node only needs to be tested once to make a selection. When analyzing an input field, not all splits are checked like in C&R Tree, and not all possible combinations of categories are tried like in C&R Tree and CHAID. This expedites the logical process [21].

Figure 1 depicts the nodes generated in the Quest model to predict the attack. Quadratic discriminant analysis is used to the specified input to find the splits between the classes of interest. Figure 1 shows the total of 8 nodes. Each node consists of three columns (Category, %, and n) [22]. The branch of the decision tree shows the values of Adj. P value, F value, and df values. Again, this strategy outperforms exhaustive search (C&R Tree) in terms of performance and finds the best possible partition [23].

Figure 2 highlights the predictors’ importance in the Quest model for predicting the attack. It highlights the predictors into two categories: Least Important and Most Important, as shown in Fig. 2. Essential predictors are actively involved in the prediction of cyber attacks [24, 25]. The destination field must be categorical, although the input fields might be continuous (numerical ranges). Any division may be made into two equally good halves. You can’t utilize weight fields. Any fields in the model that represent an ordinal (ordered set) must be able to store numbers (not strings). The Reclassify node can be used to adjust if necessary [26].

**Table 3** Confidence matrix for quest model

‘Partition’= 1_Training			
	Brute_Force	Normal	Port_Scan
Brute_Force	61,954	19	0
HTTP_DDoS	18	405	0
ICMP_Flood	1	33	0
Normal	3,689	15,791	410
Port_Scan	533	200	7,107
Web_Crawling	10	9	0
‘Partition’= 2_Testing			
	Brute_Force	Normal	Port_Scan
Brute_Force	26,519	10	0
HTTP_DDoS	7	211	0
ICMP_Flood	2	9	0
Normal	1,596	6,830	186
Port_Scan	212	77	2,952
Web_Crawling	4	5	0

Flow Duration <= 5604390.743 [Mode: Brute\_Force ] (53,526)  
 Down/Up Ratio <= 0.585 [Mode: Port\_Scan ] (7,286)  
     SYN Flag Cnt<= 0.753 [Mode: Normal ] => Normal (2,027; 0.897)  
     SYN Flag Cnt> 0.753 [Mode: Port\_Scan ] =>Port\_Scan (5,259; 0.946)  
 Down/Up Ratio > 0.585 [Mode: Brute\_Force ] =>Brute\_Force (46,240; 0.935)  
 Flow Duration > 5604390.743 [Mode: Normal] => Normal (9,527; 0.973)

QUEST, like CHAID but unlike C&R Tree, employs statistical tests to determine whether or not an input field is utilized. It also decouples the input selection and splitting problems so that they may be addressed independently using distinct criteria [27]. In CHAID, on the other hand, the split is determined by the same statistical test result that selects the variables to study. The impurity-change measure is also used in C&R Tree for input field selection and branching [28].

**Bayesian network**

A prevalent type of probabilistic graphical model is the Bayesian network. They have a framework, and then there are the parameters. A directed acyclic graph (DAG)

shows interdependencies and conditional independences between random variables at each node [29].

Figure 3 depicts the predictors and targets in Bayesian Network for predicting the attack. The color code represents the level of different importance. The Fig. 3 splits the attributes into two categories indicated by different colors. One category is Predictors, and another is Target. The importance of predictors lies between 0 and 1 (0.0, 0.2, 0.4, 0.6, 0.8 & 1.0). The parameters are a set of node-specific conditional probability distributions. Suppose you need a compact, adaptable, and easily interpretable representation of a joint probability distribution [30]. In that case, a Bayesian network is a way to go. Furthermore,

because directed acyclic networks permit modeling causal links between variables, they are valuable in knowledge discovery. A Bayesian network is often trained using empirical data [31].

Figure 4 highlights the values of conditional Probabilities of the Level Target variable. The probabilities of these categories of cyber attacks are shown in Fig. 4. The probability of Brute Force is 0.69, and the port scan is 0.09. For a Bayesian network to function, the idea of conditional independence must be front and centre. A random variable is considered independent if it does not affect other variables [32].

Figure 5 displays the conditional probabilities of the URG Flag CNT Input variable concerning the target level variable. This figure depicts the conditional probability of the URG flag into two categories. These categories lie depending on the parent values. In probability

theory, a dependent variable is one whose livelihood depends on the values of one or more independent variables. A given random variable may be conditionally independent of one or more other random variables, describing the connection between numerous random variables [33]. This does not imply that the variable is unrelated to any known random variables; instead, it provides a detailed description of the independence of the variable in question [34]. An example of a probabilistic graphical model is a Bayesian Network, which allows for including unknown (latent) variables while still articulating all of the conditional independence requirements for the known variables [35].

**Neural network**

A Neural network is an assortment of calculations in light of a harsh model of the human cerebrum. Marking

