International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 02 | Feb 2025 www.irjet.net p-ISSN: 2395-0072

Advancing Cyber Security Through Optimized Deep Learning Algorithm: A Comprehensive Approach to Threat Detection

Sumalatha M S[1], Manju J[2]

[1] *Associate Professor, Department of computer science and engineering, Mahaguru Institute of Technology, Kayamkulam*
[2]*Professor, Mahaguru Institute of Technology, Kayamkulam*

Abstract - *Threat detection in cybersecurity involves identifying and responding to potential cyber threat, such as malware, phishing attacks, and unauthorized access, by analysing network traffic and system behavior. Effective threat detection enables organizations to proactively mitigate risks, keep sensitive data, and ensure the resilience of their digital infrastructures. Traditional threat detection techniques struggle to detect new and evolving threats like zero-day attacks. They often produce high false positives and negatives, reducing their effectiveness. Additionally, these methods rely on known patterns, making them slow to adapt to emerging cyber threats. This study investigates the potential of deep learning (DL) for cybersecurity by employing the ResNet101 model to enhance threat detection using the STIN dataset. This study focuses on enhancing the accuracy of threat detection, minimizing false positives, and ensuring rapid responses to emerging cybersecurity risks. The ResNet101 model achieved exceptional performance, with an accuracy of 96.24%, precision of 94.28%, F1 score of 96.47, and recall of 98.67%. These results highlight the model's reliability in identifying and mitigating cybersecurity threats. Additionally, the ResNet101 model processes the STIN dataset in approximately 55 seconds, demonstrating its suitability for real-time threat detection applications. This study presents a comprehensive analysis of dataset pre-processing techniques, model optimization strategies, and performance evaluation methods. The findings contribute the knowledge in AI-driven cybersecurity, offering valuable insights for both practitioners and researchers employing to strengthen digital infrastructure against sophisticated cyber threats*.

Key Words: *Threat Detection, cyber security, ResNet 101, deep learning, intrusion detection.*

1.INTRODUCTION

Due to the fast growth of digital infrastructures and networked systems, which exposes enterprises to an ever-increasing number of sophisticated cyber threats, cybersecurity has become a crucial area in the digital age. Businesses, governments, and individuals are at serious risk from cyberattacks, which can range from advanced persistent threats (APTs) to malware infections and phishing schemes. Because of the complexity and magnitude of contemporary cyber threats, traditional cybersecurity techniques like heuristic analysis and signature-based detection have not been able to keep up. This difficulty highlights the necessity for cutting-edge approaches that can improve the capacity to detect threats and offer preventative defenses. A subset of AI, DL has emerged as a potent instrument in cybersecurity, with the ability to identify and eliminate cyberthreats with high precision and effectiveness. DL algorithms can evaluate enormous volumes of data, spot complex patterns, and spot abnormalities that can point to criminal activity by utilizing complex neural network structures. DL models are extremely effective in a constantly changing threat landscape because, in contrast to previous techniques, they can continuously learn and adapt to new attack patterns [1].

In order to improve the threat detection DL algorithms, presenting a holistic approach to cybersecurity. To attain better performance, DL models must be optimized by adjusting hyperparameters, enhancing feature extraction methods, and incorporating innovative architectures. The suggested method seeks to balance detection accuracy, computing efficiency, and real-time response capabilities by utilizing optimization techniques. The study also examines the difficulties in putting DL-based cybersecurity solutions into practice, such as data availability and quality, adversarial assaults, and the interpretability of model results. For AI-driven security solutions to be dependable and trustworthy, these issues must be resolved. In order to develop thorough and complete threat detection models, the study also explores the significance of utilizing a variety of data source and threat intelligence feeds. The summarization of this study are as follows:

- An effective threat detection framework that makes use of the pre-trained model and ResNet 101 architecture.
- Offers a thorough examination of current DL methods used in cybersecurity, pointing out their advantages and disadvantages.
- The suggested ResNet 101 and pre-trained model performance are assessed using a standard performance metrics.

The following is an arrangement of the remaining sections: A review of the literature assessing previous research is given in Section 2. The suggested threat detection

methodology is described in Section 3. Section 4 offers results and discussion emphasizing the model's excellent accuracy and performance metrics and Section 5 concludes the study by summarizing findings.

