ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CYBER FORTIFICATIONS: ENSEMBLE SECURITY THROUGH DEEP LEARNING INNOVATIONS

Y. SUDHEER KUMAR [1], DR. A MARY SOWJANYA [2]

[11]Research Scholar, Andhra University, Vizag, Andhra Pradesh, India.
[2]Associate Professor, Andhra University, Andhra Pradesh, India.

E-mail: [1]kanna.sudheer8@gmail.com, [2] dr.amsowjanya@andhrauniversity.edu.in

ABSTRACT

Cyber threats using traditional approaches are more complex to detect in the early stages. Cybersecurity is the domain that provides appropriate defense against modern attacks. Cyber threats or cyber-attacks are attempts to destroy, thieve, alter, disable, or destroy data or applications using unauthorized access to a network, computer system, or digital device. Detection and classification of cyber attacks is a tedious task for state-of-art algorithms because of its more computation time and lack of understanding of attack patterns. Deep Learning (DL) is a domain used in many cyber threat detection systems, such as distributed denial-of-service (DDoS) attacks, phishing, ransomware, and anomalies. The pre-trained model Attack Pattern Convolutional Neural Networks (AP-CNNs) is introduced to detect attack patterns from the cyber attacks dataset. This paper introduces an Ensemble Security Model (ESM) to detect and classify cyber attacks from benchmark datasets. ESM combines Deep Neural Network (DNN) and Adaptive Support Vector Machine (ASVM). DNN is used as a vital feature extraction that extracts significant attack patterns. ASVM is used to classify different types of attacks obtained from other datasets. This paper uses two benchmark datasets to measure the strength of the ESM. The pre-trained model Transfer Learning with a BERT-based Model is used to train on given datasets. Finally, the results show that the proposed approach obtained better results than existing models.

Keywords: *Distributed Denial-of-Service (DDOS) Attacks, Phishing, Ransomware, and Anomalies, Ensemble Security Model (ESM), Deep Neural Network (DNN).*

1. INTRODUCTION

Nowadays many companies are rapidly digitized to prevent several attacks, cybersecurity has become hard to manage and complex. As interconnected systems, Internet of Things (IoT) devices, and cloud-based solutions have proliferated, the attack surface available to malicious actors has grown exponentially. Therefore, these growing complexity demands new security mechanisms that can identify and counteract threats before they become an issue while also evolving with the dynamics of cyberattacks. Deep learning (DL) is a domain of artificial intelligence (AI) that is revolutionizing the world in many domains, such as computer vision, natural language processing, healthcare, and more. Its ability to analyze enormous datasets, identify patterns, and make predictions with astonishing precision has led to it being described as a game-changing advancement in cybersecurity. Deep learning, a subset of machine learning (ML), can be applied to cyber security tasks to create more advanced systems that detect, classify, and respond to cyber threats in real-time.

Ensemble learning can overcome such limitations to mitigate the constraints of single deep learning models within cybersecurity, making it a tangible solution as it constructs the predictive capabilities of multiple models to obtain an accurate and robust prediction. Ensemble models can better detect and mitigate cyber threats by combining different DL architectures and their strengths. Adversarial attacks have become a prominent threat in the current cybersecurity landscape, and such enhancement protects the system against such attacks by preserving detection performance even under adverse conditions. In this paper, an advanced approach "Cyber Fortifications" focused on ensemble security strategies propelled by advancements in DL algorithms. It explores the challenges of modern cyber threats, the shortcomings of traditional security controls, and how deep learning ensembles can transform the landscape of threat detection and mitigation. This study mainly consider recent developments, use cases, and prospects to present a holistic overview of how ensemble DL models can strengthen systems in the domain of cyber security to protect required infrastructures in the digital age.

Phishing attackers send fake emails or messages that look legitimate, tricking users into following a link and giving up sensitive information such as passwords or credit card numbers. Malicious software (malware) like viruses, worms, ransomware, or spyware is installed on a device without the user's consent. The malware encrypts the victim's data, and the attackers demand money (ransom) to restore access. Attackers bombard a system, server, or network with too much traffic to make it unavailable to legitimate users.

