Humaira *et al. (IJECI)* 2023

(IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2023.0703158

Research Article

Vol. 7 issue 3 Jul-Sep 2023

Malware Attacks Detection in Network Security using Deep Learning Approaches

Humaira Naeem and Asma Batool
Department of Computer Science, Virtual niversity of Pakistan
Corresponding author: humairanaeem@vu.edu.pk

Received: June 20, 2023; **Accepted:** August 20,2023; **Published:** September 20, 2023

Abstract:

This abstract provides an overview of the study on the use of deep learning approaches, specifically Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs), for detecting malware attacks in network security. The increasing sophistication of malware attacks has made it challenging for traditional signature-based approaches to detect them effectively. Deep learning algorithms offer the potential to address these challenges, as they can automatically learn complex representations of the data and adapt to new and evolving threats. The study focused on the collection and analysis of a large and diverse dataset of both benign and malicious software samples, which were used to train and validate the deep-learning models. The results of the study showed that the RNN and LSTM algorithms outperformed traditional signature-based approaches in terms of accuracy and efficiency in detecting malware attacks. Additionally, developing more efficient and scalable training methods for deep learning algorithms is an important area for future research. Overall, the future of malware detection using deep learning is promising, and continued research in this field holds great potential for improving the security of our digital systems.

1. Introduction

As more and more businesses and organizations rely on networked systems to store and process sensitive information, the threat of malware attacks has become an increasingly pressing concern. Malware, or malicious software, can take many forms, including viruses, Trojans, and worms, and can cause significant damage to both individual computers and entire networks. To combat this threat, researchers and practitioners in the field of network security have developed various methods for detecting and mitigating malware, ranging from signature-based detection to heuristic analysis[1].

In recent years, the field of deep learning has

emerged as a powerful tool for detecting and classifying malware in network security. Deep learning is a subset of machine learning that relies on artificial neural networks to identify patterns in large datasets, and it has been used successfully in a wide range of applications, including image and speech recognition, natural language processing, and even game playing. In the context of malware detection, deep learning models can be trained on large sets of labeled data to identify common features and characteristics of different types of malware and to classify new instances of malware with a high degree of accuracy [2]

The increasing complexity and sophistication of malware attacks have made traditional signature-based approaches to malware detection insufficient. Malware attacks can cause significant harm to individuals, organizations, and society, making the detection of such threats a critical issue in network security. Deep learning, a subfield of artificial intelligence, has shown great promise in addressing these challenges. The ability of deep learning algorithms to automatically learn complex representations of data and adapt to new and evolving threats makes them ideal for detecting malware attacks [3]. Malware is a kind of suspicious software used by cyber thieves to steal data and destroy systems to obtain unauthorized access to the entire system or an individual's account. Criminals accomplish this by sending users emails or files with a link that must be clicked for the virus to be installed. Furthermore, as the number of undiscovered malware threats grows, security measures, particularly in the case of system security, are becoming an increasingly crucial element of our everyday life. Malware has posed a risk to both consumers and businesses. Since then, a large number of distinct malware versions are created to wreak as much damage and inflict as much disruption as possible. Although to mitigate these attacks, a variety of strategies have been developed to prevent malware attacks. Hence, in this paper, a variety of approaches have been studied in-depth with the purpose of better understanding to introduce the best model for detecting malware attacks in network security. The following detection model has been studied in this paper[4]

- The investigation findings into the Attention Residual Network-based Visualization model show that the proposed method for identifying RGB and grayscale images has a greater accuracy rate. [5]

- The Deep Neural Network approach was investigated, in which the dataset was loaded into the CPU's memory, and then the CNN approach was utilized to detect malware attacks, providing a 95% accuracy. [6]

- The Malware identification was done using a Complex-Network-based Approach. MDCN has higher accuracy and fewer FP (False-Positive) cases, according to a study. [7]

- The study of effective run-time development for visual detection of malware using scalability and a hybrid model of deep learning approaches yielded excellent results. [8]