2. RELATED WORKS

Akgun et al. [2] presented an intrusion detection system (IDS) based on a DL model to detect Distributed Denial of Service (DDoS) attacks. The authors selected the CICDDoS 2019 dataset, which contained 12 classes. To identify the most effective architecture, various models (DNN, CNN, LSTM) with varying layer configurations were compared. Feature elimination and selection techniques were applied to reduce the original 88 features to 40 key attributes. The model achieved a better accuracy for both multi-class classification and binary classification. For real-time analysis, the study was limited to evaluating the inference times of candidate models to understand their real-time performance. Ashiku et al. [3] proposed a DL architecture for developing an adaptive and resilient framework for an IDS to detect and classify network attacks effectively. To validate the effectiveness of the model, the authors utilized the UNSW-NB15 dataset, which reflects real-world network communication patterns and synthetic-generated attack activities. The model showed the overall accuracy of 95.4%. Despite promising results, the model's performance was limited by optimized feature reduction techniques.

Mughaid et al. [4] proposed a comprehensive summary of ML and phishing attacks, highlighting the prevalence of phishing emails as the most effective tactic. A detection model used ML techniques and was trained on three different datasets. The results showed that using a higher number of features yielded more accurate and efficient outcomes. The ML algorithm achieved 88% accuracy on boosted decision trees on the respective datasets. The research faced limitations in detection rate due to challenges in identifying the phishing emails copying legitimate mail. Yu et al. [5] introduced a DL-based detection scheme for advanced persistent threats (APTs) in IoT environments and adopted a bidirectional encoder representation from transformers (BERT). The APT sequence was improved to enhance the model's effectiveness in judging prolonged sequences. The model had a limitation in accuracy when applied to the IoT platform.

Hnamte et al. [6] introduced a DL-based network IDS to secure network from cyberattacks. The study presented an improved DCNNBiLSTM model for NID, utilizing the hybrid CNN and LSTM networks. The model was trained on real-time network data from Edge IoT and the CICIDS2018 dataset, and the model's performance was examined by using multiclass classification, and the model attained better accuracy. However, the model trained on balanced data did not yield superior performance, and the imbalanced data also had its limitations. Mighan et al. [7] presented a hybrid SVM-SAE model for effective cybersecurity intrusion detection. The system used an autoencoder (SAE) network and a SVM model. The efficacy of the model was evaluated using the Apache Spark ML technique. The authors assessed the SVM-SAE framework by applying dimensionally reduction to the 42-dimensional ISCX dataset to 75% of its unique features. The reduced data was then classified using SVM. Processing time was reduced by removing the redundant features from the ISCX 2012 dataset. Adding decision tree and SAE improved both accuracy and speed. However, the model failed to achieve optimal results.

A DL method for insider threat identification using behavioural analysis of cyberattacks was presented by Nasir et al. [8]. The technique detected insiders with a better accuracy and low FP rate by utilising a rich feature set that included practical units, user roles, and logon/logoff events. In particular, LSTM-Autoencoder was used to detect insider threats. The CMU CERT v4.2 datasets were used for training and testing, and the model's accuracy was 90.60%. One major obstacle was the absence of threat scenarios that were accessible to the general public. Al-Abassi et al. [9] introduced a DL approach to address class imbalance in industrial control system (ICS) datasets. The proposed method produced balanced representations of the data, which are then fed into an ensemble DL model for attack detection. The model combined decision trees (DTs) and DNNs to identify cyberattacks. Evaluated using 10-fold cross-validation on two real-world ICS datasets, the model achieved better performance over conventional classifiers (RF, DNN, and AdaBoost) and recent state-of-the-art models. The model performed better than existing techniques but struggled with limited samples. Limitations in identifying attack types and locations, delaying effective downtime prevention.

Maddireddy et al. [10] presented a comprehensive study to enhance early threat detection to prevent cyberattacks by utilizing AI for proactive cyber defence, integrating ML, DL, and data analytics. The methodology involved collecting and analysing diverse data sources, applying ML algorithms for anomaly detection, and utilizing DL models (CNNs and RNNs) for in-depth analysis. ML algorithms showed better accuracy in identifying anomalous activities, empowering security teams to proactively measures and effectively mitigate potential cyber threats. However, the study faced several limitations, including data availability and sample size constraints, as well as the inherent complexities of AI-driven cybersecurity analysis. Additionally, biases in data collection and analysis processes impacted the relevance of the findings.