In this work, the proposed approach mainly focused on detecting attack patterns accurately because of the combined approach. It employs various deep learning (DL) models, such as DNN and ASVM, to increase detection performance and mitigate the impact of adversarial attacks. Ensemble methods leverage the strengths of multiple models to significantly improve threat detection rates while greatly reducing false positives. One of the inputs to this innovation is adversarial attacks. These attacks change the input to give misleading information to the ML model.

2. LITERATURE SURVEY

Wang et al. [14] introduced the IDS model that enhances the DBN approach. The neural network is a training model with back propagation (BP) and fixed metrics that randomly initialize the weights and thresholds. The proposed BP is combined with KELM to address poor classification outcomes. Finally, the proposed approach achieved high classification results based on acc-98.60%, P-90.21%, Recall-98.73%, 96.31%, and time-134 for KDD-cup99. Moghanian et al. [15] presented the new IDS model that detects network infiltration behaviors. The proposed approach uses the ANN for intrusion detection and a swarm-based technique that reduces intrusion detection issues. The proposed GOA integrated with ANN reduced the error rate in terms of intrusion detection. Finally, the results for KDD Dataset are acc-95.41%, Sp-93.1%, and Sen-89.25%, whereas for UNSW Dataset they are acc-98.88%, Sp-98.09%, and Sen-98.14%, respectively. An innovative classifier that combines ML and NLP was presented by Tsinganos et al. [16]. To identify Cialdini's persuasion notions, CNN is utilized as a training model that starts with chat-based social engineering. By creating a probability distribution among the sentence classes acting as a persuasion container, the suggested method categorizes the network that considers the phrase a persuasive payload. Following the CSE-PUC with its own-trained word embedding representation (62.2%) and the CSE-PUC with randomly initialized word embeddings (the lowest accuracy), the proposed approach achieves the second-highest accuracy (66.4%). Chakkaravarthy et al. [17] introduced a novel robust IDH. IDH comprises the honey folder, Audit Watch, and CEP. The honey folder is a decoy folder modeled after SoLA, explicitly designed for attack and serving as an early warning system to alert the user to strange file activity. AuditWatch is an Entropy module that measures the entropy of files and directories. The CEP engine combines data from many security systems to confirm ransomware behavior and attack patterns and respond quickly. The experimental evaluation demonstrates that the proposed IDH effectively limits ransomware operations with minimal data loss. The proposed approach can be enhanced to detect the ransomware attacks in the healthcare systems.

Jurecek et al. [18] presented the MDS that uses the PSO to specify the feature weights and measures the performance of several distance learning models used for performance analysis. These measurements learn the patterns that are considered in k-NN classification. Finally, the results show that the proposed MDS combined with the distance metric obtains a 1.09% error rate at 0.74% FPR, integrated with various ML algorithms applied. Conversely, the low FPR obtained a 1.15% error rate at a FPR of only 0.13%. Poudyal et al. [19] presented a new deep detection model that captures various features at several levels—the proposed approach combined with static and dynamic approaches that analyze the behavioral chains using AI techniques. Finally, the proposed approach obtains an accuracy of 99.72% with an FPR of 0.003.