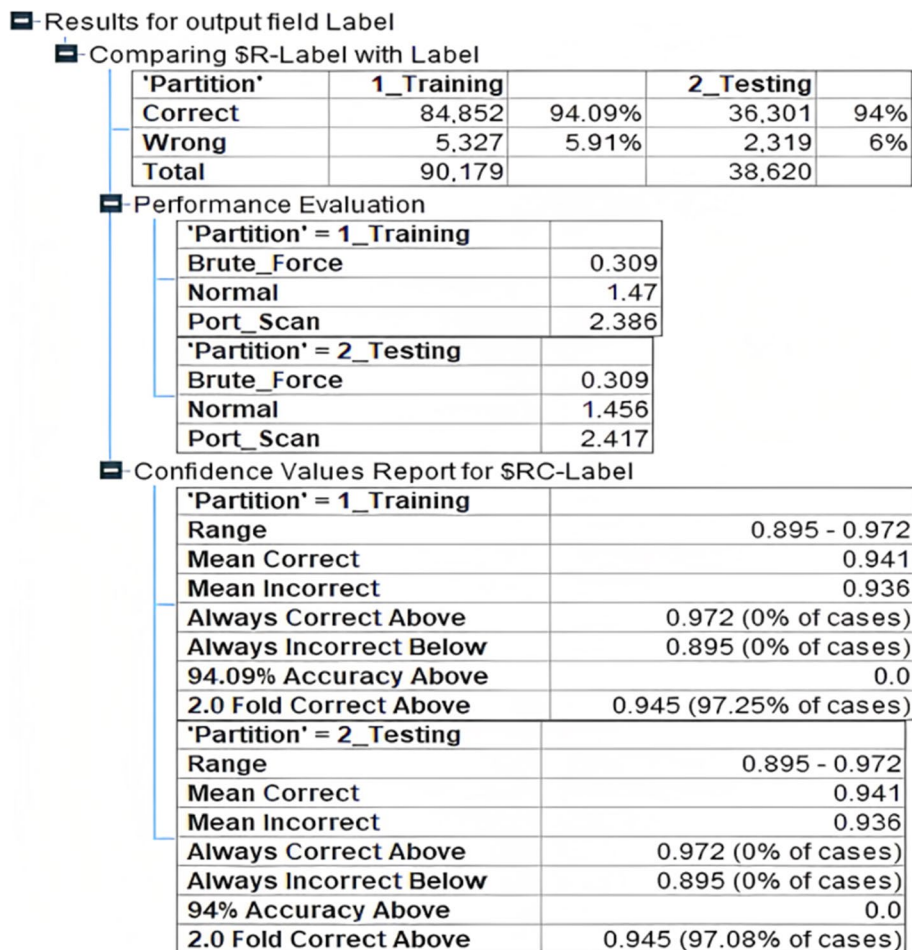


Fig. 7 Accuracy level in quest model

or gathering crude information is one way they decipher tactile information as machine discernment. All certifiable information, including pictures, sounds, text, and time series, should be converted into the mathematical examples they comprehend, which are put away in vectors [36]. We can utilize brain organizations to arrange and classes information. Bunching and grouping might be considered a layer on top of the information you store and make due. Utilizing a marked dataset to prepare, they help bunch unlabeled information in light of similitudes among the model information sources [37].

**Artificial neural networks architecture**

Neural networks work in a way like that of neurons in the human sensory system. Warren S McCulloch and Walter Pitts concocted the expression “Brain Networks” in the mid-1970s. We should take a gander at the design of ANNs to find out how they work. A neuron is enclosed by its membrane. The membrane on the end bulb is called the presynaptic membrane, and the membrane upon which the end bulb strikes is called the postsynaptic membrane. There are three crucial layers in a brain organization [38].

- Input Layers: An ANN’s initial layer, the input layer, accepts input data in text, numbers, audio files, picture pixels, etc. It is responsible for parsing this data.
- Hidden Layers: The hidden layers of the ANN model may be found in the centre. As in the case of a perceptron, there can be only one hidden layer, or there can be several. These hidden layers ex-

cute various mathematical computations on the incoming data and detect the patterns that are part of them.

- Output Layer: The result of the center layer’s careful calculations is acquired in the result layer. Various variables and hyper-boundaries impact the model’s exhibition in a brain organization. These boundaries altogether affect the result of ANNs. Weights, biases, learning rates, batch sizes, etc., are some of these factors. The ANN’s nodes are all equally important [39, 40].

Figure 6 displays the neural network designed for the current problem. It consists of 18 neurons and 3 Biases on the network. The Hopfield model is both an optimization model as well an association model. Hopfield is a constraint satisfaction algorithms-based model. It is symmetric and asynchronous in nature as compared to perceptron. As a node in a network, each one has its unique weight [41]. The transfer function is employed in conjunction with the bias to calculate the weighted total of the inputs and bias. There are nodes in each tier. In a node, computing occurs similarly to how neurons activate when they receive enough input in the human brain [42]. When an algorithm is trying to learn how to classify data, a node uses coefficients or weights to either amplify or dampen each input. This helps the algorithm decide which inputs are most important for learning how to classify data correctly. Node activation functions evaluate the total of these input-weighted products to see if and to what degree that signal should be sent through the network, for example, to affect the

**Table 4** Confidence matrix for Bayesian network

'Partition'= 1_Training							
	Brute_Force	HTTP_DDoS	ICMP_Flood	Normal	Port_Scan	Web_Crawling	
Brute_Force	61,919	48	0	5	1	0	
HTTP_DDoS	3	404	0	12	4	0	
ICMP_Flood	0	0	9	25	0	0	
Normal	1,102	221	1	17,971	567	28	
Port_Scan	201	66	0	147	7,425	1	
Web_Crawling	1	1	0	12	0	5	
'Partition'= 2_Testing							
	Brute_Force	HTTP_DDoS	ICMP_Flood	Normal	Port_Scan	Web_Crawling	
Brute_Force	26,506	19	0	4	0	0	
HTTP_DDoS	2	210	0	5	1	0	
ICMP_Flood	0	0	2	9	0	0	
Normal	493	91	0	7,748	248	23	
Port_Scan	70	34	0	68	3,069	0	
Web_Crawling	2	1	0	8	0	0	

outcome. The neuron is said to be “activated” if the signals flow through it. Here’s a visual representation of a single node [43].

Table 2 depicts the current problem’s full description of the Neural Network. This table showed that the hyperbolic tangent is being used for Hidden layer activation, and Soft-max has been applied in Output layer activation. It is a line of those neuron-like switches that turn on or off when info is sent through the net. Beginning with an underlying information layer that accepts your information, the result of one layer is simultaneously the following contribution. We pair the model’s alterable loads with those elements to give weight to enter attributes in the brain organization’s characterization and bunching process. The quantity of hub layers that information should course through in a multi-step

design acknowledgment process recognizes profound gaining networks from more traditional single-stowed away layer brain organizations [44].

The early perceptron, for instance, were shallow brain networks with just a piece of single information, one result layer, and a solitary secret layer between them. Multiple layers of information and result characterize “Profound” learning. The expression “so significant” isn’t simply a trick to cause calculations to seem like they’ve understood Sartre and pay attention to dark musical crews. This expression has more than one hidden layer with an unmistakable significance [45]. The ANNs can be classified in several ways. Based on Connection types, we can have Static (feed-forward) or Dynamic (feedback) ANN, whereas based on topology, we can have a Single layer, Multilayer, Recurrent