Traditional threat detection methods in cybersecurity face several significant challenges cyber threats. Traditional signature-based threat detection methods in cyber networks are inherently flawed, as they fail to identify zero-day attacks and often yield high false positive and false negative rates, compromising their effectiveness. Researchers faced limitations in threat detection rates, primarily due to the

difficulty in identifying phishing emails that convincingly replicate the branding, language, and structure of authentic messages [4]. The model's accuracy was compromised in the IoT platform, due to the complexity and diversity of IoT devices, disparate communication protocols, and the massive volume of generated data, which are resulting in increased exposure to potential susceptibilities and elevated risk of missed detections [5]. The scarcity of publicly available threat scenarios posed a significant challenge, as it limited the ability to test and validate detection models. This gap highlighted the need for more realistic and diverse scenario development, facilitating researchers and practitioners to better simulate and prepare for the dynamic range of cyber threats in real-world environments [8].

3. MATERIALS AND METHODS

The study presents a comprehensive method to threat detection using predictive modeling with a ResNet 101 architecture. The methodology, illustrated in Fig. 1, involves a systematic process of attribute extraction from the STIN dataset, followed by extensive data pre-processing to ensure consistency, quality, and quantity. The trained model is utilized for threat detection, achieving efficient and accurate results. The computational efficiency of the model is assessed by measuring its execution time on the STIN dataset, providing valuable insights into its performance. The proposed approach provides high-level network security by analysing and examining threat detection, making it an appreciated influence in cyber security field.

The use of a pre-trained ResNet 101 architecture enables the model to learn composite designs and features from the data, resulting in improved threat detection accuracy. The systematic approach presented in this research provides a robust framework for threat detection, which can be applied to various network security scenarios.

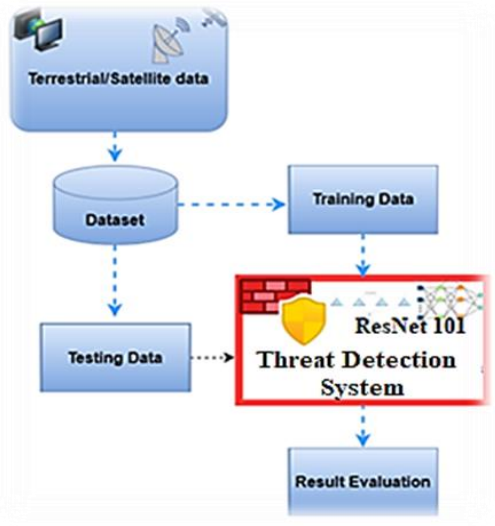

Fig - 1: shows the architecture of the suggested threat detection system

3.1 Data Description

This study evaluates the performance effectiveness of a proposed thread detection framework using the STIN dataset. The STIN security dataset encompasses various types of attacks in both satellite and terrestrial networks, comprising nine terrestrial-type attacks and two satellite-type attacks. The dataset is built using flow-based features, which are detailed in Table 1.

Table – 1: Features of SKIN dataset

Domain	Attack Type
Terrestrial attacks	MSSQL DDoS
	UDP DDoS
	Portmap DDoS
	Backdoor
	Syn DDoS
	LDAP DDoS
	NetBIOS DDoS
	Web attack
	Botnet
Satellite attacks	DUP DDoS
	Syn DDoS

3.2 Data Pre-processing

To formulate cybersecurity data for analysis, collect labelled data such as system logs and network traffic, which provide insights into network activity (e.g., data traffic heat maps). Pre-process the data through standardization or normalization to safeguard features are on the identical scale. Transforming raw network traffic or log data into CNN-friendly formats, such as time-series representations or 2D heat maps, is represented in Equation 1.

$$\hat{x} = \frac{x-\mu}{\sigma} \quad (1)$$

where x represents the raw data, μ and σ represent the dataset's mean and standard deviation, and $\hat{x}$ represents the normalized data.

