Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 5s (2024)

A Hybrid Malware Detection System for Enhanced Cloud Security Utilizing Trust-Based Glow-Worm Swarm Optimization and Recurrent Deep Neural Networks

R Swathi[1], Sivakumar Depuru[2], M. Sakthivel[3], S. Sivanantham[4], K Amala[5], Pavan Kumar Ande[6]

[1]Department of CSA, Sri Venkateswara College of Engineering Tirupati Andhra Pradesh, India, Email: swathi.mani08@gmail.com

[2]Department of CSE, School of Computing, Mohan Babu University, Tirupati, AP, india Email: sivakumar.d@vidyanikethan.edu

[3]Dept. of. AI & DS, Sri Shanmugha College of Engineering and Technology, Sankari, Salem, sakthisalem@gmail.com

[4]Dept. of CSE, Saveetha School of Engineering, SIMATS, Chennai, sivananthams.sse@saveetha.com

[5]Department of ECE, Sri Venkateswara College of Engineering (Autonomous), Tirupati, Email: amala.k@svcolleges.edu.in

[6]Dept. of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, 522302, Andhra Pradesh, India, Email: apavankumar@kluniversity.in

Corresponding author: sivakumar.d@vidyanikethan.edu

Article History:

Received: *26-04-2024*

Revised: *05-06-2024*

Accepted: *24-06-2024*

Abstract:

User credentials are vulnerable to exposure in demilitarized zones due to software vulnerabilities and hardware threats. This research aims to mitigate these risks by proposing a sophisticated trust-based malware detection (T-MALWARE DETECTION) method that can accurately classify data. The proposed system utilizes an enhanced Glow-Worm Swarm Optimization (IGWSO) technique to efficiently cluster datasets. To classify potential intrusions and assign trust levels to cloud data after clustering, a Recurrent Neural Network (RNN) approach is employed. The effectiveness of the Trust-oriented Malware Detection System (T-MALWARE DETECTIONS) is evaluated using metrics such as detection rate, precision, recall, and F-measure. This system is developed using Java and the CloudSimulator (CloudSim) tool, allowing for a thorough evaluation of its performance in comparison to contemporary state-of-the-art systems.

Keywords: Malware Detection System; Enhanced Cloud Security.

1. Introduction

The term "cloud-based computing" describes the modern IT phenomena of transmitting data and computations from personal PCs to massive data centres. Computation is made possible in Cloud environments through the integration of data centres in multiple spaces and linked high-velocity networks [1]. A group of scattered PCs known as the cloud gives networked users access to computing resources and services whenever they need them. Supply organization strategies assist in managing IT resources when both cloud workers and users do administrative tasks. Infrastructure as a Service (IaaS) in cloud computing is subdivided into physical, virtual, and managerial layers. Clients can access virtualized services across these various levels. This model utilizes virtual computing resources to create and manage physical resources. Clients are allocated multiple virtual

computing resources according to their requirements, and the management of these virtual resources is conducted through virtual machine monitoring (VMM) [2].

Evidently, security has a high vulnerability rating among the various cloud services. Numerous cloud services are necessary in the public, private, and industrial areas. Significant changes are occurring in the need-ability spectrum for these services [17]. Protection and resilience are at the top of the key need ability index. These suggest that a cloud should be designed to consider targeted attacks on cloud infrastructure and should only react to such attacks. The standard operating procedures on relevant CC structures reduce defense response and efficacy against novel threats that are likely to arise in the dynamic distributed computing environment. These special assaults clearly and unequivocally misuse the standard signature-based detection techniques [3]. Malware assaults have increased in frequency due to the massive quantity of information stored on cloud platforms. Malware in CC can be identified via either distributed or hypervisor-based techniques. Distributed detection techniques house the VMs agent within a guest virtual machine by default [18]. Anomalies are identified using a few well-known techniques using different denominators. The projected clustering mechanism does not show any dimension. The key piece of advice is to use a cluster to gather the input dataset and then classify the remaining data from the cluster center for characterizing the data that is already accessible and preventing intrusions [4]. The results matrix and visual notations show a significant rise in malware identification, yielding a precision of more than 92.45% [19, 29].