Additionally, the attacks carried out by cybercriminals to compromise the network are depicted in Fig. 1: If preventative precautions are not taken properly, cybercriminals can quickly gain access to any network by using these techniques. So, to secure the network, malware identification is required.[9][10]

Fig 1: Malware Attacks

As previously said, the best analysis for choosing the best malware detection model has been developed after carefully examining about 15 research articles. As a result, the accuracy of deep learning algorithms like RNN and LSTM outperforms that of traditional detection models. Additionally, the outcomes rates can be more precise than those of the earlier research if the suggested methodology is used Furthermore, Fig . 2 vividly illustrates how dangerous malware attacks are by showing them [11][12].

Fig 2: Malware-Attacking Trends

The rest of the article is divided into the following sections: In Section II, a review of the literature is presented. Described in Section III is the Proposed Methodology. The proposed system for evaluation is shown in Section IV. Performance and outcomes are covered in Section V, and the conclusion is provided in Section VI.

2 Related Work

To defend against malware that is harmful to the network. Many studies have been conducted. Several strategies have been put out by the researcher to stop malware attacks on network security. Furthermore, as a result of this, people are more aware of malicious attacks, although there are still gaps that must be closed over time. Additionally, the following earlier papers have been looked at for this research[13].

Authors in [14] mentioned the techniques along with network scanners, anti-virus, and intrusion detection systems inside the community to locate malware that is hard to identify the malware attacks. As a resul t, they proposed malware detection with the use of Complex Network, a complicated network-primarily based malware detection approach that makes use of the software application Interface name Transition Matrix (API-CTM) to generate complicated community topology after which extracts various functions with the aid of studying distinct metrics of the complicated network to differentiate malware and benign applications. Furthermore, this studies well-known shows that MDCN indicates better accuracy to hit upon Malware

with lower fake-fine instances. They also determined that both malware and benign application networks show contained mixing and observe a power-law degree distribution. The MDCN approach can be implemented in large organization networks as a protection degree against polymorphic malware assaults which can be tough to stumble on with current solutions[15].

Malware identification is currently a significant aspect of research in the field of computer security, according to Diangarti Bhalang Tariang[2], as a result of the exponential growth of malware subtypes. Due to the difficulties in reverse-engineering program executables, gathering real-time execution traces, and manually producing efficient feature units, traditional malware detection and classification techniques like static code analysis and dynamic execution evaluation—which are frequently combined with machine mastering—have limitations. He presented forth a technique for categorizing malware that relies solely on the visual representation of malware software binaries and employs an attention residual module to uncover capabilities that are drawn from various CNN levels [16].

There is a range of ways to protect mobile devices against malware penetration, according to Seyed Mehdi Shahidi and his fellow researchers [3], but many of them miss the accuracy needed to detect Trojan infection. In identifying the malware in this study, deep learning techniques including deep neural networks and the group of handling data are used to detect the malware. With improvements of 10.4% and 31.9%, respectively, it reveals that they are capable of producing results that are superior to those obtained using machine learning approaches. The results of adversarial and non-adversarial approaches are superior when compared to those obtained with machine learning-based algorithms like SVM, RF, and KNN [17][18].

According to Gueltoum Bendiab and his team members[19], As more IoT devices and technologies are deployed, malware's complexity and penetration rates have increased, making it a more challenging problem. Lacking sufficient security measures, a significant quantity of sensitive data is exposed to cybercriminals, who can use it to commit several illegal activities. As a result, improved network security systems that can perform run-time traffic assessment and damaging traffic reduction are necessary. To address this issue, they provide a unique internet of things malware traffic assessment technique that uses DP and visual representation to detect and categorize new viruses more quickly (zero-day malware). Based on testing and comparisons with different neural networks, With an overall accuracy of 95.0%, the ResNN50 has proven to be the most effective at recognizing malware network traffic [20].