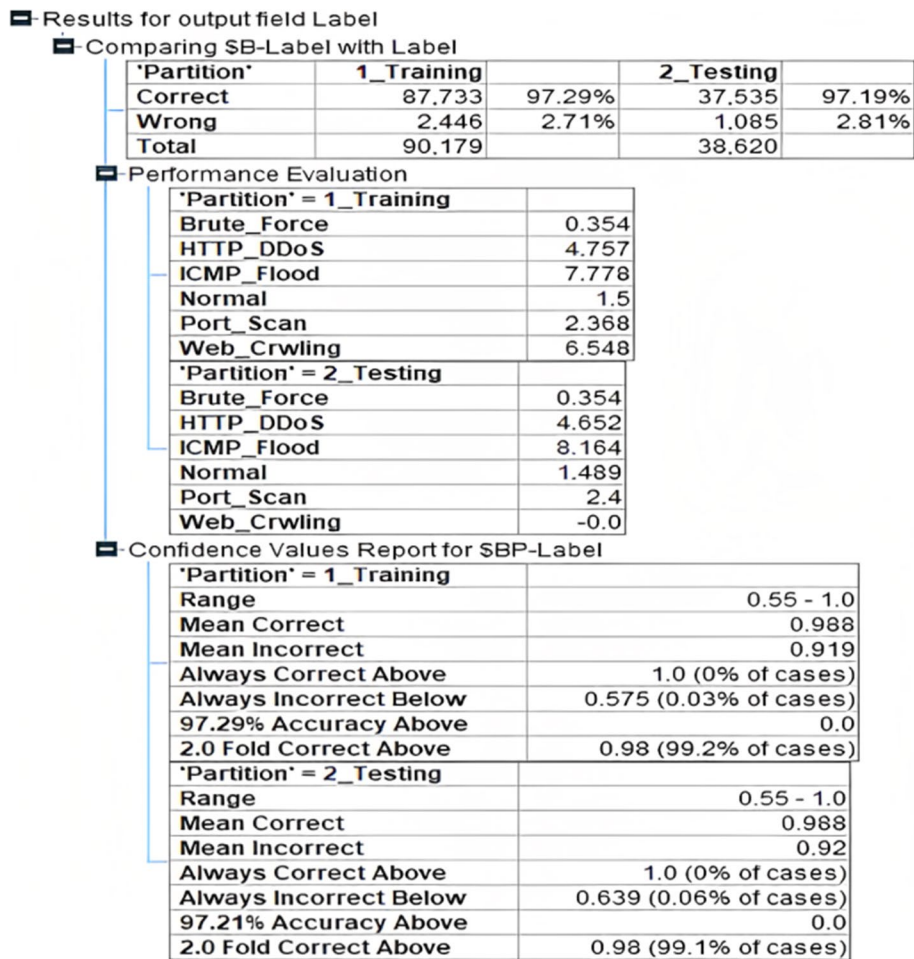


Fig. 8 Accuracy level in Bayesian network



or Self-organized type of Artificial Neural Network. However, we have classified types of ANN on the following basis:

Based on Interconnection:

- Feed Forward.
- Feedback/Recurrent Networks.

Based on the Functionality:

- Perceptron Network.
- Back Propagation Network.
- Hopfield Network.
- Cascading neural network.
- Counter propagation network.

In a feed-forward network, the output is constant and is a function of network input only. The feed-forward neural network is used if propagating and combining activation successively in a forward direction from input to output to characterize the inference behavior. Nodes in deep-learning networks use the output of previous layers to train on new sets of features. Nodes in the neural network increasingly recognize ever-more-complex characteristics as you progress through the layers since they collect and recombine information from previous layers [46]. As an intermediate step between explicitly providing output and altogether withholding any input on how a system is performing, reinforcement learning is a hybrid approach that combines the benefits of both approaches. Instead of providing specifics on what an ideal output response for each network input should look like, reinforcement learning gives the algorithm merely a letter grade. The network's grade (or score) may be calculated using a set of inputs.

In comparison to supervised learning, this style of education is rare. System control looks to be its most vital advantage. In other words, this learning involves rewarding and penalizing action in an environment w.r.t. attaining a specific goal [47].

## Materials and methods

### Dataset

This dataset contains seven files. Two scenarios for multi-step cyber-attacks are included in the MSCAD [48]. Following are the two multi-step assault scenarios:

- Multi-step Attack Scenario A: In this scenario, an attacker wants to undertake a brute force attack

(password cracking) on any host in the target network. The attacker uses three key phases to carry out this assault. Port scanning was carried out simultaneously. As a second option, the HTTrack Website Copier was utilized to capture an offline copy of web application pages. Four hundred seventy attempts were made to break the password using a list of 47 items and a user list of 10 entries. Eventually, the Brute force script was run, and it was successful.

- Multi-step Attack Scenario B: For example, in scenario B, the attacker attempts to conduct a volume-based DDoS on any host in the target network, regardless of location. Volume-based DDoS was executed in three phases. "A port scan assault (Full, SYN, FIN, and UDP Scan) is the initial phase of a volume-based DDoS attack. To begin the DDoS assault on the APP, we will use HTTP Slowloris DDoS. They were finally utilizing the Radware tool to carry out the volume-based DDoS assault. The volume-based DDoS assault took an hour to infect three hosts (192.168.159.131, 192.168.159.14, and 192.168.159.16).

### Data pre-processing

The first step in developing a machine learning model is gathering data. Machine learning models may be trained to analyze raw data in a way that is intelligible to the model. This is a critical step, and it should be carried out

**Table 5** Confidence matrix for neural network

'Partition' = 1_Training				
	Brute_Force	HTTP_DDoS	Normal	Port_Scan
Brute_Force	61,881	5	75	12
HTTP_DDoS	3	398	15	7
ICMP_Flood	0	0	34	0
Normal	215	1	19,558	16
Port_Scan	81	25	226	7,508
Web_Crwlng	0	1	17	1
'Partition' = 2_Testing				
	Brute_Force	HTTP_DDoS	Normal	Port_Scan
Brute_Force	26,492	2	31	4
HTTP_DDoS	2	206	6	4
ICMP_Flood	0	0	11	0
Normal	81	4	8,484	43
Port_Scan	30	12	99	3,100
Web_Crwlng	1	1	6	1

■ Results for output field Label

■ Comparing \$N-Label with Label

'Partition'	1_Training		2_Testing	
Correct	89,345	99.08%	38,282	99.12%
Wrong	834	0.92%	338	0.88%
Total	90,179		38,620	

■ Performance Evaluation

<b>'Partition' = 1_Training</b>	
Brute_Force	0.37
HTTP_DDoS	5.285
Normal	1.493
Port_Scan	2.425
<b>'Partition' = 2_Testing</b>	
Brute_Force	0.371
HTTP_DDoS	5.089
Normal	1.483
Port_Scan	2.461

■ Confidence Values Report for \$NC-Label

<b>'Partition' = 1_Training</b>	
Range	0.34 - 1.0
Mean Correct	0.991
Mean Incorrect	0.798
Always Correct Above	1.0 (0% of cases)
Always Incorrect Below	0.34 (0% of cases)
99.08% Accuracy Above	0.0
2.0 Fold Correct Above	0.867 (99.54% of cases)
<b>'Partition' = 2_Testing</b>	
Range	0.369 - 1.0
Mean Correct	0.99
Mean Incorrect	0.778
Always Correct Above	1.0 (0% of cases)
Always Incorrect Below	0.369 (0% of cases)
99.12% Accuracy Above	0.0
2.0 Fold Correct Above	0.797 (99.56% of cases)

Fig. 9 Accuracy level in neural network

correctly. Models are more accurate when they access a well-prepared dataset [49].

