International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 03 | Mar 2025 www.irjet.net p-ISSN: 2395-0072

Enhancing Cyber security Using an Enhanced Deep Learning Algorithm: An All-Inclusive Method for Threat Identification

Mahesh S Markad[1]

[1]*Head of Computer engineering Department Samajbhushan Eknathrao Dhakane College of Engineering Shevgaon*

Abstract - *In cybersecurity, threat detection is examining network traffic and system behavior to find and address possible cyberthreats such malware, phishing scams, and unauthorized access. Organizations may proactively reduce risks, preserve sensitive data, and guarantee the robustness of their digital infrastructures through effective threat detection. New and changing threats, like as zero-day assaults, are difficult for traditional threat detection systems to identify. Their efficacy is diminished since they frequently generate a significant number of false positives and negatives. Furthermore, because these techniques depend on well-established patterns, they are sluggish to adjust to new cyberthreats. Using the ResNet101 model to improve threat detection with the STIN dataset, this work explores the possibilities of deep learning (DL) for cybersecurity. Improving threat detection accuracy, reducing false positives, and guaranteeing prompt reactions to new cybersecurity threats are the main objectives of this study. With an accuracy of 96.24%, precision of 94.28%, F1 score of 96.47, and recall of 98.67%, the ResNet101 model demonstrated outstanding performance.*

These outcomes demonstrate how well the model detects and reduces cybersecurity risks. Furthermore, the ResNet101 model shows its potential for real-time threat detection applications by processing the STIN dataset in about 55 seconds. This paper offers a thorough examination of model optimization tactics, performance assessment methodologies, and dataset pre-processing approaches.

The results advance our understanding of AI-driven cybersecurity and provide insightful information that practitioners and researchers may use to fortify digital infrastructure against advanced cyberthreats.

Key Words: Cyber security, deep learning, ResNet 101, threat detection, and intrusion detection.

1.INTRODUCTION

Cybersecurity has become a critical topic in the digital age due to the rapid expansion of digital infrastructures and networked systems, which exposes businesses to an ever-increasing number of sophisticated cyber threats.

Cyber attacks, which can include malware infections, phishing schemes, and advanced persistent threats (APTs), pose a major risk to individuals, governments, and businesses. Given the intricacy and scope of today's cyberthreats, traditional cybersecurity. Methods such as signature-based detection and heuristic analysis have fallen behind. This challenge emphasizes the need for innovative strategies that can enhance the ability to identify threats and provide preemptive measures. DL, a subset of AI, has become a powerful tool in cybersecurity, able to accurately and successfully detect and eliminate cyberthreats.

By employing sophisticated neural network architectures, DL algorithms are able to analyze vast amounts of data, identify intricate patterns, and identify anomalies that may indicate illegal activities. Unlike earlier methods, DL models can continuously train and adapt to new attack patterns, which makes them incredibly effective in a threat landscape that is always evolving. offering a comprehensive strategy to cyber security in order to enhance the threat detection DL algorithms. DL models must be optimized by modifying hyper parameters, improving feature extraction techniques, and implementing novel architectures in order to get higher performance. By applying optimization approaches, the proposed approach aims to strike a compromise between detection accuracy, computational efficiency, and real-time response capabilities. The paper also looks at the challenges of implementing DL-based cybersecurity solutions, including adversarial attacks, data quality and availability, and the interpretability of model output. These problems need to be fixed if AI-driven security solutions are to be reliable and trustworthy. The study also examines the importance of using a range of data sources and threat intelligence feeds in order to create comprehensive and full threat detection models. The following is a summary of this study:

A powerful threat detection framework that utilizes the ResNet 101 architecture and the pre-trained model.

Examines contemporary DL techniques in cybersecurity in detail, highlighting both their benefits and drawbacks.

Standard performance measures are used to evaluate the performance of the pre-trained model and the proposed ResNet 101. The remaining sections are arranged as follows:

Section 2 provides a review of the literature evaluating earlier studies. The recommended threat identification

Section 3 provides a description of methods. Results and a discussion of the model's exceptional accuracy and performance metrics are provided in Section 4, and a

summary of the study's conclusions is provided in Section 5…/ n

2. CONNECTED WORKS

In order to identify Distributed Denial of Service (DDoS) assaults, Akgun et al. [2] created an intrusion detection system (IDS) built on a DL model. The CICDDoS 2019 dataset, which included 12 classes, was chosen by the authors. Different models (DNN, CNN, and LSTM) with different layer configurations were examined in order to determine the best efficient architecture.