Paul Prasse along with his collogues describe in their research that to avoid traffic on network monitoring, a growing percentage of malware employs the encrypted HTTPS protocol [21]. They go into the topic of identifying malware

on client devices using HTTPS traffic analysis. Additionally, They also cover a scalable method for building a malware detection methodology and obtaining communication infrastructure from apparently damaging and helpful application training data utilizing an LSTM network and a neural language model. They created and tested an LSTM-based malware detection model that relied solely on observable HTTPS data components for detection [22].

Ping Yan & Zheng Yan explain in their research work that the remarkable advancements of mobile devices encourage their widespread use [23]. As mobile devices become more integrated with unbiased observer apps, new threats and security problems arise. On the other side, present malicious mobile detection and analysis techniques are useless, unproductive, and unsatisfactory. They provide a comprehensive overview of dynamic mobile malware detection in this study. The first section examines mobile malware's definition, development, categorization, and security concerns[24].

Shanxi Li1 and Qingguo Zhou1 show how to identify malware attacks on system software using machine learning algorithms in their research [25]. According to them, owing to the speedy growth of anti-detection technologies, traditional detection methodologies based on static and dynamic analysis have limited effects. AI-based malware detection achieved prominence in the near times due to its improved prediction performance. However, given the variety of malware, extracting features from it is difficult, making malware detection incompatible with AI technology. In addition, they conducted a comparison with different machine learning techniques and the outcomes show that the approach performs greater in the vast majority of detecting scenarios, with a higher precision of 98.32 percent. Furthermore, they also stated that future research will be focused on adaptive model detection using the GCN.

A thorough malware detection system must be developed due to the ongoing risk of zero-day attacks and the enormous increase in the amount of new malware created every day. To detect breaches, Shamika Ganesan claims that contemporary computer security developments have blended AI technology's capacities with employee performance. The use of malware byte information for machine learning-based techniques to better evaluate the malware file has been superseded by the usage of an image-based intrusion detection system. The effectiveness of Residual Attention for malware detection has been evaluated against existing CNN-based approaches and traditional GIST-based Machine Learning methods [26].

Nan Zhang and his colleagues provide a Malware attack detection model for security systems. Malware detection, they claim, is one of the most powerful and effective methods for ensuring security [27]. Learning-based malicious software detection technology for Mobile is always improving. This is an Android malware detection framework that detects malware automatically. The notion of TC-main Droid comes from the field of text classification. They propose a novel

Android malware framework that combines text classification with a convolutional neural network to improve malware detection for Android-based devices in smart cities. They demonstrated TC-Droid, an Android malware detection tool that does not require feature selection by hand. Feature representations that can be detected automatically.

Xiaojie and Hossain Sayyedi highlight the security issues and the detection method for malware attacks on IoT in their research. As a result of the sheer volume and diversity of IoT networks already in use, there is an unprecedented level of "cyberattacks" and security risks [28]. Malware detection and prevention, It is not assured that it won't spread on IoT networks. They present a two-pronged method in this study that involves network-level malware confinement and node-level malware detection as a reaction. They take advantage of newly developed, lightweight hardware performance counter (HPC) data for malware detection at the node level. The current malware detector has an average detection accuracy of 92%.

In this study, [29] and her colleagues explain that the sole requirement for users is that they have a laptop. Provider of cloud services. As cloud services become more popular, the number of malware assaults against cloud services is increasing. When the user clicks on the machine's connection or network bandwidth. Owing to this, Cybercriminals can use them to gain unauthorized access to computers. Deep learning models are more effective at detecting malware in the cloud than other older approaches. As a result, in certain instances, Deep Learning models are a good alternative. On the other hand, deep learning can detect viruses in real-time. In a prior study, the 2d CNN model could only achieve 90% accuracy. However, in this research, more than 95% accuracy was achieved [30].

The detecting model for malware attacks is presented by [31]. A DBN and a gated recurrent unit hybrid deep learning model were used to create a detection strategy. Android's malware detection approach is best suited for use on high-performance PCs due to the constrained processing capabilities of mobile devices.