3.3 Proposed methodology

The ResNet101 architecture consists of 101 layers, formed by residual blocks that utilize skip connections to enable efficient information transfer between layers as illustrated in Fig. 2. ResNet core concept involves learning residual mappings, rather than the original unreferenced mapping, allowing for effective training of deeper models without degradation issues [11].

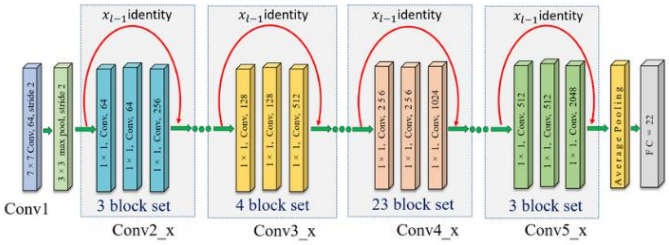

Fig - 2: Architecture of the ResNet 101 model

The ResNet101 model utilizes residual blocks to extract deep features from input data, capturing complex patterns essential for risk detection, as represented in Equation 2. The ResNet 101 model is shown in Fig. 2.

$$F = ResNet101(\hat{x}) \quad (2)$$

where F is the feature vector that results from running the ResNet101 model on the input $\hat{x}$. Back propagation and gradient descent techniques, usually Adam or SGD (Stochastic Gradient Descent), are used to optimize the model weights during the training phase in order to minimize a loss function. Cross-entropy loss, as shown in Equation 3, is utilized as the loss function for classification problems.

$$L = -\sum_{i=1}^{n} y_i \log(\widehat{y_i}) \quad (3)$$

Where, y_i is the actual label of class, $\widehat{y_i}$ class is the expected probability, n is the entire number of classes (such as various attack kinds). The modified weights W are shown in Equation 4

$$W = W - \eta \nabla W L \quad (4)$$

ResNet101 output will be passed through a softmax function to assign a probability in each class which is represented in Equation 5.

$$\widehat{p_i} = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \quad (5)$$

When class I's output score is represented by z_i and $\hat{p}_i$, the class's expected probability is denoted by i. The anticipated threat type is chosen from the class with the highest probability. Equation 6 provides a formal definition of each residual block in ResNet101.

$$y = F(x, \{W_i\}) + x \quad (6)$$

The residual block's output (y) is computed by adding the block's input (x) to the output of the convolutional layers $F(x, \{Wi\})$ where, $\{Wi\}$ represents weights of the convolutional layers. By doing so, the network learns the residual function $F(x)$, which is a simpler mapping to learn than the direct mapping $H(x)$, enabling easier optimization of the residual functions.

3.4 Experiment setup

The proposed model was implemented and trained using Python and Keras on Google Colab. The experimental system ran on Windows 10, with TensorFlow as the backend, 12.75 GB RAM, and GPU acceleration. The system utilized an NVIDIA GeForce GTX 1080Ti GPU, providing a significant boost in processing power, and an Intel Core i7-6850K processor. The model's architecture was optimized through hyperparameter tuning, involving systematic exploration of hyperparameter configurations to identify the optimal combination yielding the best classification performance. The results of hyperparameter tuning are presented in Table 2. The system's configuration enabled efficient processing and storage of large datasets, rapid training and testing of the model, and seamless execution of DL computations. Overall, the system provided a robust and efficient environment for developing and testing the proposed model. Tuning the hyperparameters was essential to improving the model's performance.

Table – 2: The suggested model's Hyperparameters

Hyper parameters	Values
Loss Function	Categorical Cross entropy
Number of Epochs	25
Learning rate	0.001
Dropout	0.2
Batch Size	16
Optimizer	Adam
Activation Function	Sigmoid

4. RESULTS AND DISCUSSION

Evaluation metrics were utilized to assess the performance of the classifier. These metrics, represented by Equations (7) to (10), provide a comprehensive comparison of algorithmic effectiveness. The assessment of these metrics is based on a confusion matrix, which predicts the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), for a detailed evaluation of classifier performance.