While the detection of malicious software initially appears to be a system's deterministic state, the rising eco system of the malevolent program is making the difficulty of recognizing it move toward a pseudo deterministic state. The tremendous difficulty presented by the systems that surround us and their rebellious neighbours in the kind of malware makes both stand out from one another. The industry's grand ambition is for a single supplier to provide a collection of algorithms for data analysis and resultant action [21, 31]. DoS attacks, which can be harmful to systems and their infrastructure if they happen simultaneously, are more likely to happen in the cloud, according to security experts. While many malware identification systems rely on machine learning techniques to identify breaches, their detection rates do not show significant improvements. A range of software bugs and computer hardware attacks expose user IDs or put them in danger of ending up in disarmed zones; an investigation is conducted, and lastly, a trust-oriented malware identification method based on the best classification of data is proposed [5]. The proposed T-MALWARE DETECTIONS is run in Java using the CloudSim tool, and it is shown to outperform existing state-of-the-art systems on the metrics of high recognition, precision rate, recall rate, as well as F-measures. The input dataset is divided into multiple groups using the IGWSO technique, and then clustering is employed to determine different levels of trust for cloud data that is used to classify a potential compromise [20, 30].

2. Literature Survey

In 2020, Rajendra Patil et al. [11, 22 32] extended their earlier study on vulnerability evaluation and patching by incorporating VM-oriented AMD systems for protecting susceptible virtual machines in the cloud. The proposed system is composed of two components: an anomaly detector for hypervisors and an agent for virtual machines. An ongoing agent looks for newly deployed

executables on virtual machines (VMs) and uses signature-based identification to locate known malware. In order to detect unforeseen attacks, it generated a profile with the optimal static attributes for a new executable. Two new fitness criteria are added to an upgraded binary bat technique to extract the best traits. The random forest technique was applied whenever the profile was given to the hypervisor in order to identify any anomalies. It classifies and generates an alert for the VM user's executable as either benign or malicious [29].

In 2021, Zahid Hussain Qaisar et al. [12] created a scalable CP-ABE-based multiagent system design to provide dependability in our proposed work and data sharing over publically available cloud storage. They recommended by means of a cloud host to act as a middleman between the end user and approved agents without endangering the system's security and confidentiality. They have also introduced a new method of protecting the cloud against malware by leveraging the highly effective influence of the state-of-the-art Gemini technique. Gemini was a useful tool for applying binary codes to detect commonalities in graph embeddings. The proposed inquiry provides a technique for cloud-oriented malware detection and addresses the challenges of efficiency and scalability. Three subjects were covered: malware detection power, scalability, and effectiveness with numerous agents.

A cloud environment-based intelligent behavior-based identification system was proposed by Omer Aslan et al. [13, 23, 33] in 2021. Using a variety of virtual machines, the proposed method first created a malware dataset that successfully recognizes distinctive characteristics. Then, in order to aid the learning-oriented and rule-based identification agents in differentiating between malware and benign samples, some features were added. A total of 10,000 program models have been analyzed in order to evaluate the efficacy of the proposed method. For known and unknown malware, the recommended solution showed a high accuracy and detection rate. In subsequent studies, they hope to apply the results of rule-based and learning-based identification. The results of a rule-based identification agent and a learning-based identification agent would be compared by a behavior-based identification agent [24][34]. Every time there were any deviations, the detection process would be repeated for those samples in an effort to reduce the rate of misclassification. Once both detecting agents have gathered identical categorization data, the client will receive the results [28].

Jian Zhang et al. [14] used MFA methods in 2021 in combination with VMI to collect a range of dynamic data from the hypervisor layer, hardware layers, and virtual machine storage. This study also proposed an adaptive feature selection method. Three diverse feature types are compared and analyzed from three angles by combining three different search techniques: security, system load, and efficiency. By changing the weight for each feature, it met the expected needs for malware detection in the cloud environment. By utilizing both the combined strategy of Voting and the AdaBoost ensemble learning approach, the identification method improves the overall classifier's detection and generalization accuracy [25][35]. By fusing the ML approach with practical features, they hope to bridge the semantic gap & attain the goal of VM introspection in this next study. The malware detection features that were already present in virtualized environments would be enhanced. They would look into more effective malware detection techniques.