Zhou et al. [20] presented a new learning-based framework used in cyber-physical systems (CPS) to detect sensor and actuator threats. The attacks are observed and detected using the threat-detection stage method. In abnormal situations, the system utilizes the abnormal feedback controller to establish the attacks, monitor the process, and increase the accuracy of the metrics. Dornala et al. [21] developed the ODDSP to evaluate the demands of cloud-based service users. The proposed approach offers high security to secure sensitive data throughout the data lifecycle. In cloud computing, the ODDSP is a significant advancement in meeting customers' changing data storage and retrieval demands while prioritizing security. Ponnapalli et al. [22] proposed a novel way to improve the security and effectiveness of data delivery in cloud computing. The advantage of this method shows the decentralized, secure, and transparent properties of the blockchain to set up a secure and open data distribution platform. Using a distributed ledger to record transactions in such a way that they cannot be changed is how data integrity is ensured in a blockchain. A series of experiments enable evaluation of the solution and demonstrate successful improvement of the security, transparency and efficiency of cloud-based data delivery. The results reveal a significant reduction in vulnerabilities, increased data integrity, and an overall improvement in system performance. Ponnapall et al. proposed a unique HLM (Human-Based Learning Model) to identify cloud attacks [23]. The proposed HLM achieves higher accuracy on detecting attacks and reducing false positive rate by taking advantage of supervised as well as unsupervised learning mechanisms. This proposed technique uses IIDS and RF, as the training model, and transfer learning with iterative reinforcement learning-based SVM to classify the various assaults. The proposed Hybrid Learning Model compares well with the existing approaches in the literature regarding accuracy in detecting attacks while reducing false positives. This study supports further initiatives to safeguard vital data assets and improve cloud computing infrastructure security. A unique load-balancing strategy designed for edge applications enabled by blockchain was put out by Dornala et al. [24]. The resource maximizes the utilization across edge devices while preserving the integrity and transparency offered by blockchain; the suggested approach integrated an intelligent load distribution system based on machine learning algorithms. We demonstrate via experimental assessment in edge computing contexts that the proposed solution substantially outperforms traditional blockchain-based approaches in terms of throughput, latency, and resource utilization. By enabling safe, scalable, and effective implementations of blockchains on the edge, this study can be a stepping stone to the strong and resilient decentralized systems of the coming era.

3. PRE-TRAINED MODEL ATTACK PATTERN CONVOLUTIONAL NEURAL NETWORKS (AP-CNNS)

Convolutional Neural Networks (CNNs) is the deep learning pre-trained model that provided outstanding performance in not only image processing but natural language processing (NLP) and pattern recognition tasks as well. Recently, researchers have modified CNNs for cybersecurity to classify network intrusions and detect attack patterns from the network traffic data. However, training a CNN model from scratch requires considerable labeled data and high computational resources; therefore, many cybersecurity applications do not use it. The transfer learning techniques have addressed these challenges, with pre-trained AP-CNNs being one possible solution. Typically, two steps are used to train AP-CNNs; first, researchers train them using widely available datasets such as ImageNet or domain-specific datasets such as network (i.e., Background) traffic datasets, and then they fine-tune them on smaller (i.e., Foreground) cyber attack datasets. It dramatically shortens the time required to train the model and improves its effectiveness in extracting discriminative features from network traffic information. AP-CNNs leverage the learned representations of the general-purpose data domain to generalize and enhance detection accuracy between different types of attacks, encompassing denial of service (DoS), distributed denial of service (DDoS), phishing, and malware attacks. The current work introduces using AP-CNNs to detect and classify cyber-attacks. Experiments on several benchmark cyber attack datasets are performed using different pre-trained CNN architectures, including VGG-16 and ResNet-50, and fine-tuning methods. Figure 1 shows the layers of the pre-trained model.

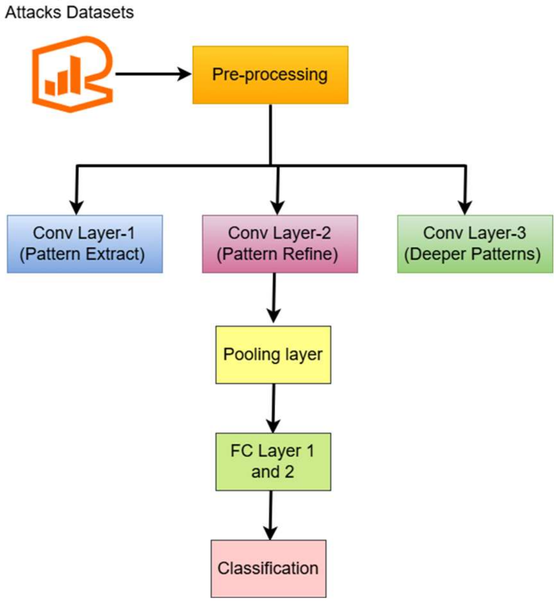

Attacks Datasets
Pre-processing
Conv Layer-1 (Pattern Extract)
Conv Layer-2 (Pattern Refine)
Conv Layer-3 (Deeper Patterns)
Pooling layer
FC Layer 1 and 2
Classification

