Efficient Deep Neural Network for Intrusion Detection Using CIC-IDS-2017 Dataset

Gopichand Bandarupalli[1]

[1]ai.ml.research.articles@gmail.com

[1]Professional M.B.A., Campbellsville university, Texas, USA

***Abstract*— Intrusion Detection Systems (IDS) are essential in identifying and reporting potential network attacks. IDS are classified into Host-based IDS (HIDS), which monitor internal computer threats, and Network-based IDS (NIDS), which monitor network-level attacks. IDS can also function as anomaly-based, using Machine Learning (ML) and Deep Learning (DL) to recognize unfamiliar attack patterns, or as rule-based, which relies on historical data-driven rules. Notably, anomaly-based IDS can detect zero-day attacks. Therefore, this study proposes a Deep Neural Network (DNN) model for anomaly-based IDS, focusing on accurate and efficient attack detection and categorization using the CIC-IDS-2017 dataset. By optimizing dataset preprocessing, the DNN architecture aims to minimize computational demands while maintaining high accuracy. Comparative evaluation with other models demonstrates the proposed model's effectiveness in attack detection despite using a simpler, more lightweight architecture than those in other studies, where more complex, less efficient approaches are often employed.**

***Index Terms*— Anomaly, Attack, DL, DNN, IDS, ML**

I. Introduction

Intrusion Detection Systems (IDS) play a critical role in modern network security by monitoring traffic to identify potential malicious activities within a network [1], [2], [3], [4], [5], [6]. IDS come in two main forms: Network-based IDS (NIDS) and Host-based IDS (HIDS), each serving specific purposes in protecting networked systems. NIDS is typically implemented on a network device, such as a router or switch, to monitor and analyze network traffic. NIDS can employ either signature-based or anomaly-based detection methods to examine data packets in real-time, allowing it to flag potential attacks and alert administrators promptly. By analyzing traffic as it passes through network devices, NIDS can identify threats at an early stage, protecting the overall network structure. However, NIDS alone cannot detect host-specific anomalies, such as irregular file access or unusual system calls, making it complementary rather than comprehensive. Whereas HIDS focus on monitoring individual devices within the network, typically servers or other endpoints, for unusual behavior patterns or unauthorized access attempts. This could include detecting unexpected system calls, unusual access of sensitive files, or non-standard port usage. However, because HIDS operate on specific devices, they may fail to detect network-level attacks, such as Distributed Denial-of-Service (DDoS) attacks. HIDS can also operate using signature-based or anomaly-based methods, making them highly effective in identifying host-level threats. Signature-based IDS operate by comparing real-time data with known attack patterns derived from historical data. This method is effective in detecting attacks that match previously documented attack types and yields a low rate of false positives since it does not flag new but benign patterns as potential threats. However, signature-based systems struggle to identify zero-day attacks, which are novel and lack historical data patterns, potentially leading to false negatives. Consequently, these systems may fail to detect new attack vectors that pose risks to network security. In contrast, anomaly-based IDS leverage Machine Learning (ML) to detect both known and unknown attacks. By applying ML, these systems adaptively learn from network data and adjust their detection models to identify new threats dynamically, making them more effective in identifying zero-day attacks. However, anomaly-based systems tend to have a higher rate of false positives, as unfamiliar but legitimate patterns in network traffic may be mistakenly classified as malicious. This trade-off between adaptability and accuracy is central to the choice of IDS approach, especially in environments where new and evolving threats are prevalent. Therefore, for robust protection, many organizations deploy both NIDS and HIDS to cover the unique blind spots associated with each system. A combined approach ensures comprehensive security by detecting both network-level and host-specific threats.