Enes Sinan Parildi et al. [15] presented an alternative method for recognizing malware in 2021 that utilizes real-time assembly opcode sequences. They initially mix DL and NLP methods to facilitate

the extraction of more intricate behavioral characteristics from sequential opcode information. These qualities render this method resistant to code modification and effective against newly discovered malware. Finally, these features are given to several ML methods for categorization. Research conducted on a dataset of 26869 data that was dispersed more uniformly revealed that this approach might yield MCC scores as high as 0.95. In the future, visualization may employ the UMAP approach. Finally, the graph clearly shows that benign and malicious clusters were distinguished for a significant number of occurrences [27][37].

Farhan Ullah et al. [16] introduced a original feature representation technique for malware identification in 2022 that combines ACGs with byte-level picture modelling. At first, the Java source code and DEX file were extracted from the APK by reverse engineering. Second, in order to depict Android applications with sophisticated features, they generated ACGs by removing API calls and API series from the CFG. The ACGs may be able to follow Android apps' actions digitally. The multi-head attention-based TL method was then used to find the learnt features vector from ACGs. Once the DEX file had been converted into a malicious image, the textured properties were taken out and shown using a combination of FAST and BRIEF [26][36]. Ultimately, textural features and ACGs were combined for effective malware detection and categorization. In the end, mining the learnt features might make use of GloVe and Fast-text trained models. In addition, the efficacy of malware detection might be measured using more classy deep learning (DL) techniques, such as reinforcement learning (RL).

3. The Proposed Model

3.1 T-Malware Detection

CC has to keep track of the minimum amount of offers that the central controller receives in massively appropriated systems, affiliate sales gathering, and data centers. Systems necessitate the usage of appropriate mix structures, window assessment, and request priority, among other things. The traffic expertise gathering in the T-Malware Detection system is carried out by the systems of the Southern Security Interfaces, which facilitate communication between the controller and the Terminal Manager Switches/Make apparatuses. Cloud clients embrace virtual machines through a variety of actions for a given amount of time, duration, and connectivity regarding their virtual machines. The customer specifies each relationship request's amount, duration, recommended virtual machine center focus, required transmit cutoff, and allowed holding time [17].

Sequential character data transmission between the customer and the virtual machines they utilize will be accomplished through the usage of these cloud connection requests. In the cloud environment, there are many relationships between different cloud users that rely on the intended mix factor. T-MALWARE DETECTION is used to comprehend this matter. All things considered, the more basic self-definitive signals that are warped by language represent a feasible means of data transmission and storage. Open data is easily accessed through human verification. Rather of merely staring at someone directly, one might utilize their gaze to create signs. Furthermore, people are drawn to rapidly construct diverse action circumstances by selective social learning based on movement experiences, which preserves a vitally advantageous path from ineffective effort and error.

This way, you may deflect attention so that someone else can investigate more avenues for obtaining confirmation from the subject, who may be quite far away due to time and money constraints. The subject's autonomy in gaining credit at the time was acknowledged as discretionary activity within the framework of the viability and certainty of social learning. T-Malware Detection depends on user-provided data. The importance of social consciousness is acknowledged throughout. It uses just one element to operate in the model as well. The guaranteed T-Malware Detection structure is demonstrated using a hard distributing multi-star game strategy by typically multi-supervisor approach, which is thought to be for minimizing the risk of the incorrectly structured interaction.

The control traffic that flows through the server that the T-Malware Detection is checking requires every client in the network. This has an impact on the internal data server of the hierarchical control structure. T is produced in Eqn. (3.1) by considering the complete focus point and ignoring the customer's proximity in SDN.

$$\mathrm{T} = [\mathrm{v1} * (\mathrm{Dmn}/\mathrm{cmn})] + [\mathrm{v2} * (\mathrm{eavg}/\mathrm{Em})] + [\mathrm{v3} * (1/\mathrm{cmn})]$$

3.2 3.2 Enhanced IGWSO

The glow-worm, noted for its captivating luminescence and reproductive behaviors, serves as a compelling analogy for this optimization model. In this framework, the nests represent potential solutions, with the most viable nests, analogous to successful egg deposits, reflecting the best solutions. While it’s rare for certain species like the Guiro cuckoo to lay eggs in exposed nests—often leading to the destruction of other nests—this model focuses on optimizing resources rather than eliminating competitors. The integration of network virtualization design with the IGWSO algorithm aims to streamline scheduling processes. The following time-domain matrix ensures the inclusion of time-domain indicators to account for the volume of requests at different intervals.