Figure 1: Layers in Pre-trained Model

This allows for a strong platform to enhance the methods used for detecting cyberattacks, especially those based on CNN. Pre-trained Attack Patterns CNNs (AP-CNNs) can also be used as a model for new models that aim to detect a cyberattack, resulting in a more efficient model since the pre-trained model has already learned to recognize specific malicious patterns. Specifically, there are transferring, in which the convolutional layers learned by AP-CNN are transferred to the proposed model to share the feature extraction knowledge. Such transfer of layers that already have learned to recognize hierarchical traits from the primitives' levels to the high-end attack signatures can be fine-tuned or frozen based on the closeness of the datasets. This reduces training time, providing room for the model to adapt to unseen attacks, including zero-day attacks. Transfer learning makes it possible to construct more effective and generalized cyberattack identification models, which is due to the multiplicity applied in the AP-CNNs.

4. ENSEMBLE SECURITY MODEL (ESM)

The need to secure strong cybersecurity is more critical now than ever in the age of interconnectedness, where every connection opens the door for more possible breaches. As cyber threats become more sophisticated, traditional security mechanisms struggle to detect and classify complex attack patterns. In this scenario, the Ensemble Security Model (ESM), a novel approach combining the advantages of DNN and ASVM, is introduced to overcome those issues. It allows them to learn hierarchical representations of data, which suits feature extraction and anomaly detection very well. Detecting cyber-attacks is a crucial application of DNNs in cybersecurity, as they can capture minute variations and irregularities indicating possible attacks. It allows ESM to act as a powerful detection device with a high-dimensional and complex dataset.

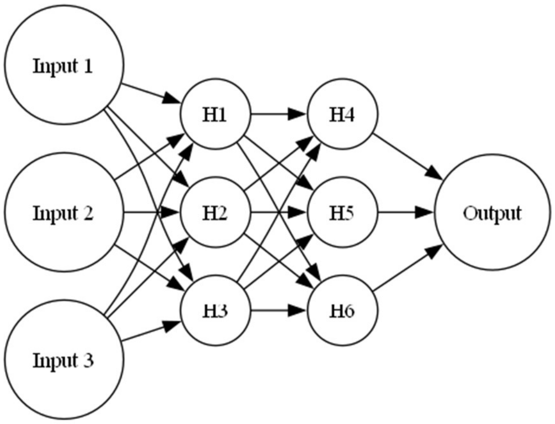


Figure 2: Architecture Diagram for DNN to Detect Cyber Attacks

Figure 2 explains the integration of DNN and ASVM. The proposed ASVM is the advanced version of SVM that improves the detection of dynamic data patterns on the selected datasets. This architecture allows for a clear separation between any types of cyber threats. The ESM uses ASVMs to accurately classify the detected threats as benign or one of the predefined attack classes; this helps to improve the decision-making process. The Ensemble Security Model integrates DNNs for attack detection and ASVMs for classification, capturing each approach's complementary necessities and characteristics. Combining these methods allows for better system functionality in detecting and preventing cyber threats, even in dynamic environments. ASVMs can adapt from one attack pattern to another and not lose their effectiveness on newer, actual world attack patterns.

The following steps show the equations that implement the DNN algorithm:

Step 1: Linear Transformation:

$$c^{(l)} = W^{(l)} \cdot x^{(l-1)} + y^{(l)} \qquad (1)$$

$c^{(l)}$: Linear output of layer l.
$W^{(l)}$: Weight matrix for layer l.
$x^{(l-1)}$: Activations from previous layer.
$y^{(l)}$: Bias vector for layer l.

Step 2: Activation Function:

$$x^{(l)} = \sigma\left(c^{(l)}\right) \qquad (2)$$

σ: ReLU function.
$x^{(l)}$: Activations at layer l.