Therefore, the primary objective of this study is to develop a Deep Neural Network (DNN) [7] model capable of effectively detecting and classifying various attack types in a network. Anomaly-based IDS are particularly well-suited for this purpose, as their use of ML allows them to detect novel attacks that signature-based systems might miss. This DNN aims to achieve high accuracy, ideally above 90%, while remaining lightweight enough to minimize latency in detection. Given the time-sensitive nature of IDS, reducing detection times without compromising accuracy is essential for real-world applicability. The DNN will be developed and tested within an experimental environment rather than a live

IDS platform, allowing for controlled tuning and performance evaluation. This approach enables the model to be refined and optimized for ideal performance before any potential deployment. The CIC-IDS2017 dataset was selected for training the model due to its comprehensive collection of real-world attack data and its extensive use in prior research, which provides a strong foundation for benchmarking and comparative analysis. One of the most significant challenges in cybersecurity today is the rapid evolution of attack strategies. Traditional signature-based IDS, reliant on pre-set rules and historical data, often fail to detect new or modified attacks. Consequently, such systems are prone to overlooking novel threats, leading to network breaches and data leaks. anomaly-based IDS provide a solution by shifting the focus from recognizing known attack patterns to identifying anomalies, which may signify a breach. When paired with a DNN, an anomaly-based IDS becomes even more powerful, as it can continually learn and improve its detection capabilities by analyzing network traffic. This adaptive learning approach enhances the system's ability to respond to new threats over time, strengthening network defense against emerging attacks. This study's model is specifically designed to detect and classify network anomalies, making it an ideal candidate for deployment in a network-based and anomaly-based IDS.

The study is as follows; the related works will be shown in the next section. The materials and methods are covered in Section III. The experimental analysis and results discussion are carried out in Section IV, and in Section V, we wrap up the study with some conclusions and future research.

II. Related Works

In IDS research, selecting an up-to-date dataset is crucial for effective DNN configurations. While the outdated KDD Cup 99[1] dataset is widely used, this study uses the CIC-IDS-2017[2] dataset, which includes current network traffic and an official split for streamlined training and testing. For instance, [8] used a 3-layer DNN using the ReLU activation function achieved optimal performance, outperforming Support Vector Machines (SVMs) and mitigating issues like the vanishing gradient problem. [9] used an advanced Hierarchical Spatial-Temporal features-based IDS (HAST-IDS) combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, enhancing accuracy by analyzing both spatial and temporal features in network traffic. [10] proposed Scale-Hybrid-Alertnet (SHIA), a hybrid IDS that combines NIDS and HIDS detection to identify malicious events. SHIA evaluates various machine learning methods and DNNs across multiple datasets, including KDD Cup 99 and CIC-IDS-2017, to determine optimal configurations. Utilizing distributed Deep Learning (DL) models and big data, SHIA achieves high accuracy without requiring specialized hardware and stores collected data in a NoSQL database. In comparison, [11] proposed Distributed IDS (DIDS) using blockchain for secure, decentralized logging of incidents, ensuring data integrity through consensus protocols, though it remains vulnerable to theoretical 51% attacks. Unlike SHIA, which integrates both NIDS and HIDS, this blockchain-based method focuses on cloud-based NIDS analysis, providing scalability but lacking SHIA's comprehensive detection capabilities. Additionally, [12] proposed a framework that divides detection and data processing into specific tasks. Agents perform packet sniffing, filtering, categorization, and storage within a Hadoop DFS environment, with reports generated by an agent manager. This framework leverages big data analytics to detect intrusions. Other frameworks, such as FESVDF [13] and HAST-IDS [14], use specialized DNNs, including CNN and LSTM, to handle spatial and temporal traffic features separately. FESVDF, in particular, demonstrates impressive accuracy, although metrics vary by dataset, and its performance slightly surpasses HAST-IDS [15], [16], [17], [18], [19], [20], [21].