*Handling missing values* Numerous real-world data sets contain placeholders, such as blanks, NaNs, or other missing values, due to the nature of many of these datasets. When it comes to scikit-estimators learning and their assumption that every value is numerical and has meaning, such datasets don't work well. The most common technique for dealing with incomplete datasets is to remove all rows and columns that contain missing values. However, this may result in the loss of crucial data (even though incomplete). A preferable technique is to infer the missing values from the existing data, i.e., the imputation of the missing values [50]. The Simple Imputer class developed in this work provides basic techniques for imputing missing values. The statistics of the columns where the missing values are situated (mean, median, or most common) can be used to fill in the blanks. This class also supports different missing value encodings.

*Handling Outliers* An outlier in a dataset is an item significantly different from the rest of the dataset. On the other hand, this definition leaves much leeway for the data analyst to determine what constitutes an anomaly. The outliers may result from measurement mistakes, execution flaws, sample difficulties, and improper data input. In statistical modeling, removing outliers is critical since

their existence increases mistakes, introduces bias, and has significant effects. To determine how many standard deviations a data point is from the mean, you may use the Z-Score (or Standard Score). The farther the data point is from the mean, the higher the Z-score. This is critical because most data points are close to the mean in an adequately distributed data collection. A data point with a big Z-score is likely to be an outlier since it's distant from the average. The Z-score in this study was generated using Scipy's function [51].

**Extremely boosted neural network for multi-stage Cyber attack prediction**

Currently, most neural network ensemble approaches aggregate all the available neural networks into one large group. However, the efficacy of this method has yet to be adequately demonstrated. When analyzing how the ensemble and its neural networks interact, combining many neural networks is proven more advantageous than all the available ones. This concept has the potential to be applied to the formation of compelling ensembles. Neural network ensemble learning is the practice of using many neural networks to solve a problem. As a result of regression and classification, this study found that ensembles with only a few rather than all available neural networks may be more effective. The majority now makes systems predictions using a mix of neural networks [52].

**Classification for Label**  
Overall Percent Correct = 99.1%

Observed	Predicted					
	Brute_Force	HTTP_DDoS	ICMP_Flood	Normal	Port_Scan	Web_Crawling
Brute_Force	99.9%	0.0%	0.0%	0.1%	0.0%	0.0%
HTTP_DDoS	0.7%	94.1%	0.0%	3.5%	1.7%	0.0%
ICMP_Flood	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%
Normal	1.1%	0.0%	0.0%	98.3%	0.6%	0.0%
Port_Scan	1.0%	0.3%	0.0%	2.9%	95.8%	0.0%
Web_Crawling	0.0%	5.3%	0.0%	89.5%	5.3%	0.0%

Fig. 10 Accuracy level in different attacks

**The learning algorithm is described as follows:-**

Z is an input matrix, and T is an output matrix.

p training pairs are { z1 d1 ,z2 d2 , .....zpdp}.

zi (I\*1) be a input vector and di (K\*1)be a output vector. i=1,2,.....p.

ith component of each zi is -1.

J is some neurons in the hidden layer.

η, be the step size or the rate of learning.

Step 1. Weights are initialized to a small random number.  
Let W(K\*J) be the weight matrix for the output layer and V(J\*I) be the weight matrix for the hidden layer.

Step 2. Choose the maximum acceptable error emax .and learning rate η>0.  
Initialize e←0. Let n←1.

Step 3. Output for the hidden layer is calculated as follows:  
yj← f( vitz) for j=1,2,.....J.  
Output for the output layer is calculated as follows:  
ok ←f( wkty) for k=1,2,.....K.  
Where f is the sigmoidal function.

Step 4. An error signal is computed.  
e←(1/2)(dk-ok)<sup>2</sup>+e for k=1,2,.....K.

Step 5. The local gradient for output later is calculated as follows:  
δo=(1/2)(dk-ok)(1-ok) for k=1,2,.....K.  
The local gradient for the hidden layer is calculated as follows:  
k  
δy=(1/2)(1-yj<sup>2</sup>)∑ δokwkj for j=1,2,.....J.  
k=1

Step 6. Weights adjustment for the output layer.  
Δwkj← ηδokyj+Δwkj(n-1) for k=1,2,.....K.  
for j=1,2,.....J.  
Weights adjustment for the hidden layer.  
Δ vji← ηδyzi+Δvji(n-1) for j=1,2,.....J.  
for i=1,2,.....J.

Step 7. Output layer weights are adjusted by:  
wkj← wkj+Δwkj for k=1,2,.....K.  
for j=1,2,.....J.  
Hidden layer weights are adjusted by:  
vji← vji+Δvji for j=1,2,.....J.  
for i=1,2,.....J.

Step 8. If e<emax, then terminate the training process.  
Output weights w, v, and e.  
If e>emax, then e←0,n←n+1and initiate a new training cycle by going to step 3.

Step Size: -

η, be the step size or the rate of learning.

When η is too tiny, then the algorithm becomes unstable. If η is too large, then the algorithm either diverges or oscillates. So maximum stable learning rate is limited by the Hessian matrix's largest eigenvalue (second derivative).

$$\eta < 2/\lambda_{max}, \lambda_{max} \text{ is largest eignvalue.}$$

Generally, η <1/2 for convergence.

**Termination Criteria**

The algorithm stops if one of the following conditions is met.

- Stop after a predefined number of iterations, which the user defines.
- Stop when the mean squared error reaches a low level, which the user defines before the Training begins.

Because it incorporates both a gradient-boosted tree and a feed-forward neural network, our model is resilient across all performance criteria [53]. XBNet is the abbreviation for 'Extremely Boosted Neural Network'. Each design layer prepares trees, and component significance and slope drop loads are used to modify layers where trees have been prepared. As info, XBNet utilizes

crude even information and trains the model utilizing an improvement approach known as Boosted Gradient Descent, which is introduced utilizing the component meaning of a slope-helped tree and afterward refreshes loads of each layer in the brain network in two phases, as shown below.

- Gradient descent is used to update the weight.
- Gradient-boosted trees can be used to update weights.

**Experiment result**

In executing the proposed model, the dataset is divided into the following partitions: 70% training and 30% testing. The proposed model has been implemented in Python language. To test the performance of the model, the dataset consists of the following instance of the Label class as given below [54]:











- Brute\_Force.
- HTTP\_DDoS.
- ICMP\_Flood.
- Normal.
- Port\_Scan.