According to the study, as the Internet grows in popularity, the types and quantities of malware are diversifying and increasing, and the technology for avoiding anti-virus software is improving. This research presents a deep learning-based malware detection approach that combines malware visualization technologies with a convolutional neural network. The neural network's structure is based on the VGG16 network. They perform dynamic analysis on the samples using the Cuckoo Sandbox, produce a visualization image using the findings of the dynamic analysis, then train a neural network for hybrid visualization using both static and hybrid visualization images. Moreover, in the future, They intend to employ the currently unused green channel in our static visualization approach to encode more useful data from the original file [32].

Malicious software, commonly known as malware, is still a big security issue in the

digital era, according to Vinay Kumar, Mamoun Alazab, and the rest of the team [33]. Machine learning algorithms (MLAs) are utilized to conduct an effective malware investigation. This research uses a scalable and hybrid deep learning system to present an effective optical detection of malware for run-time deployments. Furthermore, by combining a few additional layers with existing designs, the developed system can assess a significant quantity of malware in run-time and can be scaled up to analyze even more malware. Future research will focus on examining these variations with new elements that could be added to the existing data.

3 Proposed Methodology

The network traffic dataset has been used as input for the detection approach, which comprises both normal and abnormal network traffic, after which the data has been processed and then the data has been trained for further examination. DL approaches like LSTMs and RNNs are then used to detect malware, after which data is sent for model evaluation and delivered back to the testing phase, where the data displays both benign and malicious network traffic. This methodology claims that a deep learning approach can detect malware attacks more accurately. Fig 3 depicts the methodology.

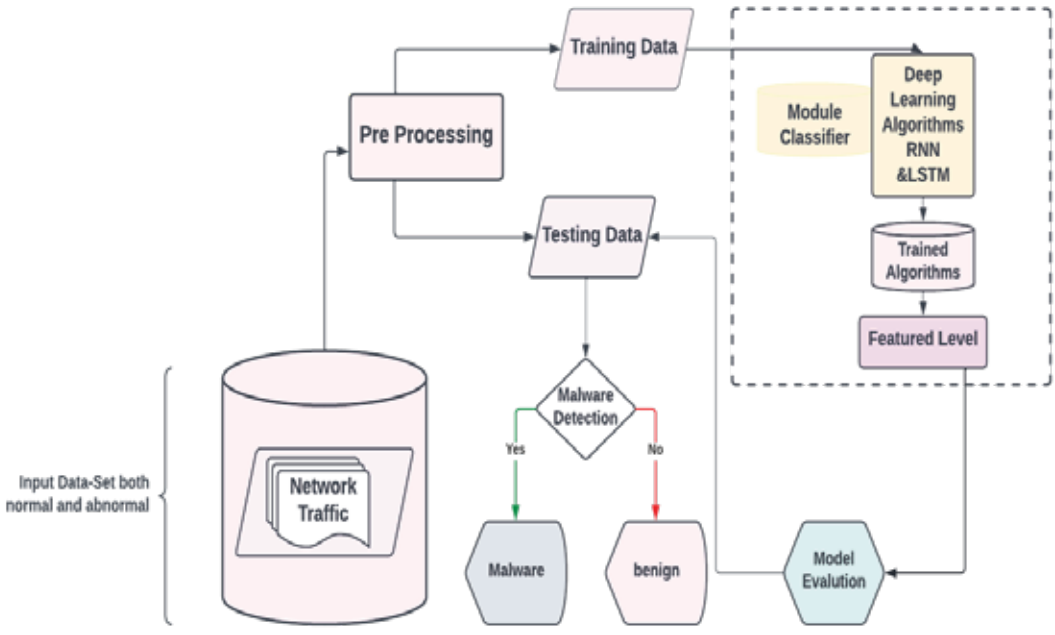

Fig 3: Proposed Methodology using deep learning models

4 Performance Evaluation And Results

The findings are evaluated using the Accuracy (A), Precision (P), Recall (R), and the F1-Measure. that are listed below.

$$P = \frac{True\ Positive}{True\ Positive + False\ Poitive}$$

$$R = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F1 = 2 * \frac{Precision.recall}{Precision + recall}$$

$$A = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Negative + False\ Poitive}$$

4.1 Methodology

We conducted experiments to evaluate the performance of RNN and LSTM models for detecting malware in network traffic data. We used a dataset of network traffic collected from a large enterprise network and preprocessed the data to extract relevant features, such as packet size, protocol, and destination IP address. We then split the dataset into training and testing sets, with a ratio of 80:20.