III. Materials and Methods

A. Dataset Analysis

The CIC-IDS-2017 dataset is a modern, comprehensive dataset designed to support the development of IDS. It includes CSV files containing network traffic data, specifically capturing a variety of attack types such as DDoS and PortScan. This dataset addresses contemporary cyber threats absent in earlier datasets but contains challenges such as class imbalance, missing values, and high dimensionality that can impede model accuracy. To mitigate noise and improve model efficiency, feature extraction is critical; non-essential features are excluded as they can introduce unnecessary complexity. The preferred feature extraction method here is ANOVA[3], chosen for its high performance and reduced computational demands compared to alternatives like Recursive Feature Extraction (RFE). ANOVA effectively identifies features that most significantly impact model performance, optimizing storage and training times. Data normalization is also essential, and the Yeo-Johnson Power Transformation[4] is also employed due to its ability to handle both positive and negative values, addressing skewness and improving distribution over methods like MinMaxScaler. Given the dataset's imbalance, sampling methods are necessary. Synthetic Minority Oversampling Technique (SMOTE) is used to augment minority classes without overfitting, while Random Undersampling (RUS) balances the majority class. After feature selection, an optimal balance of 36 features is used, ensuring improved performance without excessive complexity. In developing an attack detection and classification model, key metrics like accuracy, precision, recall, and F1-score are chosen for their relevance to model evaluation. Accuracy measures correct predictions, while precision, recall, and F1-score provide nuanced insight into positive classification performance. Specificity is included to assess misclassification rates, and confusion matrices offer a detailed breakdown of results across attack types. The dataset, refined from raw data to reduce bias, excludes potentially influential features like IP addresses, Flow ID, and

[1] https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[2] https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset

[3] A statistical technique called ANOVA (Analysis of Variance) examines group means to ascertain whether differences are statistically significant.
[4] By managing both positive and negative values, the Yeo-Johnson Power Transformation normalizes data, enabling regression and enhancing model performance.

timestamps, ensuring better generalization. Data preprocessing includes cleaning, normalization, and feature selection using ANOVA to optimize model performance and avoid overfitting.

B. Model Analysis

The model for attack detection employs a ReLU activation function to address the vanishing gradient issue and a softmax function in the final layer to enhance classification. Initial tests evaluated varying dense network sizes, with callback functions included to mitigate overfitting by halting training upon detecting learning plateaus as shown in Table I. The primary architecture utilized four hidden layers with an input size of 36 units and progressively smaller layer sizes, with dropout layers set to 0.2 between each layer except the input. This structure demonstrated satisfactory results, with high accuracy and recall rates across most classes, although some classes—such as DoS Hulk and Web Attack Brute Force—experienced challenges with false negatives. The confusion matrix revealed common challenges for DNNs in detecting benign packets without mistaking them for attacks, an issue often encountered in anomaly-based IDSs. Upon evaluating an alternative architecture with increased units per layer, resulting in a total of 52,704 trainable parameters compared to the original 15,600, the model showed a decrease in performance. Although there was a slight improvement in the macro recall score, the model's accuracy, specificity, and F1 scores were lower than the initial architecture. The increased model complexity also led to higher rates of false positives and false negatives in several classes, indicating diminished performance and questioning the utility of the expanded parameter count.

TABLE I
PERFORMANCE METRICS WTH DIFFERENT ARCHITECTURE

Architecture Name	Accuracy	Specificity	Recall Macro	Recall Micro	Recall Weighted	Precision Macro	Precision Micro	Precision Weighted	F1-Score Macro	F1-Score Micro	F1-Score Weighted
Initial Architecture	97.6%	0.9825	0.86367	0.976	0.976	0.6491	0.976	0.9943	0.672	0.9976	0.9844
Second Architecture	97.34%	0.9823	0.9061	0.97355	0.97355	0.6431	0.9735	0.9928	0.6653	0.9735	0.9923
Fourth Architecture	97.499%	0.9821	0.9092	0.97992	0.97992	0.661	0.974	0.9939	0.681	0.975	0.994
Fifth Architecture	97.34%	0.9816	0.9088	0.9734	0.9734	0.6289	0.9734	0.9933	0.6588	0.973	0.993

Further analysis explored an additional fifth hidden layer with varying units (256 to 32) as shown in Table II. While this demonstrated a slight recall improvement, overall accuracy decreased compared to the initial model. Increased false positives and negatives in certain classes, notably DoS Hulk and Web Attack Brute Force, revealed limitations in misclassification handling. The model's complexity yielded no significant performance benefit over the simpler initial model, highlighting diminishing returns from added layers. Further exploration introduced a six-layer architecture, adding another 512-unit layer. This model also failed to surpass the baseline model's accuracy and recall scores, further emphasizing the drawbacks of increasing network depth in this application. The architecture struggled with specificity, and the confusion matrix reflected poor classification of benign packets, suggesting the model could not sufficiently capture the benign patterns amid attack classes. Following the realization that deeper architectures did not enhance performance, the study evaluated simpler architectures by reducing layer sizes. By removing one of the hidden layers in the initial model, the model achieved improved specificity and a reduced false positive rate. Importantly, sensitivity and precision improved, counterbalancing an increase in false negatives for specific attack types. This reduction in network size resulted in a more balanced architecture with favorable performance in most metrics, despite the smaller parameter count.