3.3 Trust Computation via RDNN

Cloud computing (CC) offers dynamic flexibility in resource allocation, allowing for resource expiration before meeting distributed computing goals. In a distributed network, users openly communicate their resource requirements and projected task deadlines. Due to unforeseen user demands, task interruptions occur more frequently than anticipated. Users can choose virtual machines and bandwidth, and can also specify start times and data transmission parameters. To maximize the efficiency of resource distribution, constraints such as bandwidth (C1), nodes (C4), domain (C3), the ratio of virtual to physical links (C2), and binary conditions (C5) are utilized. The proposed Recurrent Deep Neural Network (RDNN) is designed to optimize these constraints. The evaluation of the RDNN begins with the initialization phase, where a random population of m initial points is created within the set (S). Each member of this population is evaluated using a specific equation, as outlined in Equation (3.2).

$$\mathrm{A}_i^t \leftarrow nt + \delta\,(mt - nt);\ t = 1, 2, \ldots, \quad (3.2)$$

Computation of functions min and max by Eq. (3.3) & Eq. (3.4)

$$\mathrm{f}_b = min\{f(\mathrm{A}i);\ i = 1, 2\ldots, n\} \quad (3.3)$$

$$\mathrm{f}_w = max\{f(\mathrm{A}i);\ i = 1, 2\ldots, n\} \quad (3.4)$$

If δ=0: A_t=n_t. No adjustment is made; A_t remains the same as n_t. If δ=1: A_t=m_t. Full adjustment is made; A_t immediately reaches m_t. For 0 < δ < 10 < 1: A_t is a weighted average of n_t and m_t, moving A_t partway towards m_t.

$$W = max\ (mt - nt);\ t = 1, 2\ldots, n \qquad (3.5)$$

The set W is empty, the population will move in a random manner. However, if W contains elements, a point that surpasses A_i in some respect should be chosen randomly. In cases where W is occupied, the process begins by modifying the attributes. Subsequently, a location within the optimal range is selected randomly. This involves choosing a file and making adjustments to A_i accordingly. Measurements are taken phase-by-phase at a given point, ensuring that the boundaries within the set and the permissible movement towards higher values are considered. The most recent point reached during this exploration is chosen for further progression through subsequent levels. This process is illustrated in Eq. (3.6).

$$A_i = \{ti;\ if\ f(tr) < f(xi)\ xi;\ otherwise \qquad (3.6)$$

Each user's strength is calculated after optimizing time-changing limits. It is shown in Eq. (3.7).

$$Ws = a1 + a2 + \cdots s = c1, c2\ldots, cn \qquad (3.7)$$

Outcome is shown in Eq. (3.8)

$$Trust = Max\ (Wa1, Wa2\ldots, Wsn) \qquad (3.8)$$

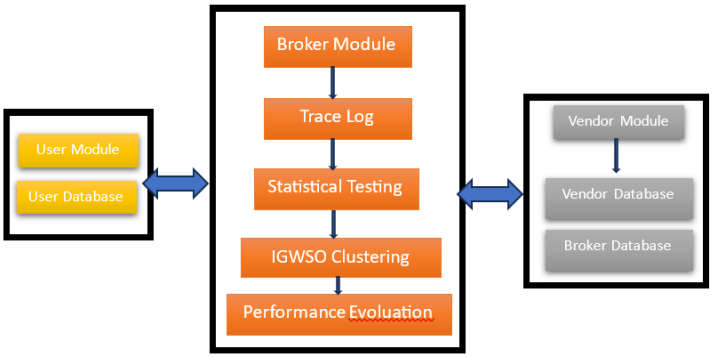


Figure 1 The Block diagram of suggested model

4. Result and Discussion

The efficacy of the proposed cloud-assisted malware file identification method is evaluated by CloudSim simulator based on multiple performance metrics. Malicious and valid files in the database are distinguished by using a DRNN classifier. The privacy constraint value k has a considerable impact on the overall number of position-based inquiries inside a specific least-obscured district, but the number of erroneous information supplied has a less significant impact. In order to mitigate the risk of flooding, it is necessary to periodically remove and handle this generated log record. The recommended approach Three execution metrics—precision, recall, and f-measure implementation—are used to evaluate performance [20]. A shroud device is used to test the procedure; the results are displayed below. The outcomes of the experiment are explained in the Table 4.1.