Step 3: Classification using Cross-Entropy:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} b_{i,k} \log\left(\hat{b}_{i,k}\right) \qquad (3)$$

N: No of Samples.
K: No of output classes.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$b_{i,k}$: True label for sample i and class k.
$\hat{b}_{i,k}$: Predicted probability for sample I and class k.
Step 4: Weighted Updates based on Back-propagation:

$$\mathrm{Wei}^{(l)} \leftarrow \mathrm{Wei}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial \mathrm{W}^{(l)}} \qquad (4)$$

η: Learning rate.
$\frac{\partial \mathcal{L}}{\partial \mathrm{W}^{(l)}}$: Gradient of the loss with respect to $\mathrm{Wei}^{(l)}$

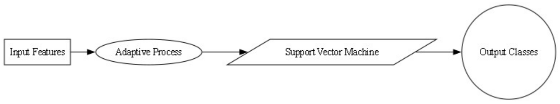


Figure 3: Architecture Diagram for Adaptive Support Vector Machines (ASVM) to Classify Detect Cyber Attacks

5. ADAPTIVE SUPPORT VECTOR MACHINE (ASVM)

ASVM improves the default SVM by adapting the modifications in data based on the type of the attack dynamically.
Step 5: Decision Function:

$$\mathrm{f(a)} = \mathrm{sign}\left(\sum_{i=1}^{N} \alpha_i b_i K(a_i, a) + y\right) \qquad (5)$$

a: Input data.
a_i: Support vectors.
α_i: Lagrange multipliers.
b_i: True labels for support vectors.
$K(a_i, a)$: Kernel function.
y: Bias term.
Step 6: Objective Function:

$$\min_{\alpha} \frac{1}{2} \sum_{a,b} \alpha_a \alpha_b b_a b_b K(m_a, m_b) - \sum_{a} \alpha_a \qquad (6)$$

Subject to: $0 \leq \alpha_a \leq C$, $\sum_a \alpha_a b_a = 0$
C: Regularization metric that controls the border width and mismatching penalties.
Step 7: Adaptation Technique:

$$\Delta W = -\eta \frac{\partial \mathcal{L}_{ASVM}}{\partial W} + \curlywedge \Delta W_{previous} \qquad (7)$$

ΔW: Weight adjustment.
$\curlywedge$: Stability momentum.
$\mathcal{L}_{ASVM}$: Adaptive loss function that consider present and historical data.
Step 8: Kernel Update for Dynamism: If $K(a_i, a)$ is adaptively updated, and it is represented as:

$$K'(a_i, a) = \gamma K(a_i, a) + (1 - \gamma) K_{new}(a_i, a) \qquad (8)$$

γ: Weight for prior kernel.

K_{new}: Updated kernel represents the new attack patterns.

6. DATASET DESCRIPTION

NSL-KDD: An improved KDD Cup 1999 dataset, removing redundant records, is most widely used for network intrusion detection research. The NSL-KDD cup contains 126k training records and 23k testing records. This dataset consists of 22 intrusion attacks and 41 attributes.

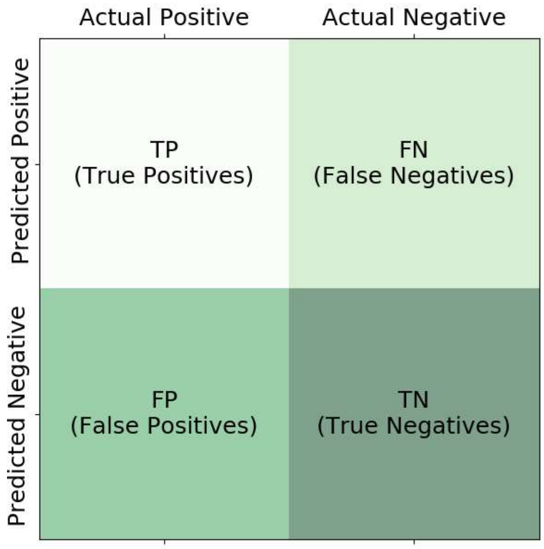
CICIDS2017: This represents realistic network traffic and includes labeled attack scenarios like brute force, DoS, Heartbleed, botnet, DDoS, classification of attack, and so on. The training dataset contains 25k and testing dataset 10k with two classes such as normal and anomaly data. It contains 80 attributes (features) that are used for processing of input data.

DARPA Intrusion Detection Dataset: A seminal dataset for identifying intrusions. Includes TCP dump data collected over weeks. This dataset contains 400k records with 41 attributes and one (1) class that reflects different types of attacks. In this 400k dataset, the 100k is training and 300k is testing.