**Performance of individual machine learning algorithms**

The QUEST, Bayesian Network, and Neural networks are implemented in the first phase of executing the machine learning algorithms to predict multi-stage cyber attacks in the cloud environment. To evaluate the performance of these algorithms, the confidence matrix and accuracy level have been calculated.

- QUEST model

Confidence matrices have been created for both phases of the proposed model to evaluate its performance. An organized decision-making tool, the confidence matrix allows you to assess several possibilities by selecting one of five distinct degrees of confidence for each of the five elements. Certain activities and research may benefit more from considerations specific to such practices and studies. The absence of data or low confidence in a factor does not imply disfavor. Table 3 consists of Confidence Matrix for Quest Model. This table is divided into two parts, i.e., Training and testing. As a systematic decision-making tool, this matrix allows you to assess potential outcomes by selecting a confidence level for each of the abovementioned considerations. Certain considerations may be more important than others in specific contexts [55].

Model	Accuracy	Method	Predictors	Model Size (Synapses)	Records
6	18.5%		63	1056	90,179
10	52.6%		63	1336	90,179
8	53.2%		64	1213	90,179
2	54.2%		62	1110	90,179
9	55.5%		63	1336	90,179
4	62.2%		64	1355	90,179
7	92.1%		65	1374	90,179
5	93.0%		63	1196	90,179
3	94.1%		62	1041	90,179
1	99.7%		66	1247	90,179

**Fig. 11** Accuracy level in different models



Figure 7 depicts the accuracy level achieved in the Quest model. The accuracy level is 94.09% and 94% in the training and testing phases, respectively. This figure also shows the statistical analysis of the model's performance in terms of mean correct. This fact demonstrates the actual working of the model in different stages.

- Bayesian Network

Table 4 highlights the Confidence Matrix for Bayesian Networks in predicting multi-stage cyber-attacks in cloud environments [49, 56, 57, 58]. This table consists of Confidence Matrix for the Bayesian Network. This table is divided into two parts, i.e., Training and testing.

Figure 8 depicts the accuracy level achieved in Bayesian Network. The accuracy level is achieved at 97.29% and 97.19% in the training and testing phase, respectively. It also highlighted the value of mean values of correct in both phases, i.e., Training and testing, as given below:

- Neural Network

Table 5 highlights the Confidence Matrix for Neural Networks in predicting multi-stage cyber-attacks in cloud environments. The confidence Matrix helps to understand the confidence level in the results of various machine learning algorithms [46, 49, 54].

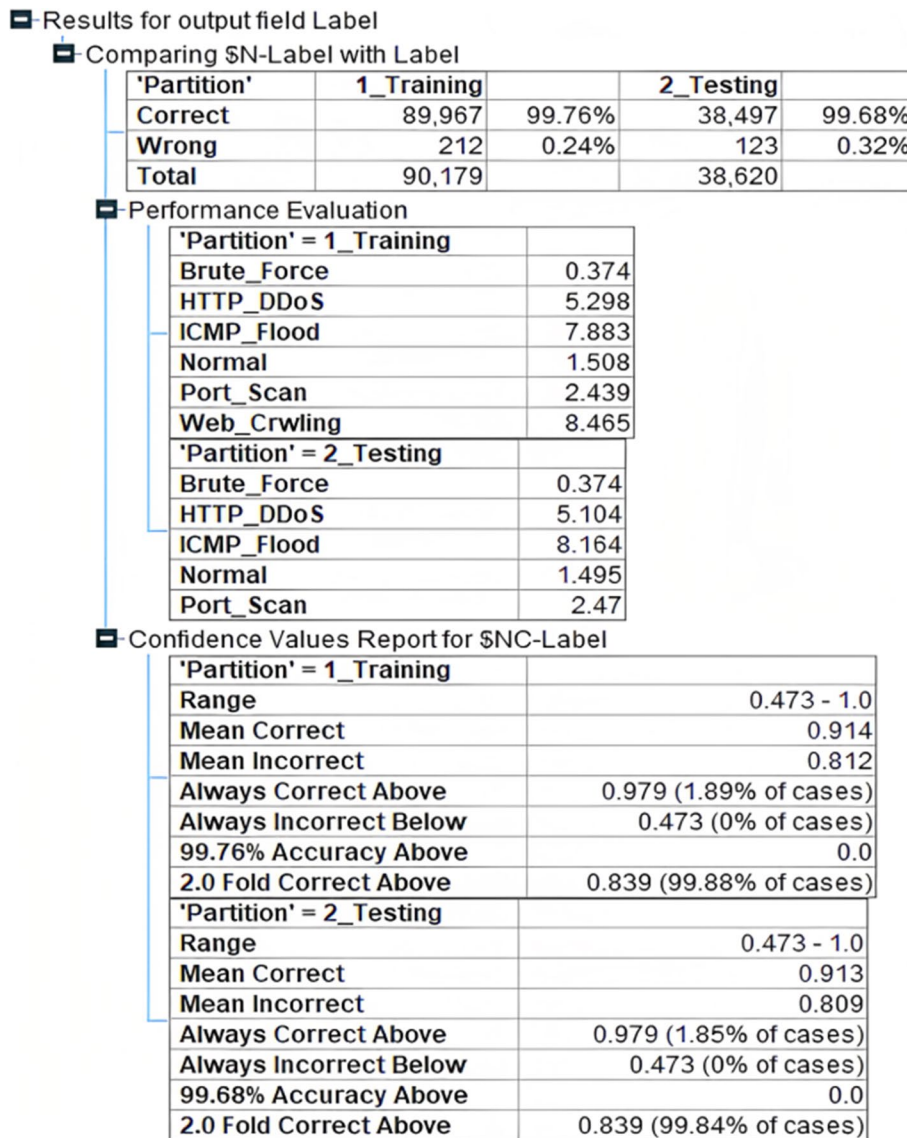


Fig. 12 Accuracy level in the proposed model

**Table 6** Result analysis

No.	Technique	Accuracy
1	Contextual information	58%
2	Cyber Security Game (CSG)	70%
3	multi-step attack alert correlation system	90%
4	Quest Model	94.09%
5	Systematic & coherent approach	97%
6	Bayesian Network	97.29%
7	Neural Network	99.09%
8	Proposed model	99.76%

Figure 9 depicts the accuracy level achieved in Neural Network. The accuracy level is 99.08% and 99.12% in the training and testing phases, respectively.

Figure 10 depicts the accuracy level achieved in neural networks. There are the following accuracy gains in the next attacks as given below:

- Brute\_Force 99.9%.
- HTTP\_DDoS 94.1%.
- ICMP\_Flood 100%.
- Normal 98.3%.
- Port\_Scan 95.8%.
- Web\_Crwing 89.5%.