$$Accuracy = \frac{TP+TN}{TP+TN+FN} \quad (7)$$

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (10)$$

STIN dataset is used to test the suggested ResNet 101 ability to offer excellent security for both satellite and terrestrial networks. The ResNet101 model's ability to deliver excellent security for terrestrial- satellite networks were thoroughly evaluated using the STIN dataset. This dataset served as a comprehensive benchmark to identify the model's performance in distinguishing and modifying various cyber threats. By utilizing the STIN dataset, the study demonstrates the ResNet101 model's efficacy in ensuring the security and integrity of network communications, thereby providing a robust defence against potential cyberattacks. The results showed promising outcomes. The ResNet101 model's performance was assessed using the STIN dataset, and the results demonstrated its superiority over other models. The model achieved 97.67% accuracy in detecting UDP_DoS attacks, showcasing its exceptional capability in identifying this specific type of threat. However, its performance was slightly lower for Syn_DDoS attacks, with an accuracy of 97.25%.

The categorization results, accessible in Table 3, further highlight exceptional performance of the model. While the accuracy scores for the "Portmap DDoS" and "LDAP DDoS" classes were lower, at 91.21% and 93.14% respectively, the overall performance of the model was still superior to that of other ML models. The lower accuracy scores for these specific classes can be recognized to the limited number of training samples available. Despite this, the ResNet101 model demonstrated its effectiveness in detecting cyber threats, building it a valuable tool in the field of network security. These results are presented in Fig. 3, which illustrates the classifier's accuracy on the STIN terrestrial dataset.

Table – 3: Classifier accuracy of the STIN terrestrial dataset

Attack Type	Accuracy of the proposed ResNet 101 model (%)
UDP DDoS	97.67
Backdoor	96.67
Syn DDoS	97.25
LDAP DDoS	93.14
Portmap DDoS	91.21
MSSQL DDoS	95.24
NetBIOS DDoS	96.65

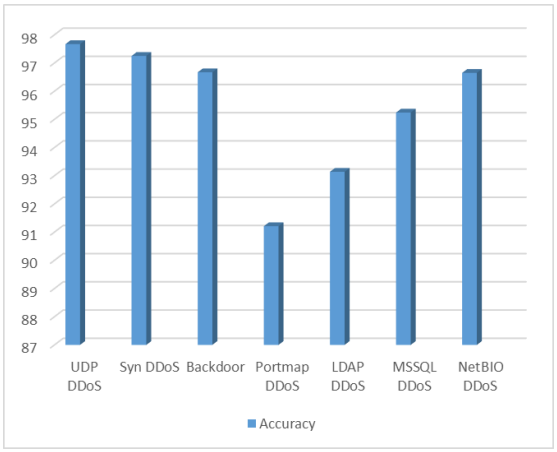

Fig – 3: Visualization of classifier accuracy on the STIN terrestrial dataset

The performance ResNet 101, was evaluated by a range of metrics. Table 4. presents the average classifier accuracy of STIN dataset, shows an overview of performance in the model. Fig. 4 shows a visual depiction of the model's effectiveness by displaying the average classification accuracy. These outcomes show how well the ResNet 101 model detects and categorises assaults, resulting in a useful tool in the area of network security. Overall, the systems strong performance across the metrics highlights its potential for real-world applications.

Table – 4: Average classifier accuracy of the STIN dataset

Model	Dataset	Parameters	Values (%)
ResNet 101	STIN	Accuracy	96.24
		Precision	94.28
		F1 Score	96.47
		Recall	98.67

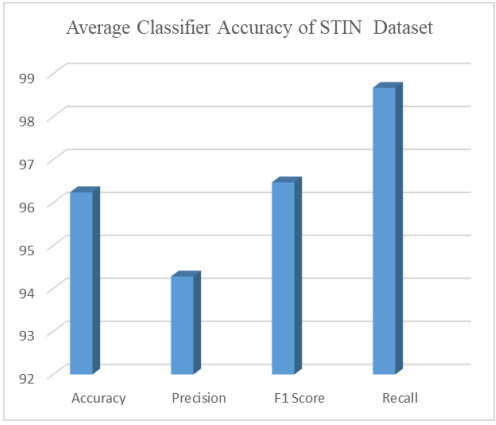

Fig. - 4: Average Classifier Accuracy of STIN Dataset

The ResNet 101 model demonstrated outstanding performance on the STIN dataset and achieved the remarkable accuracy. The receiver operating characteristics (ROC) curve, illustrated in Fig. 5, further highlights the model's superiority. Notably, the model's computational complexity was assessed, revealing an estimated execution time of 55 seconds on the STIN dataset. These findings collectively underscore the ResNet 101 model's effectiveness and efficiency in detecting cyber threats, making it a valuable tool for providing the security of satellite and terrestrial networks.