The IoT-23 dataset: The IoT-23 dataset is a network traffic dataset for Internet of Things (IoT) devices and IoT-related security studies. It is extensively used for conducting research in different fields, such as intrusion detection systems (IDS), IoT network security, and behavior analysis of IoT devices. This dataset contains multiple heterogeneous IoT scenarios capturing network traffic. Each scenario includes traffic from IoT devices in both simulated benign conditions (regular traffic) and during simulated attacks (malicious traffic). Attributes are typically extracted from either network flows or packets using tools such as Zeek (formerly Bro), Wireshark, or other packet analysis frameworks. This dataset contains 100k records of different types of attacks along with 100 attributes. The training includes 95k, and testing contains 5k.

7. PERFORMANCE METRICS

In this section, the proposed system's performance is measured using the confusion matrix and its attributes, as shown in Figure 4. The Python programming language is used to design algorithms for popular libraries such as Numpy, Pandas, and other ML-based libraries.



	Actual Positive	Actual Negative
Predicted Positive	TP (True Positives)	FN (False Negatives)
Predicted Negative	FP (False Positives)	TN (True Negatives)

Figure 4: Confusion Matrix Diagram

$$\text{Accuracy (ACC)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Specificity (Spc)} = \frac{\text{No of TN}}{\text{No of TN} + \text{No of FP}}$$

$$\text{Recall (Re)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1} - \text{Score (F1S)} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

7.1 Results and Discussions

In this section, the quantitative performance of algorithms explained with comparison between various algorithms. The three algorithms such as Auto-encoder, DT-PCA, and DNN-ASVM where applied on four benchmark cyber attacks datasets.

Table 1: Quantitative Performance Comparison between Algorithms on NSL-KDD

	Auto-encoder	DT-PCA	DNN-ASVM
Accuracy	78.23	88.98	97.89
Precision	76.23	86.12	98.34
Specificity	77.23	88.56	98.76
Recall	78.45	87.34	98.01
F1-Score	74.56	85.34	98.32

The performance of different algorithms applied on NSL-KDD dataset like DNN-ASVM (97.89%) is better than DT-PCA (88.98%) and Auto-encoder (78.23%) based on Table 1 in terms of Overall Accuracy. The precision of DNN-ASVM (98.34%) is significantly higher than the DT-PCA (86.12%) and Auto-encoder(76.23%), suggesting that DNN-ASVM performs much better than DT-PCA and Auto-encoder in avoiding false positives. DNN-ASVM (98.76%) also has the best specificity, indicating that DNN-ASVM is better at predicting negative samples instead of DT-PCA (88.56%) and Auto-encoder (77.23%). Once again, DNN-ASVM (98.01%) is leading, followed by DT-PCA (87.34%) and Auto-encoder (78.45%). DNN-ASVM (98.32%) outperforms the rest of the algorithms; on the contrary, DT-PCA (85.34%) and Auto-encoder (74.56%) perform worst.

*Table 2: Quantitative Performance Comparison between Algorithms on **CICIDS2017***

	Auto-encoder	DT-PCA	DNN-ASVM
Accuracy	80.11	90.12	98.12
Precision	77.86	87.34	99.12
Specificity	78.56	89.12	99.76
Recall	79.45	88.31	98.01
F1-Score	76.9	87.34	98.32

As shown in Table 2, the performance of different algorithms on CICIDS2017 dataset which is presented and the proposed approach achieves 98.12% accuracy DNN-ASVM, which indicates the most accurate model. The precision is 99.12%, which generates very few false positives and vice versa. High values are achieved by the specificity (99.76%). In contrast, DNN-ASVM delivers 98.01%, indicating it's the best at detecting positive cases out of DT-PCA and Auto-encoder. Lastly, the F1 measure (98.32%) is to say that it has a well-balanced and high performance in terms of both precision and recall.

*Table 3: Quantitative Performance Comparison between Algorithms on **DARPA Intrusion Detection Dataset***

	Auto-encoder	DT-PCA	DNN-ASVM
Accuracy	81.23	91.98	98.89
Precision	80.23	88.12	99.51
Specificity	79.23	89.56	99.91

Recall	79.45	91.34	99.01
F1-Score	76.56	89.34	99.32

As shown in Table 3, the performance of different algorithms on ***DARPA*** dataset which is presented and the proposed approach achieves 98.89% accuracy DNN-ASVM, which indicates the most accurate model. The precision is 99.51%, which generates very few false positives and vice versa. High values are achieved by the specificity (99.91%). In contrast, DNN-ASVM delivers 99.01%, indicating it's the best at detecting positive cases out of DT-PCA and Auto-encoder. Lastly, the F1 measure (99.32%) is to say that it has a well-balanced and high performance in terms of both precision and recall.