**Performance of extremely boosted neural network**

We have presented a novel approach to predicting the multi-stage cyber attack using an Extremely Boosted Neural Network. Our general formulation can include

the prediction of attacks at the level of Brute Force, HTTP DDoS, ICMPFlood, Normal, and Ports [59].

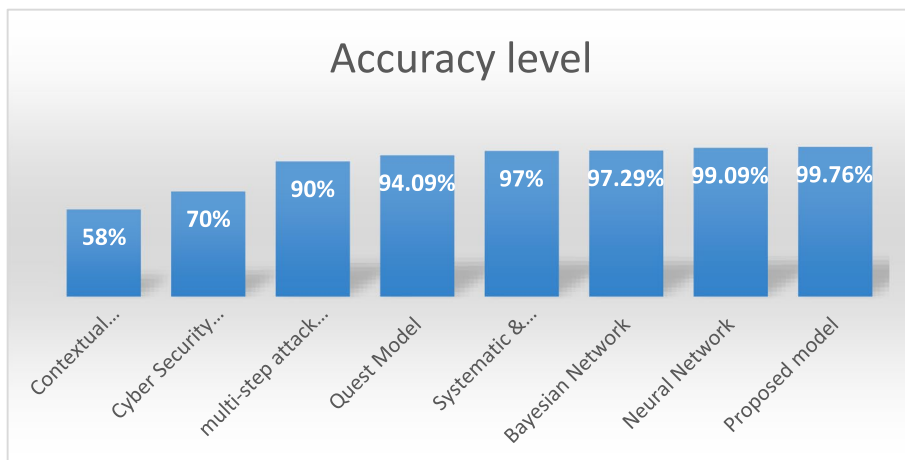
Figure 11 shows the different accuracy levels achieved in different models. It shows the accuracy level of the model's different numbers, and predictors used to predict the cyber attacks. As compared to prior research, this proposed method can generate significantly accurate labels, as presented in Fig. 12.

Lastly, the authors have shown the proposed model's performance with existing techniques studied in the related work and different machine learning algorithms in Table 6. This table demonstrates the accuracy level achieved by seven different techniques, as shown below:

Table 6 may be reconstructed as including the detailed performance of the proposed model. The information in this table is graphically represented by Fig. 13, which proves that the proposed model achieved the maximum level of accuracy (99.798%), as displayed below.

**Conclusion**

The proposed neural network for predicting multi-stage cyber assaults is developed in this study. It puts the intricate assaults into perspective by illustrating how they may be detected and investigated, two of the essential functions in the security area. Here, we outline a complete framework for studying complex assaults, their related analytical methodologies, and their primary uses in security: detection and investigation. This paradigm makes it easier to categorize new, complex dangers and the countermeasures that go along with them, such as Artificial Intelligence. Our model for Multi-stage Cyber attack prediction outperforms other discussed models in terms of accuracy for the given dataset.



**Fig. 13** Result analysis

### Acknowledgements

Authors would like to acknowledge and grateful for the anonymous reviewers for wonderful suggestions.

### Author' contributions

Conceptualization by Surjeet Dalal; Methodology by Bijeta Seth; Design, and methods by Umesh Kumar Lilhore and Deema Mohammed Alsekait; Validation by Poongodi M and Mounir Hamdi; Results and analysis by Sarita Simaiya ; Discussion , Data collection , and Final Draft Review by Surjeet Dalal and Umesh Kumar Lilhore ; Funding Acquisition by Poongodi M and Kaamran Raahemifar; Introduction, Background, and corresponding author by Poongodi M; Conclusion and Review of the first draft by Kaamran Raahemifar. The author(s) read and approved the final manuscript.

### Availability of data and materials

The corresponding author may provide the supporting data on request.

### Declarations

#### Ethics approval and consent to participate

No ethical approval is required, and the authors express their consent to participate in the paper.

#### Consent for publication

Authors provide consent for publication.

#### Competing interests

The authors declare that they have no competing interests.