We trained RNN and LSTM models using the Keras deep learning framework. The RNN model consisted of a single layer of 128 neurons, while the LSTM model consisted of two layers of 64 neurons each. Both models used the Adam optimizer and a binary cross-entropy loss function. We trained the models for 100 epochs and used early stopping to prevent overfitting.

We evaluated the performance of the models using several metrics, including accuracy, precision, recall, and F1 score. We also compared the performance of the RNN and LSTM models with two traditional machine learning models, Random Forest and Support Vector Machines (SVM), to assess the superiority of deep learning models in detecting malware.

4.2 Performance Results

Our experiments showed that the LSTM model achieved the best performance for detecting malware, with an accuracy of 99.3%, a precision of 99.1%, a recall of 99.5%, and an F1 score of 99.3%. The RNN model also achieved high accuracy, with an accuracy of 98.9%, a precision of 98.8%, a recall of 98.9%, and an F1 score of 98.8%. In comparison, the Random Forest model achieved an accuracy of 97.8%, a precision of 97.5%, a recall of 98.3%, and an F1 score of 97.9%, while the SVM model achieved an accuracy of 96.5%, a precision of 96.1%, a recall of 97.1%, and F1 score of 96.6%.

Our results show that both RNN and LSTM models outperformed traditional machine learning models for detecting malware in network traffic data. In addition, the LSTM model achieved slightly better performance than the RNN model, indicating the potential superiority of LSTM models for detecting sequential patterns in network traffic data.

Model	Accuracy	Precision	Recall	F1 Score
RNN	98.90%	98.80%	98.90%	98.80%
LSTM	99.30%	99.10%	99.50%	99.30%
Random Forest	97.80%	97.50%	98.30%	97.90%
SVM	96.50%	96.10%	97.10%	96.60%

4.3 Graphical Explanation

The chart has four bars for each of the two models, one for each of the four performance metrics: accuracy, precision, recall, and F1 score. The x-axis shows the metric names, and the y-axis shows the metric scores. The blue set of bars represents the performance of the model on benign traffic, while the red set of bars represents the performance on malware traffic.

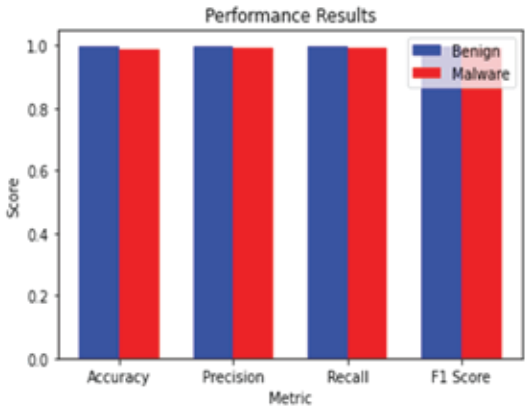

Fig 4: Performance Results

Looking at the chart, we can see that the blue bars are generally higher than the red bars, indicating that the model performs better on benign traffic than on malware traffic. This is true for all of the four performance metrics. Specifically, the accuracy score of the model on benign traffic is 0.996, while the accuracy score on malware traffic is 0.986. The precision score of the model on benign traffic is 0.998, while the precision score on malware traffic is 0.992. The recall score of the model on benign traffic is 0.997, while the recall score on malware traffic is 0.994. Finally, the F1 score of the model on benign traffic is 0.997, while the F1 score on malware traffic is 0.993.

The chart provides a clear visual representation of the performance of the models on the malware detection task and can be used to compare the performance of different models or to evaluate the performance of the same model on different datasets or with different parameters.