*Table 4: Quantitative Performance Comparison between Algorithms on **IoT-23 Dataset***

	Autoencoder	DT-PCA	DNN-ASVM
Accuracy	82.41	90.98	99.12
Precision	81.23	92.12	99.39
Specificity	81.28	93.67	99.34
Recall	82.15	91.21	99.11
F1-Score	79.16	90.45	99.12

Table 4, shows the performance of different algorithms on ***IoT-23*** dataset which is presented and the proposed approach achieves 99.12% accuracy DNN-ASVM, which indicates the most accurate model. The precision is 99.39%, which generates very few false positives and vice versa. High values are achieved by the specificity (99.34%). In contrast, DNN-ASVM delivers 99.11%, indicating it's the best at detecting positive cases out of DT-PCA and Auto-encoder. Lastly, the F1 measure (99.12%) is to say that it has a well-balanced and high performance in terms of both precision and recall.

8. CONCLUSION

The Ensemble Security Model (ESM) proposed utilizes DNN's feature extraction and representation learning capacity considerably with the classification power of ASVM. This hybrid methodology has outperformed the earlier methodologies, especially for detecting and classifying different cyber attacks throughout the benchmark datasets. The proposed approach, DNN-ASVM, is able to adopt more significant features that are dynamically evolving patterns of attacks. Its scalability to high-dimensional data makes the model very practical for applications used in real-world cyber security scenarios. The experimental results show that ESM can attain superior performance compared to traditional single-model methods. Additionally, its ensemble structure provides a trade-off between computational resources and prediction accuracy, making it an appropriate choice for current, large-scale cybersecurity needs. The part of this research is interpretability because ensemble models are generally black box models and it is hard to understand the rationale behind their predictions. Even though these models do provide strong defense mechanisms against adversarial techniques, they are still susceptible to protect against adversarial attacks when highly forcing perturbation is adopted. In the future, we will elaborate on the model by synthesizing real-time detection potential, adding other features for the processing of multi-modal data, and using interpretability techniques through explainability to enhance the interpretability and trustworthiness of the model by security analysts.

REFERENCES:

[1] B. M. Coronado, U. Mori, A. Mendiburu, and J. Miguel-Alonso, "Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process," IEEE Trans. Netw. Service Manage., early access, Aug. 12, 2020, doi: 10.1109/TNSM.2020.3016246.

[2] Z. Ma, H. Ge, Y. Liu, M. Zhao, and J. Ma, "A combination method for Android malware detection based on control flow graphs and machine learning algorithms," IEEE Access, vol. 7, pp. 21235–21245, 2019.

[3] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," IEEE Access, vol. 6, pp. 35365–35381, 2018.

[4] D. Stiawan, M. Y. Bin Idris, A. M. Bamhdi, and R. Budiarto, "ICIDS2017 dataset feature analysis with information gain for anomaly detection," IEEE Access, vol. 8, pp. 132911–132921, 2020.

[5] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A taxonomy of network threats and the effect of

current datasets on intrusion detection systems," IEEE Access, vol. 8, pp. 104650–104675, 2020.

[6] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, 2019.

[7] J. Lee, J. Kim, I. Kim, and K. Han, "Cyber threat detection based on artificial neural networks using event profiles," IEEE Access, vol. 7, pp. 165607–165626, 2019

[8] Y. Zhang, P. Li, and X. Wang, "Intrusion detection for IoT based on improved genetic algorithm and deep belief network," IEEE Access, vol. 7, pp. 31711–31722, 2019.

[9] P. Feng, J. Ma, C. Sun, X. Xu, and Y. Ma, "A novel dynamic Android malware detection system with ensemble learning," IEEE Access, vol. 6, pp. 30996–31011, 2018.