Received: 13 August 2022 Accepted: 2 November 2022

Published online: 23 January 2023

### References

1. Simaiya S, VinayGautam UK, Lilhore (2021) AtulGarg, PinakiGhosh, Naresh Kumar Trivedi, and AbhineetAnand. "EPPSA: Energy Efficiency Priority Scheduling Algorithm for Cloud Computing." In 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), pp. 1064–1069. IEEE, Lilhore UK, Simaiya S, Maheshwari S, Manhar A, Kumar S (2020) Cloud performance evaluation: hybrid load balancing model based on modified particle swarm optimization and improved metaheuristic firefly algorithms. *Int J Adv Sci Technol* 29(5):12315–12331
2. Torkura KA, Sukmana MH, Cheng F, Meinel C (2020) CloudStrike: Chaos Engineering for Security and Resiliency in Cloud Infrastructure. *IEEE Access* 8:123044–123060. <https://doi.org/10.1109/ACCESS.2020.3007338>
3. Alturki R, Alyamani HJ, Ikram MA, Rahman MA, Alshehri MD, Khan F, Haleem M (2021) Sensor-Cloud Architecture: Taxonomy of Security Issues in Cloud-Assisted Sensor Networks. *IEEE Access* 9:89344–89359. <https://doi.org/10.1109/ACCESS.2021.3088225>
4. Alouffi B, Hasnain M, Alharbi A, Alosaimi W, Alyami H, Ayaz M (2021) A Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies. *IEEE Access* 9:57792–57807. <https://doi.org/10.1109/ACCESS.2021.3073203>
5. Musman S, Turner AJ (2018) A game-oriented approach to minimizing cybersecurity risk. *Int J Saf Secur Eng* 8(2):212–222. <https://doi.org/10.2495/SAFE-V8-N2-212-222>
6. Musman S, Turner A (2018) A game theoretic approach to cyber security risk management. *J Defense Modeling Simul* 15(2):127–146. <https://doi.org/10.1177/1548512917699724>
7. Mirsky Y, Doitshman T, Elovici Y, Shabtai A (2018) Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. pp 1308–1309. <https://doi.org/10.14722/ndss.2018.23204>
8. Parrend P, Navarro J, Guigou F, Deruyver A, Collet P (2018) Foundations and applications of artificial intelligence for zero-day and multi-step attack detection. *Eurasip J Inf Secur* 1:1–21. <https://doi.org/10.1186/s13635-018-0074-y>
9. Aparicio-Navarro FJ, Kyriakopoulos KG, Ghafir I, Lambbotharan S, Chambers JA (2019) Multi-Stage Attack Detection Using Contextual Information. *Proceedings - IEEE Military Communications Conference MILCOM*, 2019-October, 920–925. <https://doi.org/10.1109/MILCOM.2018.8599708>
10. Sen O, Van Der Velde D, Wehrmeister KA, Hacker I, Henze M, Andres M (2021) Towards an approach to contextual detection of multi-stage cyber attacks in smart grids. *SEST 2021–4th International Conference on Smart Energy Systems and Technologies*. <https://doi.org/10.1109/SEST50973.2021.9543359>
11. Hu Z, Yu X, Shi J, Ye L (2021) Abnormal event correlation and detection based on network big data analysis. *Computers Mater Continua* 69(1):695–711. <https://doi.org/10.32604/cmc.2021.017574>
12. Al-Sawwa J, Almseidin M, Alkasasbeh M (2022) Multi-Step Cyber-Attack Dataset (MSCAD) [Data set]. *Kaggle*. <https://doi.org/10.34740/KAGGLE/DSV/3830715>
13. Wang X, Gong X, Yu L, Liu J (2021) MAAC: Novel Alert Correlation Method to Detect Multi-step Attack. *Proceedings – 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2021*, 726–733. <https://doi.org/10.1109/TrustCom53373.2021.00106>
14. Tang X, Wu M, Li M, Ding B (2021) On Designing the Event-Triggered Multistep Model Predictive Control for Nonlinear System Over Networks With Packet Dropouts and Cyber Attacks. *IEEE Trans Cybernetics* 1–13. <https://doi.org/10.1109/TCYB.2021.3062056>
15. George PG, Renjith VR (2021) Evolution of Safety and Security Risk Assessment methodologies towards the use of Bayesian Networks in Process Industries. *Process Saf Environ Prot* 149:758–775. <https://doi.org/10.1016/j.psep.2021.03.031>
16. Kabir S, Papadopoulos Y (2019) Applications of Bayesian networks and Petri nets in safety, reliability, and risk assessments: A review. *Saf Sci* 115:154–175. <https://doi.org/10.1016/j.ssci.2019.02.009>
17. Benrhaïem W, Hafid AS (2020) Bayesian networks based reliable broadcast in vehicular networks. *Veh Commun* 21:100181. <https://doi.org/10.1016/j.vehcom.2019.100181>
18. Ramon J, Palacios-marqués D, Ribeiro-soriano D (2021) Using data mining techniques to explore security issues in smart living environments. *Twitter Comput Commun* 179(September):285–295. <https://doi.org/10.1016/j.comcom.2021.08.021>
19. Lilhore UK, Simaiya S, Guleria K, Prasad D (2020) An efficient load balancing method by using machine learning-based VM distribution and dynamic resource mapping. *J Comput Theor Nanosci* 17(6):2545–2551
20. Ghayvat H, Awais M, Gope P, Pandya S (2021) ReCognizing SUSpect and Predicting The EspRead of Contagion Based on Mobile Phone Location Data (COUNTERACT): A system of identifying COVID-19 infectious and hazardous sites, detecting disease outbreaks based on the Internet of things, edge computing. *Sustainable Cities and Society*, 69(August 2020), 102798. <https://doi.org/10.1016/j.scs.2021.102798>
21. Poongodi M, Malviya M, Kumar C, Hamdi M, Vijayakumar V, Nebhen J, Alyamani H (2022) New York City taxi trip duration prediction using MLP and XGBoost. *Int J Syst Assur Eng Manage* 13(1):16–27
22. Pawar N, Lilhore UK, Agrawal N (2017) A hybrid ACHBDF load balancing method for optimum resource utilization in cloud computing. *Int J Sci Res Comput Sci Engineer-ing Inform Technol* 3307:367–373
23. Poongodi M, Hamdi M, Wang H (2022) Image and audio caps: automated captioning of background sounds and images using deep learning. *Multimed Syst* 26:1–9
24. Poongodi M, Hamdi M, Gao J, Rauf HT (2021), December A Novel Security Mechanism of 6G for IMD using Authentication and Key Agreement Scheme. In 2021 IEEE Globecom Workshops (GC Wkshps) (pp. 1–6). IEEE
25. Makarius EE, Mukherjee D, Fox JD (2019) Fox AK (2020) Rising with the machines: A sociotechnical framework for bringing artificial intelligence into the organization. *Journal of Business Research* 120:262–273. <https://doi.org/10.1016/j.jbusres.2020.07.045>
26. Ramesh TR, Vijayaragavan M, Poongodi M, Hamdi M, Wang H, Bourouis S (2022) Peer-to-peer trust management in intelligent transportation system: An Aumann's agreement theorem based approach. *ICT Express*
27. Simaiya S, Gautam V, Lilhore UK, Garg A, Ghosh P, Trivedi NK, Anand A (2021), October EPPSA: Energy Efficiency Priority Scheduling Algorithm for Cloud Computing. In 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC) (pp. 1064–1069). IEEE
28. Gupta I, Singh AK, Directions F (2022) *IEEE Access*, 10(June), 71247–71277. <https://doi.org/10.1109/ACCESS.2022.3188110>