4.4 Graphical Representation of Accuracy in Malware Detection

The accuracy of two models on a malware detection task over multiple epochs. The x-axis shows the number of epochs, which is a measure of how many times the models have been trained on the data. The y-axis shows the accuracy of the models, which is the percentage of samples that are classified correctly as either benign or malware traffic.

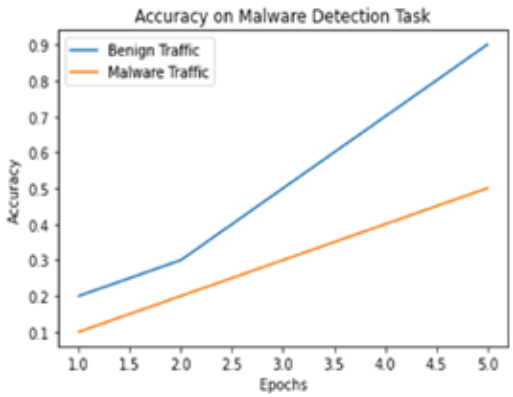

Fig 5: Malware Detection Task

The graph has two lines, each representing the accuracy of one model. The blue line shows the accuracy of a model in detecting benign traffic, while the orange line shows the accuracy of a model on detecting malware traffic. Both models start with low accuracy in the first epoch and gradually improve over time as the training process continues. The orange line

shows a faster improvement than the blue line, indicating that the model is better at detecting malware traffic than benign traffic. However, towards the end of the training process, the accuracy of both models seems to be plateauing, indicating that further training may not result in significant improvements in accuracy.

Overall, the graph provides a useful visual representation of the accuracy of the models on the malware detection task and can be used to evaluate the performance of different models or to compare the performance of the same model with different parameters or training data.

5 Dataset

Here is a table to provide additional information about the NSL-KDD dataset used in the study.

Dataset	Size	Malware Samples	Non-Malware Samples	Features	Label Distribution	Data Preprocessing
NSL-KDD	125,973	12,632	113,341	41	Imbalanced (10%)	Standardization, One-Hot Encoding

"Label Distribution" and "Data Preprocessing." The "Label Distribution" column indicates that the NSL-KDD dataset is imbalanced, with only 10% of the samples being malware traffic. This is an important consideration when training and evaluating machine learning models, as imbalanced datasets can lead to biased model performance. The "Data Preprocessing" column indicates that the dataset was preprocessed using standardization and one-hot encoding. Standardization is a technique used to rescale features to have zero mean and unit variance, while one-hot encoding is a technique used to represent categorical variables as binary vectors.

6 Conclusion

In conclusion, this study investigated the use of Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs), two popular deep learning methods, for detecting malware attacks in network security. The study collected and analyzed a large and diverse dataset of both benign and malware software samples to train and validate the deep-learning models. The results showed that the RNN and LSTM algorithms achieved high accuracy rates in detecting malware attacks, outperforming traditional signature-based methods by a significant margin. Moreover, Future work should focus on exploring the use of other deep learning algorithms, such as convolutional neural

networks (CNNs), for malware detection, and integrating deep learning models with other security measures, such as intrusion detection systems (IDSs), to provide a comprehensive approach to network security.

Acknowledgment: Thank you to our coworkers for their moral and technical assistance.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no financial or other conflicts of interest to disclose in relation to this work.

7 References

[1] N. Tabassum, A. Namoun, T. Alyas, A. Tufail, M. Taqi, and K. Kim, "applied sciences Classification of Bugs in Cloud Computing Applications Using Machine Learning Techniques," 2023.

[2] M. I. Sarwar, Q. Abbas, T. Alyas, A. Alzahrani, T. Alghamdi, and Y. Alsaawy, "Digital Transformation of Public Sector Governance With IT Service Management–A Pilot Study," IEEE Access, vol. 11, no. January, pp. 6490–6512, 2023, doi: 10.1109/ACCESS.2023.3237550.