The proposed ResNet 101 model can be used as a benchmark for future research. Overall, this research has made a valuable role in field of cyber security and has provided a valuable tool in detection and classification for cyber threats in satellite- terrestrial networks.

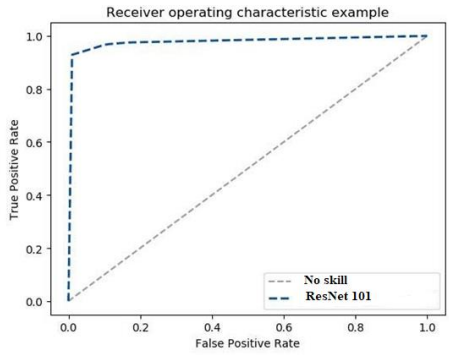

Fig – 5: The suggested ResNet 101 model's ROC curve

5. CONCLUSION

This study concludes by highlighting the significant contribution of the ResNet101 model to advancing cybersecurity through the application of optimal DL methods. With an accuracy of 96.24%, precision of 94.28%, F1 score of 96.47, and recall of 98.67, the model demonstrates outstanding performance in reliable threat detection. Its effectiveness for real-time applications is further underscored by its processing time of approximately 55 seconds, making it a viable solution for addressing modern cybersecurity challenges. The approach demonstrates its outstanding ability to rapidly and accurately detect and address attacks by using the STIN dataset. This study emphasises how important DL approaches are for improving security procedures and adjusting possible hazards in dynamically settings. Future work should explore the integration of diverse datasets and the development of hybrid strategies that combine traditional security approaches with DL. Additionally, the model's ability to adapt to evolving cyber threats can be enhanced through the incorporation of adaptive learning techniques. Overall, the study contribute to the advancement of AI-powered cybersecurity solutions and lay a solid foundation for the effective protection of digital infrastructures.

REFERENCES

[1] Li, Y., & Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. Energy Reports, 7, 8176-8186.
[2] Akgun, D., Hizal, S., & Cavusoglu, U. (2022). A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. Computers & Security, 118, 102748.
[3] Ashiku, L., & Dagli, C. (2021). Network intrusion detection system using deep learning. Procedia Computer Science, 185, 239-247.
[4] Mughaid, A., AlZu'bi, S., Hnaif, A., Taamneh, S., Alnajjar, A., & Elsoud, E. A. (2022). An intelligent cyber security phishing detection system using deep learning techniques. Cluster Computing, 25(6), 3819-3828.
[5] Yu, K., Tan, L., Mumtaz, S., Al-Rubaye, S., Al-Dulaimi, A., Bashir, A. K., & Khan, F. A. (2021). Securing critical infrastructures: deep-learning-based threat detection in IIoT. IEEE Communications Magazine, 59(10), 76-82.
[6] Hnamte, V., & Hussain, J. (2023). DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system. Telematics and Informatics Reports, 10, 100053.
[7] Mighan, S. N., & Kahani, M. (2021). A novel scalable intrusion detection system based on deep learning. International Journal of Information Security, 20(3), 387-403.
[8] Nasir, R., Afzal, M., Latif, R., & Iqbal, W. (2021). Behavioral based insider threat detection using deep learning. IEEE Access, 9, 143266-143274.
[9] Maddireddy, B. R., & Maddireddy, B. R. (2020). Proactive Cyber Defense: Utilizing AI for Early Threat Detection and Risk Assessment. International Journal of Advanced Engineering Technologies and Innovations, 1(2), 64-83.
[10] Singh, P.; Pankaj, A.; Mitra, R. Edge-detect: Edge-centric network intrusion detection using deep neural network. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021; pp. 1–6.
[11] Demir, A., Yilmaz, F., & Kose, O. (2019, October). Early detection of skin cancer using deep learning architectures: resnet-101 and inception-v3. In 2019 medical technologies congress (TIPTEKNO) (pp. 1-4). IEEE.