Training Size	Suggested Model		
	Recall	Precision	F-Measure
60	92.13	90.12	60.14
80	95.56	91.44	73.05
90	95.23	86.52	65.33

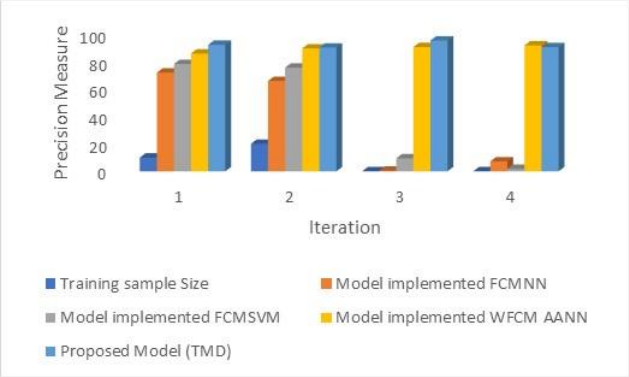
The overview has been evaluated using two metrics: the proportion of correctly recognized malware files as proximity (FP) and the fraction of properly recognized malware files as nonattendance (TP) for analysis. Unidentified malware file findings are also completing accuracy evaluation and f-measure, which gives frauds and FN a higher weight. The true classification of the detection in the context of the chaotic network, as well as the predicted classification, depend on the need for malware proximity as well as non-nearness. Traditional bifurcated representation estimates, including the TP, FP, FN, and TN functional labels, are utilized. The system's feasibility as suggested While FP is the number of realistic record transfers expected to be fraudulent, TP addresses the exact number of file transfers intended to be fraudulent extortion. The formation of FP can also be triggered when a significant proportion of the systems attain a peak prediction of TP. Accurate estimations are determined by using the formulas in the tables below.

5. Precision

This technique can be used to determine what proportion of the dataset's malware files were accurately classified and which were mistakenly classified as regular files. Table 4.2 provide an explanation of the comparison of precision values with the recommended and previous procedures. Several iterations were used to determine the accuracy level. By contrasting the algorithm's output with the success rate in identifying malicious files, one can evaluate a malware detection algorithm. More excellent instances were found in the training phase. In order to make the test and training sets larger, it was selected at random from the dataset.

Training sample Size	Model implemented			Proposed Model (TMD)
	FCMNN	FCMSVM	WFCM AANN	
10	71.89	78.15	85.78	92.15
20	65.78	75.48	89.45	90.11
0	0.45	9.45	90.45	95.23
0	7.18	1.78	91.68	90.4

The below figure provides an explanation of the comparison of precision values with the recommended and previous procedures. Several iterations were used to determine the accuracy level. By contrasting the algorithm's output with the success rate in identifying malicious files, one can evaluate a malware detection algorithm. More excellent instances were found in the training phase. In order to make the test and training sets larger, it was selected at random from the dataset.

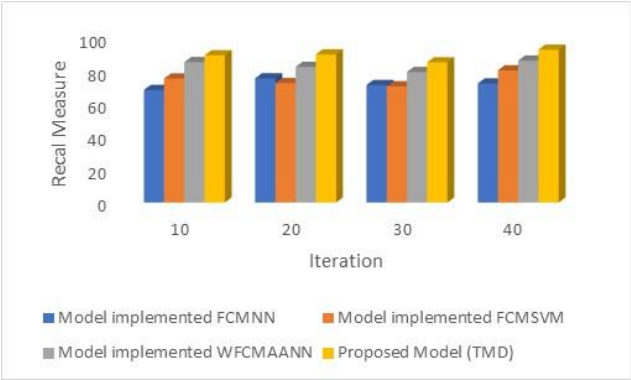


6. Recall

It calculates the overall percentage of malware files that have been accurately and inaccurately classified when compared to traditional files within datasets.