TABLE II
PERFORMANCE METRICS WTH DIFFERENT ARCHITECTURE (REDUCED UNITS)

Architecture Name	Accuracy	Specificity	Recall Macro	Recall Micro	Recall Weighted	Precision Macro	Precision Micro	Precision Weighted	F1-Score Macro	F1-Score Micro	F1-Score Weighted
Initial Architecture	97.6%	0.9825	0.86367	0.976	0.976	0.6491	0.976	0.9943	0.672	0.9976	0.9844
Second Architecture	97.34%	0.9823	0.9061	0.97355	0.97355	0.6431	0.9735	0.9928	0.6653	0.9735	0.9923
Fourth Architecture	97.499%	0.9821	0.9092	0.97992	0.97992	0.661	0.974	0.9939	0.681	0.975	0.994
Fifth Architecture	97.34%	0.9816	0.9088	0.9734	0.9734	0.6289	0.9734	0.9933	0.6588	0.973	0.993
Improved Initial Architecture	97.62%	0.9842	0.858	0.9762	0.9762	0.6529	0.9762	0.9935	0.6716	0.9762	0.9840

IV. Experimental Analysis

A. Experimental Setup

This study evaluates multiple neural network architectures for optimizing an anomaly-based NIDS, assessing both smaller and larger models with variations in layers and units. ReLU was selected as the activation function due to its efficiency in reducing training time and preventing the vanishing gradient problem, while softmax was used for the output layer to manage the multi-class classification task. Dropout layers were applied throughout to improve model robustness and prevent overfitting. The Adam optimizer, chosen for its minimal configuration and adaptability, supported efficient backpropagation. In smaller architectures, initial tests favored a reduced baseline model, which exhibited lower loss values and higher specificity compared to the original model. The reduced model achieved notable gains in recall and a balanced F1 score, particularly effective in avoiding false positives—a key consideration for NIDS applications, where minimizing misclassification is critical. Although it produced more false negatives, the reduced baseline's simplicity and performance balance made it an effective choice. Larger architectures were evaluated next, including a model with doubled units and another with an added 512-unit layer. The fourth architecture, with an additional layer of 16 units, emerged as the most

balanced in terms of recall, precision, and F1 score, outperforming others in accuracy and loss function. Although this architecture showed slight improvements, it had limitations, notably an increased rate of false positives. Comparison of confusion matrices confirmed that the fourth model's high accuracy was offset by a greater complexity. Ultimately, the reduced baseline architecture was recommended for its optimal balance of simplicity and performance. While slightly less effective in precision metrics compared to the larger models, its reduced parameter count (15,000 versus over 53,000 in the fourth model) made it a more efficient solution for real-time intrusion detection without significant sacrifices in accuracy.