The initial 88 characteristics were reduced to 40 essential properties using feature reduction and selection strategies. For both binary and multi-class classification, the model's accuracy was improved. The study was restricted to assessing candidate models' inference times in order to comprehend their real-time performance.In order to provide an adaptable and resilient framework for an IDS to efficiently identify and categorize network assaults, Ashikuetal. [3] suggested a DL architecture. The UNSW-NB15 dataset, which replicates actual network communication patterns and artificially manufactured attack activity, was used by the authors to verify the model's efficacy. The total accuracy of the model was 95.4%. Optimized feature reduction strategies hampered the model's performance despite encouraging results.

A thorough overview of ML and phishing attempts was put out by Mughaid et al. [4], who also noted that phishing emails were the most common and successful strategy. Three distinct datasets were utilized to develop a detection model that employed machine learning techniques. The findings demonstrated that additional characteristics produced more accurate and effective results. On the corresponding datasets, the ML algorithm's accuracy on boosted decision trees was 88%. The study encountered detection limits as a result of difficulties recognizing phishing emails that mimic authentic communications.Yu et al. [5] used a bidirectional encoder representation from transformers (BERT) and presented a DL-based detection strategy for advanced persistent threats (APTs) in IoT contexts. To increase the model's ability to judge lengthy sequences, the APT sequence was modified. When used on the IoT platform, the model's accuracy was limited. To protect networks from cyberattacks, Hnamte et al. [6] created a DL-based network intrusion detection system. The study used the hybrid CNN and LSTM networks to provide an enhanced DCNNBiLSTM model for NID. The CICIDS2018 dataset and real-time network data from Edge IoT were used to train the model. Multiclass classification was used to assess the model's performance, and it achieved higher accuracy. Nevertheless, the model that was built with balanced data did not perform better, and the unbalanced data had drawbacks as well. A hybrid SVM-SAE model was introduced by Mighan et al. [7] for efficient cybersecurity intrusion detection. detection. The system made use of an SVM model and an autoencoder (SAE) network. The Apache SparkML approach was used to assess the model's effectiveness.By applying dimensional reduction to 75% of the 42-dimensional ISCX dataset's unique characteristics, the authors evaluated the SVM-SAE framework. SVM was then used to classify the generated data. The ISCX 2012 dataset's duplicate characteristics were eliminated to save processing time. Accuracy and speed were increased by including decision trees and SAE. Nevertheless, the model did not produce the best outcomes.

Nasir et al. [8] introduced a DL technique for insider threat detection via behavioral analysis of cyberattacks. The method used a broad feature set that comprised logon/logoff events, user roles, and practical units to detect insiders with a low FP rate and higher accuracy.

Specifically, insider threats were detected using the LSTM-Autoencoder. The accuracy of the model, which was trained and tested using the CMUCERTv4.2 datasets, was 90.60%. The lack of publicly available threat scenarios was one of the main challenges. A DL technique was presented by Al-Abassi et al. [9] to handle class imbalance in datasets from industrial control systems (ICS). An ensemble DL model is then used to detect attacks using the balanced representations of the data that the suggested approach generated. To detect cyberattacks, the model included decision trees (DTs) with DNNs.When tested using 10-fold cross-validation on two real-world ICS datasets, the model outperformed both contemporary state-of-the-art models and traditional classifiers (RF, DNN, and AdaBoost). The model struggled with small sample sizes but outperformed current methods. Effective downtime prevention is delayed by limitations in detecting attack kinds and locations.

Maddireddy et al. [10]provided a thorough study that integrated ML, DL, and data analytics with AI for proactive cyber defense to improve early threat identification and stop intrusions.The approach comprised gathering and evaluating various data sources, using DL models (CNNs and RNNs) for in-depth analysis, and using ML algorithms for anomaly identification. Better accuracy in detecting unusual activity was demonstrated by ML algorithms, enabling security teams to take preventative action and successfully lessen any cyberthreats. The study did, however, have a number of drawbacks, such as restrictions on sample size and data availability in addition to the inherent difficulties of AI-driven cybersecurity analysis. Furthermore, biases in the procedures used for data collecting and analysis affected the findings' applicability.