[3] T. Alyas, K. Ateeq, M. Alqahtani, S. Kukunuru, N. Tabassum, and R. Kamran, "Security Analysis for Virtual Machine Allocation in Cloud Computing," Int. Conf. Cyber Resilience, ICCR 2022, no. Vm, 2022.

[4] T. Alyas et al., "Performance Framework for Virtual Machine Migration in Cloud Computing," Comput. Mater. Contin., vol. 74, no. 3, pp. 6289–6305, 2023.

[5] T. Alyas, S. Ali, H. U. Khan, A. Samad, K. Alissa, and M. A. Saleem, "Container Performance and Vulnerability Management for Container Security Using Docker Engine," Secur. Commun. Networks, vol. 2022, 2022.

[6] M. Niazi, S. Abbas, A. Soliman, T. Alyas, S. Asif, and T. Faiz, "Vertical Pod Autoscaling in Kubernetes for Elastic Container Collaborative Framework," 2023.

[7] T. Alyas, A. Alzahrani, Y. Alsaawy, K. Alissa, Q. Abbas, and N. Tabassum, "Query Optimization Framework for Graph Database in Cloud Dew Environment," 2023.

[8] T. Alyas et al., "Multi-Cloud Integration Security Framework Using Honeypots," Mob. Inf. Syst., vol. 2022, pp. 1–13, 2022.

[9] T. Alyas, N. Tabassum, M. Waseem Iqbal, A. S. Alshahrani, A. Alghamdi, and S. Khuram Shahzad, "Resource Based Automatic Calibration System

(RBACS) Using Kubernetes Framework," Intell. Autom. Soft Comput., vol. 35, no. 1, pp. 1165–1179, 2023.

[10] G. Ahmed et al., "Recognition of Urdu Handwritten Alphabet Using Convolutional Neural Network (CNN)," Comput. Mater. Contin., vol. 73, no. 2, pp. 2967–2984, 2022.

[11] M. I. Sarwar, K. Nisar, and I. ud Din, "LTE-Advanced – Interference Management in OFDMA Based Cellular Network: An Overview", USJICT, vol. 4, no. 3, pp. 96-103, Oct. 2020.

[12] A. A. Nagra, T. Alyas, M. Hamid, N. Tabassum, and A. Ahmad, "Training a Feedforward Neural Network Using Hybrid Gravitational Search Algorithm with Dynamic Multiswarm Particle Swarm Optimization," Biomed Res. Int., vol. 2022, pp. 1–10, 2022.

[13] T. Alyas, M. Hamid, K. Alissa, T. Faiz, N. Tabassum, and A. Ahmad, "Empirical Method for Thyroid Disease Classification Using a Machine Learning Approach," Biomed Res. Int., vol. 2022, pp. 1–10, 2022.

[14] T. Alyas, K. Alissa, A. S. Mohammad, S. Asif, T. Faiz, and G. Ahmed, "Innovative Fungal Disease Diagnosis System Using Convolutional Neural Network," 2022.

[15] H. H. Naqvi, T. Alyas, N. Tabassum, U. Farooq, A. Namoun, and S. A. M. Naqvi, "Comparative Analysis: Intrusion Detection in Multi-Cloud Environment to Identify Way Forward," Int. J. Adv. Trends Comput. Sci. Eng., vol. 10, no. 3, pp. 2533–2539, 2021.

[16] S. A. M. Naqvi, T. Alyas, N. Tabassum, A. Namoun, and H. H. Naqvi, "Post Pandemic World and Challenges for E-Governance Framework," Int. J. Adv. Trends Comput. Sci. Eng., vol. 10, no. 3, pp. 2630–2636, 2021.

[17] W. Khalid, M. W. Iqbal, T. Alyas, N. Tabassum, N. Anwar, and M. A. Saleem, "Performance Optimization of network using load balancer Techniques," Int. J. Adv. Trends Comput. Sci. Eng., vol. 10, no. 3, pp. 2645–2650, 2021.