B. Result Analysis

In this section, the proposed improved initial architecture, chosen from various evaluated configurations, is benchmarked against several established architectures in the field as shown in Table III. Though direct comparisons are challenging due to differences in datasets and configurations, this analysis provides insights by examining key performance metrics, including accuracy, recall, precision, specificity, and F1-score. A key limitation in comparing the proposed architecture to others in literature is the dataset disparity; most of the reviewed architectures use different datasets, which impacts direct comparisons. Nevertheless, Macro scores from the proposed architecture have been selected to facilitate a fairer comparison, as these scores provide a balanced view of real-world performance. Various architectures outperform the proposed model in specific areas. For instance, AI-SIEM [22] employs multiple neural networks and extensive pre-processing to discriminate effectively between true and false positives, achieving superior specificity and recall. However, this approach also requires significantly more computational resources, training epochs (over 1,000 vs. 25 in the proposed model), and complex neural structures, including CNN, LSTM, and Fully-Connected Networks (FCNN), which introduce higher overheads. Despite the advanced performance, the simplicity and computational efficiency of the proposed model could still make it a viable option in time-sensitive settings. The HAST-IDS [14] framework combines CNN and LSTM architectures, producing impressive detection capabilities, with superior accuracy compared to the proposed architecture. Its hybrid approach efficiently leverages the strengths of both architectures to boost detection rates. However, HAST-IDS requires more resources, highlighting a trade-off between model complexity and efficiency that favors simpler architectures for faster, albeit slightly less accurate, performance. Another architecture, FESVDF [13], takes a modular approach targeting specific attack types, resulting in enhanced performance. Yet, its specialized design requires running multiple neural networks in tandem, potentially leading to slower detection times. While FESVDF demonstrates the advantage of modularity, the proposed model's monolithic nature enables streamlined processing at the cost of slightly lower detection accuracy. The GNP IDS [23] model, which combines rule-based and anomaly-based approaches, showcases excellent performance. Its mixed-method design effectively reduces false positives and false negatives, outperforming the proposed architecture in accuracy and recall. Still, deploying a simpler, high-performance architecture with fewer trainable parameters—as is the case with the proposed model—offers operational benefits, especially where computational constraints are a factor. Within the SGM-CNN [24] architecture, a CNN model paired with advanced sampling (SMOTE with under-sampling) and a clustering technique (GMM) improves feature extraction, leading to high detection accuracy. SGM-CNN trained with larger batches and more epochs than the proposed model, which likely contributed to its robust performance. However, the extensive training, combined with more complex models, highlights the computational overhead inherent in CNN architectures compared to simpler Feed-Forward Networks (FFNN) like the proposed one. Similarly, SGM-MLP [24], which uses Nadam as an optimizer, also performed well but required more training resources and a larger feature set (77 features vs. 36 in the proposed model). The DMLP IDS [25], which adopts a Multi-Layer Perceptron (MLP) design, aligns closely with the structure of the proposed architecture. The DMLP employs RFE to optimize feature selection, contrasting with the ANOVA feature selection used in the proposed architecture. While DMLP exhibits commendable performance, especially when using a reduced feature set, it falls short in scenarios requiring complex, multi-class classification, as it functions primarily as a binary classifier for DDoS attacks. Additionally, the DMLP model's lack of pre-processing, apart from RFE, likely affected its performance, underscoring the importance of data normalization and optimization in boosting detection efficiency. The SHIA [10] framework combines host-based and network-based IDS functionalities, making it a comprehensive detection system with significant adaptability across different types of threats. However, SHIA utilizes a more complex architecture, with larger input layers (77 neurons versus 36 in the proposed model) and smaller dropout rates, which help boost accuracy but at the expense of simplicity. The proposed architecture, on the other hand, maintains computational efficiency and faster processing, owing to its reduced complexity and fewer trainable parameters. SHIA's superior performance can largely be attributed to its higher number of features, extensive training epochs, and the reduced number of classes it needs to classify due to attack grouping.

TABLE III
AN OVERVIEW OF THE KEY INDICATORS FOR THE SUGGESTED ARCHITECTURE AND EARLIER RESEARCH

Name	Accuracy	Recall	Precision	F1-Score	Specificity	Dataset
Shallow DNN [26]	93%	0.915	0.997	0.955	–	KDD Cup 99
HAST-IDS [14]	99.89%	0.9696	–	–	0.98	CIC-IDS-2012
FESVDF [13]	99.65%	–	–	–	–	KDD Cup 99
GNP IDS [23]	90.26%	0.925	–	–	0.9866	KDD Cup 99
SHIA [10]	96.2%	0.962	0.972	0.965	–	CIC-IDS-2017
AI-SIEM [22]	98.97%	0.982	–	0.65	0.992	CIC-IDS-2017
SGM-CNN [24]	99.85%	0.9985	0.9988	0.9986	0.9869	CIC-IDS-2017
SGM-MLP [24]	–	0.9963	0.9976	0.9968	–	CIC-IDS-2017
DMLP IDS [25]	91%	–	–	–	–	CIC-IDS-2017
Proposed Architecture	**97.62%**	**0.8858**	**0.6529**	**0.6716**	**0.9842**	**CIC-IDS-2017**