Traditional threat detection techniques in cybersecurity confront a number of serious obstacles from online attacks. Conventional signature-based threat detection techniques are intrinsically faulty since they are unable to detect zero-day assaults and frequently produce large false positive and false negative rates, which reduces their efficacy. Threat detection rates were limited for researchers, mostly because of the Finding phishing emails that accurately mimic the language, branding, and structure of real emails can be challenging

[4].Due to the complexity and diversity of IoT devices, different communication protocols, and the enormous amount of data created, the model's accuracy was impaired in the IoT platform. This increased susceptibility to potential vulnerabilities and higher chance of missed detections is the outcome [5]. A major obstacle was the lack of publicly accessible threat scenarios, which made it difficult to test and certify detection algorithms.In order to help academics and practitioners better model and prepare for the dynamic spectrum of cyber threats in real-world situations, this gap brought attention to the need for more realistic and varied scenario creation [8].

3. SUPPLIES AND TECHNIQUES

Using a ResNet 101 architecture and predictive modeling, the work offers a thorough approach to threat identification. In order to assure consistency, quality, and quantity, the methodology—which is shown in Fig. 1—involves a methodical process of attribute extraction from the STIN dataset, followed by intensive data pre-processing. When applied to threat detection, the trained model produces accurate and efficient results. By assessing the model's execution time on the STIN dataset, its computational efficiency is evaluated, offering important information about how well it performs.

The suggested method is a valued impact in the field of cyber security as it offers high-level network security through threat detection analysis and examination. The model can learn composite designs and characteristics from the data thanks to the usage of a pre-trained ResNet 101 architecture, which improves the accuracy of threat detection. This study's methodical methodology offers a strong foundation for threat identification that may be used in a variety of network security situations.

Figure 1 illustrates the proposed threat detection system's design.

3.1 Description of the Data

Using the STIN dataset, this study assesses the performance efficacy of a suggested thread detecting system. There are two satellite-type attacks and nine terrestrial-type attacks in the STIN security dataset, which covers a variety of assaults in both satellite and terrestrial networks. Flow-based characteristics, which are described in detail in Table 1, are used to construct the dataset.

Table 1: SKIN dataset characteristics

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 Pre-processing of Data

Gather labeled data, such as system logs and network traffic, which offer insights into network activity (e.g., data traffic heat maps), in order to synthesize cybersecurity data for analysis. To ensure that features are on the same scale, preprocess the data by standardizing or normalizing it. Equation 1 illustrates the process of converting unprocessed network traffic or log data into CNN-friendly forms, including time-series representations or 2D heat maps.

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

Where x denotes the raw data, denotes the $\hat{x}$ normalized data, μ and σ denotes the mean and standard deviation of the dataset.

3.3 Suggested methodology

The 101 layers of the ResNet101 design are made up of residual blocks that use skip connections to facilitate effective information flow across levels, as shown in

Figure 2.

ResNet basic principle is learning residual mappings, rather than the original unreferenced mapping, enabling for

effective training of deeper models without degradation concerns [11].

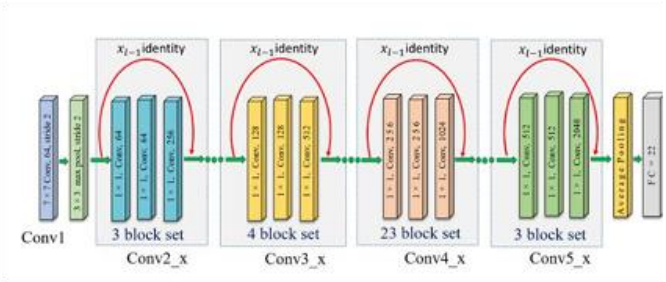


Figure 2: ResNet 101 model architecture

Equation 2 illustrates how the ResNet101 model uses residual blocks to extract deep features from input data, capturing intricate patterns crucial for risk identification.

In Figure 2, the ResNet 101 model is displayed.

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

The feature vector that emerges from applying the ResNet101 model to the input is denoted by F. The model weights are optimized during the training phase to minimize a loss function using gradient descent and back propagation techniques, often Adam or SGD (Stochastic Gradient Descent). Equation 3 illustrates how cross-entropy loss is used as the loss function in classification tasks.