30. Hamdi M, Bourouis S, Rastislav K, Mohamed F (2022) Evaluation of Neuro Image for the Diagnosis of Alzheimer's Disease Using Deep Learning Neural Network. *Frontiers in Public Health*, 35
31. Ziegler V, Member S, Schneider P, Viswanathan H, Montag M, Kanugovi S, Member S, Rezaki ALI (2021) Security and Trust in the 6G Era. *IEEE Access* 9:142314–142327. <https://doi.org/10.1109/ACCESS.2021.3120143>
32. Ozgur F, Kuzlu M, Catak E, Cali U, Unal D (2022) Security concerns on machine learning solutions for 6G networks in mmWave beam prediction. *Phys Communication* 52:101626. <https://doi.org/10.1016/j.phycom.2022.101626>
33. Poongodi M, Hamdi M, Malviya M, Sharma A, Dhiman G, Vimal S (2022) Diagnosis and combating COVID-19 using wearable Oura smart ring with deep learning methods. *Personal Uniquit Comput* 26(1):25–35
34. Yousefnezhad N, Malhi A, Främpling K (2020) Journal of Network and Computer Applications Security in product lifecycle of IoT devices: A survey. *J Netw Comput Appl* 171(June):102779. <https://doi.org/10.1016/j.jnca.2020.102779>
35. Dalal S, Seth B, Jaglan V, Malik M, Dahiya N, Rani U, Hu YC (2022) An adaptive traffic routing approach toward load balancing and congestion control in Cloud-MANET ad hoc networks. *Soft Comput* 26(11):5377–5388
36. Rathore MS, Poongodi M, Saurabh P, Lilhore UK, Bourouis S, Alhakami W, Hamdi M (2022) A novel trust-based security and privacy model for Internet of Vehicles using encryption and steganography. *Comput Electrical Eng* 102
37. Clusters MCE (2021) SmartX Multi-Sec: A Visibility-Centric Multi-Tiered Security Framework for. *IEEE Access* 9:134208–134222. <https://doi.org/10.1109/ACCESS.2021.3115523>
38. Radoglou-Grammatikis P, Sarigiannidis P, Iturbe E, Rios E, Martinez S, Sarigiannidis A, Eftathopoulos G, Spyridis Y, Sesis A, Vakakis N, Tzovaras D, Kafetzakis E, Giannoulakis I, Tzifas M, Giannakoulis A, Angelopoulos M, Ramos F (2021) SPEAR SIEM: A Security Information and Event Management system for the Smart Grid. *Comput Netw* 193. <https://doi.org/10.1016/j.comnet.2021.108008>
39. Gupta S, Iyer S, Agarwal G, Manoharan P, Algarni AD, Aldehim G, Raahemifar K (2022) Efficient Prioritization and Processor Selection Schemes for HEFT Algorithm: A Makespan Optimizer for Task Scheduling in Cloud Environment. *Electronics* 11(16):2557
40. Balyan AK, Ahuja S, Lilhore UK, Sharma SK, Manoharan P, Algarni AD, Raahemifar K (2022) A Hybrid Intrusion Detection Model Using EGA-PSO and Improved Random Forest Method. *Sensors* 22(16):5986
41. Poongodi M, Bourouis S, Ahmed AN, Vijayaragavan M, Venkatesan KGS, Alhakami W, Hamdi M (2022) A Novel Secured Multi-Access Edge Computing based VANET with Neuro fuzzy systems based Blockchain Framework. *Computer Communications*
42. Fischer-hübner S, Alcaraz C, Ferreira A, Fernandez-gago C, Lopez J, Markatos E, Islami L, Akil M (2021) Journal of Information Security and Applications Stakeholder perspectives and requirements on cybersecurity in Europe. *J Inform Secur Appl* 61(June):102916. <https://doi.org/10.1016/j.jisa.2021.102916>
43. Gomez-fernandez M, Higley K, Tokuhiko A, Welter K, Wong W (2020) Status of research and development of learning-based approaches in nuclear science and engineering: A review. *Nuclear Eng Design* 359
44. Guenduez AA, Mettler T (2022) Strategically constructed narratives on artificial intelligence: What stories are told in governmental artificial intelligence policies? *GovInf Q*. 22
45. Shokry M, Ismail A, Abd-ellah MK (2022) Systematic survey of advanced metering infrastructure security: Vulnerabilities, attacks, countermeasures, and future vision. *Future Generation Computer Systems* 136:358–377. <https://doi.org/10.1016/j.future.2022.06.013>
46. Negro-calduch E, Azzopardi-muscat N, Krishnamurthy RS, Novillo-ortiz D (2021) International Journal of Medical Informatics Technological progress in electronic health record system optimization: Systematic review of systematic literature reviews. *Int J Med Informatics* 152:104507. <https://doi.org/10.1016/j.ijmedinf.2021.104507>
47. Peischl B, Tazl OA, Wotawa F (2022) The Journal of Systems & Software Testing anticipatory systems: A systematic mapping study on the state of the art. *J Syst Softw* 192:111387. <https://doi.org/10.1016/j.jss.2022.111387>
48. Rejeb A, Rejeb K, Simske S, Treiblmaier H (2022) Internet of Things The big picture on the Internet of things and the smart city: a review of what we know and what we need to know. *Internet of Things* 19(July):100565. <https://doi.org/10.1016/j.iot.2022.100565>
49. Alekseeva D, Ometov A, Arponen O, Simona E (2022) The future of computing paradigms for medical and emergency applications. *Comput Sci Rev* 45:100494. <https://doi.org/10.1016/j.cosrev.2022.100494>
50. Onyema EM, Dalal S, Romero CAT, Seth B, Young P, Wajid MA (2022) Design of Intrusion Detection System based on Cyborg intelligence for security of Cloud Network Traffic of Smart Cities. *J Cloud Comput* 11(1):1–20
51. Chuah SH, Yu J (2021) Journal of Retailing and Consumer Services The future of service: The power of emotion in human-robot interaction. *J Retailing Consumer Serv* 61(March):102551. <https://doi.org/10.1016/j.jretconser.2021.102551>
52. Agirre JA, Etxeberria L, Barbosa R, Basagiannis S, Giantamidis G, Bauer T (2021) Microprocessors and Microsystems The VALU3S ECSEL project: Verification and Validation of automated systems safety and security. *Microprocess Microsyst* 87:104349. <https://doi.org/10.1016/j.micpro.2021.104349>
53. Kathiravelu P, Zaiman Z, Gichoya J, Veiga L, Banerjee I (2022) Towards an internet-scale overlay network for latency-aware decentralized workflows at the edge. *Comput Netw* 203
54. Latif SA, Wen FBX, Iwendi C, Li-li FW, Mohsin SM, Han Z, Band SS (2022) AI-empowered, blockchain and SDN integrated security architecture for IoT network of cyber physical systems. *Comput Commun* 181:274–283
55. Ramesh TR, Lilhore UK, Poongodi M, Simaiya S, Kaur A, Hamdi M (2022) Predictive analysis of heart diseases with machine learning approaches. *Malaysian J Comput Sci* 31:132–148
56. Iwendi C et al (2020) "KeySplitWatermark: Zero Watermarking Algorithm for Software Protection Against Cyber-Attacks," *IEEE Access* 8:72650–72660. doi: <https://doi.org/10.1109/ACCESS.2020.2988160>
57. Dhiman P, Kukreja V, Manoharan P, Kaur A, Kamruzzaman MM, Dhaou IB, Iwendi C (2022) A Novel Deep Learning Model for Detection of Severity Level of the Disease in Citrus Fruits. *Electronics* 11(3):495
58. Manoharan P, Walia R, Iwendi C, Ahanger TA, Suganthi ST, Kamruzzaman MM, Hamdi M (2022) SVM-based generative adversarial networks for federated learning and edge computing attack model and outpoising. *Expert Syst* 9
59. Poongodi M, Malviya M, Hamdi M, Vijayakumar V, Mohammed MA, Rauf HT, Al-Dhlan KA (2022) 5G based Blockchain network for authentic and ethical keyword search engine. *IET Commun* 16(5):442–448

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen® journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)