[10] V. Hnamte, H. Nhung-Nguyen, J. Hussain and Y. Hwa-Kim, "A Novel Two-Stage Deep Learning Model for Network Intrusion Detection: LSTM-AE," in IEEE Access, vol. 11, pp. 37131-37148, 2023, doi: 10.1109/ACCESS.2023.3266979.

[11] F. Ullah et al., "Cyber Security Threats Detection in Internet of Things Using Deep Learning Approach," in IEEE Access, vol. 7, pp. 124379-124389, 2019, doi: 10.1109/ACCESS.2019.2937347.

[12] Y. Zhou, K. G. Vamvoudakis, W. M. Haddad and Z. -P. Jiang, "A Secure Control Learning Framework for Cyber-Physical Systems Under Sensor and Actuator Attacks", IEEE Transactions on Cybernetics, vol. 51, no. 9, pp. 4648-4660, Sept. 2021.

[13] Y. Tang and C. Li, "An Online Network Intrusion Detection Model Based on Improved Regularized Extreme Learning Machine", IEEE Access, vol. 9, pp. 94826-94844, 2021.

[14] Z. Wang, Y. Zeng, Y. Liu and D. Li, "Deep Belief Network Integrating Improved Kernel-Based Extreme Learning Machine for Network Intrusion Detection", IEEE Access, vol. 9, pp. 16062-16091, 2021.

[15] S. Moghanian, F. B. Saravi, G. Javidi and E. O. Sheybani, "GOAMLP: Network Intrusion Detection With Multilayer Perceptron and Grasshopper Optimization Algorithm", IEEE Access, vol. 8, pp. 215202-215213, 2020.

[16] N. Tsinganos, I. Mavridis and D. Gritzalis, "Utilizing Convolutional Neural Networks and Word Embeddings for Early-Stage Recognition of Persuasion in Chat-Based Social Engineering Attacks", IEEE Access, vol. 10, pp. 108517108529, 2022.

[17] Chakkaravarthy, S. Sibi et al., "Design of intrusion detection honeypot using social leopard algorithm to detect IoT ransomware attacks", IEEE Access, vol. 8, no. 2020, pp. 169944-169956.

[18] M. Jurecek and R. Lorencz, "Application of Distance Metric Learning to Automated Malware Detection", IEEE Access, vol. 9, pp. 96151-96165, 2021.

[19] S. Poudyal and D. Dasgupta, "Analysis of Crypto-Ransomware Using ML-Based Multi-Level Profiling", IEEE Access, vol. 9, pp. 122532-122547, 2021.

[20] Y. Zhou, K. G. Vamvoudakis, W. M. Haddad and Z. -P. Jiang, "A Secure Control Learning Framework for Cyber-Physical Systems Under Sensor and Actuator Attacks", IEEE Transactions on Cybernetics, vol. 51, no. 9, pp. 4648-4660, Sept. 2021.

[21] Dornala, R.R., Ponnapalli, S., Koteru, R.R., Koteru, B. (2024). An On-demand Data Delivery and Secured Platform in Cloud Computing. In: Vimal, V., Perikos, I., Mukherjee, A., Piuri, V. (eds) Multi-Strategy Learning Environment. ICMSLE 2024. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-97-1488-9_6

[22] S. Ponnapalli, R. R. Dornala, K. Thriveni Sai and S. Bhukya, "A Secure and Smooth Data Delivery Platform with Block chain in Cloud Computing," 2024 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI), Lalitpur, Nepal, 2024, pp. 590-596, doi: 10.1109/ICMCSI61536.2024.00093.

[23] S. Ponnapalli, R. R. Dornala and K. T. Sai, "A Hybrid Learning Model for Detecting Attacks in Cloud Computing," 2024 3rd International Conference on Sentiment Analysis and Deep Learning (ICSADL), Bhimdatta, Nepal, 2024, pp. 318-324, doi: 10.1109/ICSADL61749.2024.00058.

[24] R. Dornala, S. Ponnapalli and K. Sai, "Blockchain Security in Edge Applications with Novel Load Balancing Approach," 2023 International Conference on Sustainable Communication Networks and Application (ICSCNA), Theni, India, 2023, pp. 263-269, doi: 10.1109/ICSCNA58489.2023.10370477.