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

Where y_i class is the actual class label, $\widehat{y_i}$ class is the predicted probability, and class is the total number of classes (e.g., different types of attacks). Equation 4 presents the adjusted weights .W

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

The output of ResNet101 will be run through a softmax function to determine a probability for each class, which is shown in Equation 5.

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

The class's predicted probability is shown by i. where the output score of class I is represented by z_i and $\hat{p}_i$. The class with the highest likelihood is used to select the expected threat type. Every residual block in ResNet101 has a formal definition given by Equation 6.

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

By adding (x) the block's input to the convolutional layers' output (y), the output of the residual block is calculated.

$F(x, \{Wi\})$ where $\{Wi\}$ denotes the convolutional layers' weights. This makes it easier to optimize the residual functions $F(x)$, since the network learns the residual function, which is a simpler mapping to learn than the direct mapping .

3.4 Experiment setup

The proposed model was implemented and trained using Python and Kerason Google Colab. The experimental system ranonWindows10,withTensor Flowasthebackend,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 hyper parameter tuning, involving systematic exploration of hyper parameter configurations to identify the optimal combination yielding the best classification performance.

The results of hyper parameter tuning are presented inTable2. 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 hyper parameters was essential to improving the model's performance.

Hyperparameters	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

The classifier's performance was evaluated using evaluation metrics. Equations (7) through (10), which reflect these indicators, offer a thorough comparison of algorithmic efficacy. For a thorough analysis of classifier performance, these metrics are evaluated using a confusion matrix that forecasts the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

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

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

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

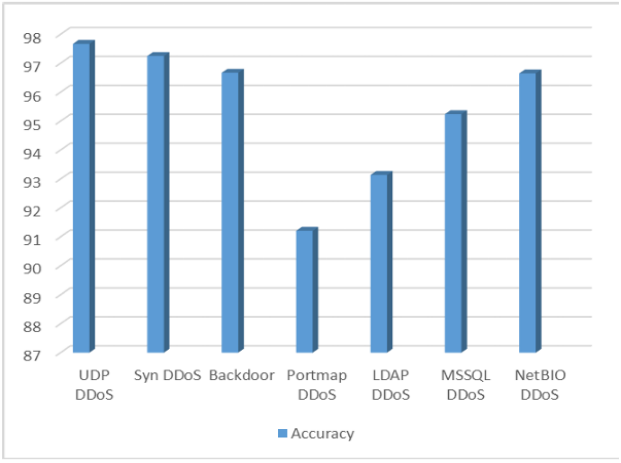
$$F1\ score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (10)$$

The STIN dataset is used to test the proposed Excellent security is provided by ResNet101ability for both terrestrial and satellite networks. Using the STIN dataset, the ResNet101 model's capacity to provide superior security for terrestrial-satellite networks was carefully assessed. This dataset was used as a thorough benchmark to determine how well the model performed in differentiating and altering different cyber threats. The study shows the effectiveness of the ResNet101 model in guaranteeing the security and integrity of network communications by using the STIN dataset, offering a strong defense against future cyber attacks. The findings were encouraging. The STIN dataset was used to evaluate the ResNet101 model's performance, and the findings showed that it outperformed other models.The model demonstrated its remarkable ability to recognize this particular threat type with a 97.67% accuracy rate in recognizing UDP_DoS assaults. With a 97.25% accuracy rate, it performed marginally worse against Syn_ DDoS assaults.

The classification results, which are shown in Table 3, further demonstrate the model's outstanding performance. The model performed better overall than previous ML models, even if the accuracy scores for the "Portmap DDoS" and "LDAP DDoS" classes were lower, at 91.21% and 93.14%, respectively. The poorer accuracy ratings for these particular classes can be attributed to the scarcity of training data. The ResNet101 model proved to be a useful tool in the field of network security despite this, since it was able to effectively detect cyber threats. Fig. 3, which displays the classifier's accuracy on the STIN terrestrial dataset, displays these findings.

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 A variety of measures were used to assess ResNet 101. Table 4 provides an overview of the model's performance and displays the average classifier accuracy for the STIN dataset. The average classification accuracy in Fig. 4 provides a visual representation of the model's efficacy. These results demonstrate how effectively the ResNet 101 model identifies and classifies attacks, producing a practical tool for network security. The system's excellent overall performance across all criteria points to its potential for practical uses.