V. Conclusion and Future Works

The proposed architecture is designed for efficiency, using a reduced dataset and avoiding grouped attacks to enhance speed without compromising performance. Comparison with other architectures, such as GNP for rule generation and AI-SIEM, reveals these models' superior performance due to their complex, multi-layered structures and higher feature count. The SHIA framework, with its focus on attack categorization and additional features, further highlights the benefits of a more intricate design. Despite these advantages, such architectures incur significant computational overhead, impacting real-time functionality. The simplicity of the proposed model offers a lightweight, versatile solution capable of detecting multiple attack types quickly. Future research should explore optimized parameters, alternative optimizers, and sampling methods, aiming for a balance between accuracy and computational efficiency. Evaluating models in real-world network environments could provide insights into optimal packet processing rates and further validate their practical viability.

VI. Declarations

A. ***Funding:*** No funds, grants, or other support was received.

B. ***Conflict of Interest:*** The authors declare that they have no known competing for financial interests or personal relationships that could have appeared to influence the work reported in this paper.

C. ***Data Availability:*** Data will be made on reasonable request.

D. Code Availability: Code will be made on reasonable request.

References

[1] S. Wazir, G. S. Kashyap, K. Malik, and A. E. I. Brownlee, "Predicting the Infection Level of COVID-19 Virus Using Normal Distribution-Based Approximation Model and PSO," Springer, Cham, 2023, pp. 75–91. doi: 10.1007/978-3-031-33183-1_5.

[2] G. S. Kashyap *et al.*, "Revolutionizing Agriculture: A Comprehensive Review of Artificial Intelligence Techniques in Farming," Feb. 2024, doi: 10.21203/RS.3.RS-3984385/V1.

[3] P. Kaur, G. S. Kashyap, A. Kumar, M. T. Nafis, S. Kumar, and V. Shokeen, "From Text to Transformation: A Comprehensive Review of Large Language Models' Versatility," Feb. 2024, Accessed: Mar. 21, 2024. [Online]. Available: https://arxiv.org/abs/2402.16142v1

[4] S. Naz and G. S. Kashyap, "Enhancing the predictive capability of a mathematical model for pseudomonas aeruginosa through artificial neural networks," *Int. J. Inf. Technol. 2024*, pp. 1–10, Feb. 2024, doi: 10.1007/S41870-023-01721-W.

[5] M. Kanojia, P. Kamani, G. S. Kashyap, S. Naz, S. Wazir, and A. Chauhan, "Alternative Agriculture Land-Use Transformation Pathways by Partial-Equilibrium Agricultural Sector Model: A Mathematical Approach," Aug. 2023, Accessed: Sep. 16, 2023. [Online]. Available: https://arxiv.org/abs/2308.11632v1

[6] S. Wazir, G. S. Kashyap, and P. Saxena, "MLOps: A Review," Aug. 2023, Accessed: Sep. 16, 2023. [Online]. Available: https://arxiv.org/abs/2308.10908v1

[7] G. BANDARUPALLI, "Advancing Smart Transportation via AI for Sustainable Traffic Solutions in Saudi Arabia," Nov. 2024, doi: 10.21203/RS.3.RS-5389235/V1.

[8] R. Shire, S. Shiaeles, K. Bendiab, B. Ghita, and N. Kolokotronis, "Malware Squid: A Novel IoT Malware Traffic Analysis Framework Using Convolutional Neural Network and Binary Visualisation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, Sep. 2019, pp. 65–76. doi: 10.1007/978-3-030-30859-9_6.

[9] M. Tubishat, N. Idris, L. Shuib, M. A. M. Abushariah, and S. Mirjalili, "Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection," *Expert Syst. Appl.*, vol. 145, p. 113122, May 2020, doi: 10.1016/j.eswa.2019.113122.

[10] 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, doi: 10.1109/ACCESS.2019.2895334.

[11] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*. Auerbach Publications, 2016. doi: 10.1201/b10867.

[12] S. Creese, M. Goldsmith, N. Moffat, J. Happa, and I. Agrafiotis, "CyberVis: Visualizing the potential impact of cyber attacks on the wider enterprise," in *2013 IEEE International Conference on Technologies for Homeland Security, HST 2013*, 2013, pp. 73–79. doi: 10.1109/THS.2013.6698979.