Training sample Size	Model implemented			Proposed Model (TMD)
	FCMNN	FCMSVM	WFCMAANN	
10	68.57	75.48	85.47	89.60
20	75.63	72.68	82.56	90.15
30	71.52	70.78	79.46	85.34
40	72.46	80.46	86.47	93.05

The figure compares conventional files from datasets and measures the percentage of accurately and inaccurately classified malware files.

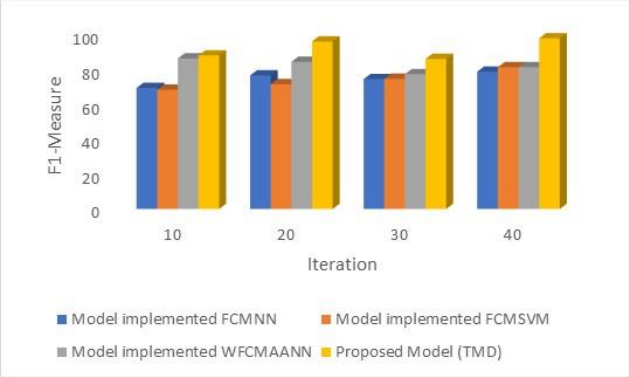


7. F1-Score

The F1 - measure is the most widely used weighted measure for precision and recall. The F1 measure takes into account both false positives and false negatives. Although it may not seem as intuitive as accuracy, it is often more valuable than precision, particularly in cases of excessive category allocation. When the costs of false positive and false negative readings are equal, then precision performs best.

Training sample Size	Model implemented			Proposed Model (TMD)
	FCMNN	FCMSVM	WFCMAANN	
10	69.56	68.48	86.47	88.19
20	76.74	71.68	84.56	96.24
30	74.58	74.78	77.46	86.20
40	78.89	81.46	81.47	98.02

The figure compares conventional files from datasets and measures F1 Score of accurately and inaccurately classified malware files.



The detection of malicious software is a difficult problem. Network engineers and IT managers face a challenging issue due to the vast and ever-growing ecosystem of malicious programs and technologies. Antivirus software is one of the most widely used techniques for detecting and blocking undesirable and hazardous programs. However, the complexity of today's malware increases, making it increasingly challenging for a single provider to produce signatures for every new threat. Malware is software created with the intention of covertly compromising or harming a computer system without the owner's awareness. Actually, the word "malware" encompasses all kinds of computer threats. A simple taxonomy of malware consists of file infectors and standalone malware. One may argue that many of the security issues surrounding CC are essentially the same as those that emerged during the Internet's early years. Despite this, the widespread use of virtualization—which is autonomous, service-oriented architecture, and utility computing—has sparked an evolution that has given rise to the detection of malware in cloud environments, or what is now commonly known as cloud computing (CC). The majority of end users have no idea what devices are, and they are no longer expected to understand, be meticulous about, or exert any type of control over the infrastructure that supports their computer activities. CC in general, its structure, and the detection techniques used to carry out each of the static evaluations and detection are the subjects of several related earlier studies: Heuristic for enhancing signature matching and identifying dynamic analysis. The optimal TMD system based on data classification has been suggested. It is suggested that the raw data set be clustered using the lGWSO method into many categories. The cloud data trust level classifies the information as intrusive or not using an RDNN. The simulation's result shows that the suggested TMD system outperforms the state-of-the-art systems in terms of high recall, F-measures, and identification precision.

References

[1] Ye, Y. F., Li, T., Adjeroh, D., & Iyengar, S. (2017). A survey on malware detection using data mining techniques. ACM Computing Surveys, 50(3), 40.

[2] Sasha Mahdavi Hezavehi and Rouhollah Rahmani, An anomaly-based framework for mitigating effects of DDoS attacks using a third party auditor in cloud computing environments, Cluster Computing, Vol. 23, 2020, pp. 2609–2627.