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

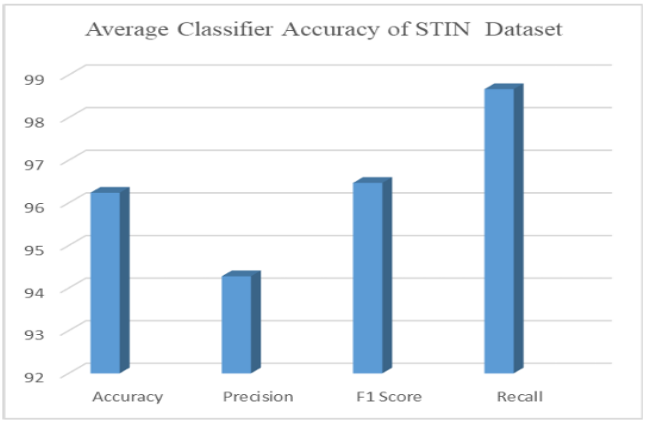
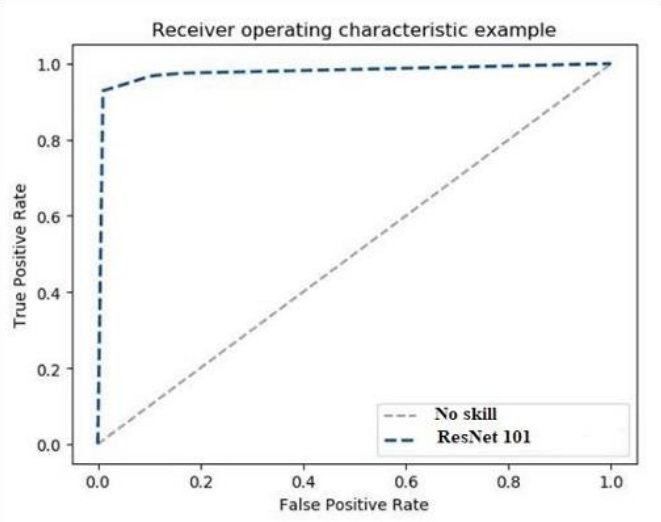


Fig.- 4: Average Classifier Accuracy of STIN Dataset

On the STIN dataset, the ResNet 101 model performed exceptionally well and attained a remarkable accuracy.The model's superiority is further demonstrated by the receiver operating characteristics (ROC) curve, which is shown in Fig. 5. Interestingly, an evaluation of the model's computational complexity showed that, using the STIN dataset, its estimated execution time was 55 seconds. Together, these results highlight how well the ResNet101 model detects cyberthreats, making it a useful tool for ensuring the security of both terrestrial and satellite networks.

Future studies can utilize the suggested ResNet 101 model as a standard. All things considered, this study has contributed significantly to the field of cyber security and offered a useful tool for identifying and categorizing cyberthreats in satellite-terrestrial networks.



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

5. CONCLUSION

The important contribution of the ResNet101 model to improving cybersecurity through the use of optimal DL techniques is highlighted in the study's conclusion. With a 96.24% accuracy rate, 94.28% precision rate, 96.47 F1 score, and 98.67 recall, the model performs exceptionally well in dependable threat detection. Its processing time of around 55 seconds further demonstrates its efficacy for real-time applications, giving it a workable answer to contemporary cybersecurity issues. Using the STIN dataset, the method exhibits its exceptional capacity to quickly and precisely identify and respond to threats. The significance of DL techniques for enhancing security protocols and modifying potential risks in dynamic environments is emphasized by this study. Future research should examine how different datasets may be integrated and how hybrid tactics that mix DL and conventional security techniques can be developed.

Additionally, by using adaptive learning strategies, the model's capacity to adjust to changing cyberthreats may be improved.All things considered, the study advances AI-powered cybersecurity solutions and establishes a strong basis for the efficient defense of digital infrastructures.

REFERENCES

[1] Li,Y.,&Liu,Q.(2021).Acomprehensivereviewstudyofcyber-attacks and cyber security; Emerging trends and recent developments. Energy Reports, 7, 8176-8186.

[2] Akgun,D.,Hizal,S., & Cavusoglu,U.(2022).A newDDoS 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-basedthreatdetectionin IIoT. IEEE Communications Magazine, 59(10), 76-82.