[13] S. Zaman and F. Karray, "Lightweight IDS based on features selection and IDS classification scheme," in *Proceedings - 12th IEEE International Conference on Computational Science and Engineering, CSE 2009*, 2009, pp. 365–370. doi: 10.1109/CSE.2009.180.

[14] W. Wang *et al.*, "HAST-IDS: Learning Hierarchical Spatial-Temporal Features Using Deep Neural Networks to Improve Intrusion Detection," *IEEE Access*, vol. 6, pp. 1792–1806, Dec. 2017, doi: 10.1109/ACCESS.2017.2780250.

[15] F. Alharbi, G. S. Kashyap, and B. A. Allehyani, "Automated Ruleset Generation for 'HTTPS Everywhere': Challenges, Implementation, and Insights," *Int. J. Inf. Secur. Priv.*, vol. 18, no. 1, pp. 1–14, Jul. 2024, doi: 10.4018/IJISP.347330.

[16] N. Marwah, V. K. Singh, G. S. Kashyap, and S. Wazir, "An analysis of the robustness of UAV agriculture field coverage using multi-agent reinforcement learning," *Int. J. Inf. Technol.*, vol. 15, no. 4, pp. 2317–2327, May 2023, doi: 10.1007/s41870-023-01264-0.

[17] H. Habib, G. S. Kashyap, N. Tabassum, and T. Nafis, "Stock Price Prediction Using Artificial Intelligence Based on LSTM– Deep Learning Model," in *Artificial Intelligence & Blockchain in Cyber Physical Systems: Technologies & Applications*, CRC Press, 2023, pp. 93–99. doi: 10.1201/9781003190301-6.

[18] G. S. Kashyap, A. Siddiqui, R. Siddiqui, K. Malik, S. Wazir, and A. E. I. Brownlee, "Prediction of Suicidal Risk Using Machine Learning Models," Dec. 25, 2021. Accessed: Feb. 04, 2024. [Online]. Available: https://papers.ssrn.com/abstract=4709789

[19] G. S. Kashyap, K. Malik, S. Wazir, and R. Khan, "Using Machine Learning to Quantify the Multimedia Risk Due to Fuzzing," *Multimed. Tools Appl.*, vol. 81, no. 25, pp. 36685–36698, Oct. 2022, doi: 10.1007/s11042-021-11558-9.

[20] F. Alharbi and G. S. Kashyap, "Empowering Network Security through Advanced Analysis of Malware Samples: Leveraging System Metrics and Network Log Data for Informed Decision-Making," *Int. J. Networked Distrib. Comput.*, pp. 1–15, Jun. 2024, doi: 10.1007/s44227-024-00032-1.

[21] G. S. Kashyap *et al.*, "Detection of a facemask in real-time using deep learning methods: Prevention of Covid 19," Jan. 2024, Accessed: Feb. 04, 2024. [Online]. Available: https://arxiv.org/abs/2401.15675v1

[22] 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, doi: 10.1109/ACCESS.2019.2953095.

[23] Y. Gong, S. Mabu, C. Chen, Y. Wang, and K. Hirasawa, "Intrusion detection system combining misuse detection and anomaly detection using genetic network programming," in *ICCAS-SICE 2009 - ICROS-SICE International Joint Conference 2009, Proceedings*, 2009, pp. 3463–3467. Accessed: Nov. 09, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5335129

[24] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," *Comput. Networks*, vol. 177, p. 107315, Aug. 2020, doi:

10.1016/j.comnet.2020.107315.

[25] S. Ustebay, Z. Turgut, and M. A. Aydin, “Intrusion Detection System with Recursive Feature Elimination by Using Random Forest and Deep Learning Classifier,” in *International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, IBIGDELFT 2018 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Jan. 2019, pp. 71–76. doi: 10.1109/IBIGDELFT.2018.8625318.

[26] V. K. Rahul, R. Vinayakumar, K. Soman, and P. Poornachandran, “Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security,” in *2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018*, Institute of Electrical and Electronics Engineers Inc., Oct. 2018. doi: 10.1109/ICCCNT.2018.8494096.