[3] J. V. Bibal Benifa and G. Venifa Mini, Modified Chebyshev polynomial-based access control mechanism for secured data access in cloud computing environment, Service Oriented Computing and Applications, Vol. 15, 2021, pp. 187–203.

[4] Heng He, Liang han Zheng, Peng Li, Li Deng, Li Huang and Xiang Chen, An efficient attribute based hierarchical data access control scheme in cloud computing, Human centric Computing and Information Sciences, Vol. 10,2020.

[5] K. Padmaja and R. Seshadri, A real-time secure medical device authentication for personal E-Healthcare services on cloud computing, International Journal of System Assurance Engineering and Management, 2021. 116

[6] Xianwei Gao, Changzhen Hu, Chun Shan, Baoxu Liu, Zequn Niu, Hui Xie, Malware classification for the cloud via semi supervised transfer learning, Journal of Information Security and Applications, vol 55, 2020.

[7] Chuanchang Liu, Jianyun Lu, Wendi Feng, Enbo Du, Luyang Di, Zhen Song,MobiPCR: Efficient, accurate, and strict ML-based mobile malware detection, Future Generation Computer Systems, vol 144, 2023.

[8] R. Aiyshwariya Devi and A.R. Arunachalam, Enhancement of IoT device security using an Improved Elliptic Curve Cryptography algorithm and malware detection utilizing deep LSTM, High-Confidence Computing, vol 3, 2023.

[9] Preeti Mishra, Ishita Verma, Saurabh Gupta, KVMInspector: KVM Based introspection approach to detect malware in cloud environment, Journal of Information Security and Applications, vol 51, 2020.

[10] Wentao Wei, Jie Wang, Zheng Yan, Wenxiu Ding, EPMDroid: Efficient and privacy preserving malware detection based on SGX through data fusion, Information Fusion, vol 82, 2022.

[11] Rajendra Patil, Harsha Dudeja, Chirag Modi, Designing in-VM-assisted lightweight agent-based malware detection framework for securing virtual machines in cloud computing, International Journal of Information Security, 2020, pp. 147–162.

[12] Zahid Hussain Qaisar, Sultan H. Almotiri, Mohammed A. Alghamdi, Arfan Ali Nagra, Ghulam Ali, "A Scalable and Efficient Multi-agent Architecture for Malware Protection in Data Sharing over Mobile Cloud", IEEE Access, Vol. 9, 2021.

[13] Omer Aslan, Merve Ozkan-Okay, and Deepti Gupta, Intelligent Behavior-Based Malware Detection System on Cloud Computing Environment, IEEE Access, Vol. 9, 2021.

[14] Jian Zhang, Cheng Gao, Liangyi Gong, Zhaojun Gu, Dapeng Man, Wu Yang, Wenzhen Li, Malware Detection Based on Multi-level and Dynamic Multi-feature Using Ensemble Learning at Hypervisor, Mobile Networks and Applications, Vol. 26, 2021, 1668–1685.

[15] Enes Sinan Parildi, Dimitrios Hatzinakos, Yuri Lawryshyn, Deep learning-aided runtime opcode-based Windows malware detection, Neural Computing and Applications, 33, 2021, pp. 11963–11983. 117

[16] Farhan Ullah, Gautam Srivastava and Shamsher Ullah, A malware detection system using a hybrid approach of multi-heads attention-based control fow traces and image visualization, Journal of Cloud Computing: Advances, Systems and Applications, 2022.

[17] Liu, C., Lu, J., Feng, W., Du, E., Di, L., & Song, Z., MOBIPCR: Efficient, accurate, and strict ML-based mobile malware detection, Future Generation Computer Systems, Vol. 144, 2023, pp140-150.

[18] Wen, Q., & Chow, K. P., CNN based zero-day malware detection using small binary segments, Forensic Science International: Digital Investigation, Vol. 38, 2021, pp301128.

[19] Vekariya, Daxa, et al. "Mengers Authentication for efficient security system using Blockchain technology for Industrial IoT (IIOT) systems." 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE). IEEE, 2023.

[20] Kumar, M. S., Girinath, S., Lakshmi, G. G. V. S., Ganesh, A. V. S., & Kumar, K. J. (2023, September). Crop Yield Prediction Using Machine Learning. In 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET) (pp. 569-573). IEEE.