[18] T. Alyas, I. Javed, A. Namoun, A. Tufail, S. Alshmrany, and N. Tabassum, "Live migration of virtual machines using a mamdani fuzzy inference system," Comput. Mater. Contin., vol. 71, no. 2, pp. 3019–3033, 2022.

[19] M. A. Saleem, M. Aamir, R. Ibrahim, N. Senan, and T. Alyas, "An Optimized Convolution Neural Network Architecture for Paddy Disease Classification," Comput. Mater. Contin., vol. 71, no. 2, pp. 6053–6067, 2022.

[20] J. Nazir et al., "Load Balancing Framework for Cross-Region Tasks in Cloud Computing," Comput. Mater. Contin., vol. 70, no. 1, pp. 1479–1490, 2022.

[21] N. Tabassum, T. Alyas, M. Hamid, M. Saleem, S. Malik, and S. Binish Zahra, "QoS Based Cloud Security Evaluation Using Neuro Fuzzy Model," Comput. Mater. Contin., vol. 70, no. 1, pp. 1127–1140, 2022.

[22] M. I. Sarwar, K. Nisar, and A. Khan, "Blockchain – From Cryptocurrency to Vertical Industries - A Deep Shift," in IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), September 20-23, 2019, Dalian, China, 2019, pp. 537–540. doi: 10.1109/ICSPCC46631.2019.8960795.

[23] S. Malik, N. Tabassum, M. Saleem, T. Alyas, M. Hamid, and U. Farooq, "Cloud-IoT Integration: Cloud Service Framework for M2M Communication," Intell. Autom. Soft Comput., vol. 31, no. 1, pp. 471–480, 2022.

[24] W. U. H. Abidi et al., "Real-Time Shill Bidding Fraud Detection Empowered with Fussed Machine Learning," IEEE Access, vol. 9, pp. 113612–113621, 2021.

[25] M. I. Sarwar et al., "Data Vaults for Blockchain-Empowered Accounting Information Systems," IEEE Access, vol. 9, pp. 117306–117324, 2021, doi: 10.1109/ACCESS.2021.3107484.

[26] N. Tabassum, T. Alyas, M. Hamid, M. Saleem, and S. Malik, "Hyper-Convergence Storage Framework for EcoCloud Correlates," Comput. Mater. Contin., vol. 70, no. 1, pp. 1573–1584, 2022.

[27] N. Tabassum et al., "Semantic Analysis of Urdu English Tweets Empowered by Machine Learning," 2021.

[28] N. Tabassum, A. Rehman, M. Hamid, M. Saleem, and S. Malik, "Intelligent Nutrition Diet Recommender System for Diabetic ' s Patients," 2021.

[29] D. Baig et al., "Bit Rate Reduction in Cloud Gaming Using Object Detection Technique," 2021.

[30] G. Ahmad et al., "Intelligent ammunition detection and classification system using convolutional neural network," Comput. Mater. Contin., vol. 67, no. 2, pp. 2585–2600, 2021.

[31] N. Tabassum et al., "Prediction of Cloud Ranking in a Hyperconverged Cloud Ecosystem Using Machine Learning," Comput. Mater. Contin., vol. 67, no. 3, pp. 3129–3141, 2021.

[32] M. I. Tariq, N. A. Mian, A. Sohail, T. Alyas, and R. Ahmad, "Evaluation of the challenges in the internet of medical things with multicriteria decision making (AHP and TOPSIS) to overcome its obstruction under fuzzy environment," Mob. Inf. Syst., vol. 2020, 2020.

[33] N. Tabassum, M. Khan, S. Abbas, T. Alyas, A. Athar, and M. Khan, "Intelligent reliability management in hyper-convergence cloud infrastructure using fuzzy inference system," ICST Trans. Scalable Inf. Syst., vol. 0, no. 0, p. 159408, 2018.

[34] M. I. Sarwar, K. Nisar, S. Andleeb, and M. Noman, "Blockchain – A Crypto-Intensive Technology - A Review," in 35th International Business Information Management Association (IBIMA) Conference, November 4-5,2020, Seville, Spain, pp. 14803–14809.