[21] Kumar, M. S., Girinath, S., Lakshmi, G. G. V. S., Ganesh, A. V. S., & Kumar, K. J. (2023, September). Crop Yield Prediction Using Machine Learning. In 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET) (pp. 569-573). IEEE.

[22] Kumar, M. Sunil, et al. "Simulation of the electrical control unit (ECU) in automated electric vehicles for reliability and safety using on-board sensors and Internet of Things." 2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE, 2023.

[23] Gandikota, H. P., Abirami, S., & Kumar, M. S. (2023). Bottleneck Feature-Based U-Net for Automated Detection and Segmentation of Gastrointestinal Tract Tumors from CT Scans. Traitement du Signal, 40(6).

[24] Kumar, M. Sunil, et al. "Enhancing Sentiment Analysis with an AttentionBased Machine Learning Model." 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS). IEEE, 2023.

[25] Natarajan, V. Anantha, and M. Sunil Kumar. "Improving qos in wireless sensor network routing using machine learning techniques." 2023 International Conference on Networking and Communications (ICNWC). IEEE, 2023.

[26] Ganesh, D., et al. "Implementation of Novel Machine Learning Methods for Analysis and Detection of Fake Reviews in Social Media." 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS). IEEE, 2023.

[27] Godala, Sravanthi, and M. Sunil Kumar. "A weight optimized deep learning model for cluster based intrusion detection system." Optical and Quantum Electronics 55.14 (2023): 1224.

[28] Sangamithra, B., BE Manjunath Swamy, and M. Sunil Kumar. "Evaluating the effectiveness of RNN and its variants for personalized web search." Optical and Quantum Electronics 55.13 (2023): 1202.

[29] Tian, D., Ying, Q., Jia, X., Ma, R., Hu, C., & Liu, W., MDCHD: A novel malware detection method in cloud using hardware trace and deep learning, Computer Networks, Vol. 198, 2021, pp108394.

[30] Gao, X., Hu, C., Shan, C., Liu, B., Niu, Z., & Xie, H., Malware classification for the cloud via semi-supervised transfer learning, Journal of Information Security and Applications, Vol. 55, 2020, pp102661.

[31] S. Depuru, P. Hari, P. Suhaas, S. R. Basha, R. Girish and P. K. Raju, "A Machine Learning based Malware Classification Framework," 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2023, pp. 1138-1143, doi: 10.1109/ICSSIT55814.2023.10060914.

[32] S. Depuru, K. Santhi, K. Amala, M. Sakthivel, S. Sivanantham and V. Akshaya, "Deep Learning-based Malware Classification Methodology of Comprehensive Study," 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2023, pp. 322-328,

[33] G. Rajeswarappa, S. Depuru and S. Sirisala, "Crop Pests Identification based on Fusion CNN Model: A Deep Learning," 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2023, pp. 968-974,

[34] Ganesh, A., Depuru, **S**. ., Reddy A., B. ., & Sujatha, G. . (2023). Streamlining Cancer Diagnosis and Prognosis System using Hybrid CNN-NPR: Deep Learning Approaches. International Journal of Intelligent Systems and Applications in Engineering, 12(3s), 190–201,

[35] Shola Usharani, Rajarajeswari Subbaraj, Appalaraju Muralidhar, Gayathri Rajakumaran, Srinivas Nandam, "Improvised Schinder Model for Anaesthesia Drug Delivery in Obese Patients with Optimized Infusion Rate and Patient Safety," International Journal of Engineering Trends and Technology, vol. 71, no. 9, pp. 256-264,

[36] Prathima Ch, R. Swathi, K. Suneetha, I. Suneetha, B. V. Suresh Reddy, "Image Capturing and Deleting Duplicate Images through Feature Extraction using Hashing Techniques," International Journal of Engineering Trends and Technology, vol. 72, no. 1, pp. 64-70, 2024.

[37] Anjana Nandam, P.A. Ramesh, M. Sakthivel, K. Amala , Sivanantham, “Human Emotion Recognition System Using Deep Learning Technique”, Journal of Pharmaceutical Negative Results, vol. 13, no. 4, pp. 1031–1035, Nov. 2022.