International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

Intelligent Intrusion Detection Framework Using Rule-Guided and Deep Learning Techniques

Md Ashique Hussain[1], Mohammed Rayan Raheem Khan[2], Mehsan Bin Saleh[3], Mohammed Omer Hussain[4]

[1]Assistant Professor, Department of CSE (Data Science),
Lords Institute of Engineering and Technology, Hyderabad, Telangana, India.

[2,3,4]UG Students, Department of CSE (Data Science),
Lords Institute of Engineering and Technology, Hyderabad, Telangana, India.

***Abstract*— This work introduces an intelligent intrusion detection framework that combines rule-guided preprocessing and advanced machine learning techniques to accurately identify and classify network attacks using the NSL-KDD dataset. The system provides an interactive interface covering all essential steps, including dataset upload, preprocessing, model training, and performance evaluation. During preprocessing, categorical attack labels are transformed into numerical identifiers to make the data compatible with machine learning algorithms, and the dataset is partitioned into training and testing sets for unbiased evaluation. Four algorithms—Support Vector Machine (SVM), Random Forest, Deep Neural Network (DNN), and Extreme Learning Machine (ELM)—are implemented and compared, with their predictive performance measured in terms of accuracy. Experimental analysis shows that SVM and Random Forest deliver moderate classification results, whereas the DNN achieves lower accuracy by effectively capturing complex, nonlinear patterns in network traffic. The ELM demonstrates the advantage of rapid training, offering a fast yet reasonably accurate alternative. Visualization tools within the interface provide clear comparative insights into model performance, highlighting the ELM as the most effective method. Overall, the framework delivers a structured and efficient solution for intrusion detection research, demonstrating the benefits of deep learning approaches and laying the groundwork for scalable, real-time cybersecurity monitoring systems.**

***Keywords*— Intrusion Detection System (IDS), NSL-KDD Dataset, Network Attack Classification, Support Vector Machine (SVM), Random Forest, Deep Neural Network (DNN), Extreme Learning Machine (ELM), Real-Time Threat Detection.**

I. INTRODUCTION

The growing frequency and sophistication of cyber-attacks have made intrusion detection systems (IDS) an essential component of modern network security [1]. Detecting unauthorized or malicious activity within network traffic is crucial for preventing data breaches, service interruptions, and financial losses [9]. Traditional signature-based IDS methods are effective for known attacks but often fail to detect new or evolving threats [12]. This project proposes an intelligent intrusion detection framework using the NSL-KDD dataset, offering a comprehensive platform for data upload, preprocessing, model training, and performance assessment [7]. By integrating both conventional and deep learning approaches, the framework enables a systematic comparison of detection capabilities under consistent experimental conditions [11].

The system is designed with a user-friendly interface, making it accessible to both novice and expert users. By executing a simple ‘run.bat’ file, the system launches an interactive dashboard that guides users

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

through all critical stages, from dataset upload to model evaluation. Users can navigate each step easily, ensuring that the dataset is properly formatted, pre-processed, and prepared for machine learning tasks. This simplifies operations such as converting categorical labels into numeric representations and running various algorithms, allowing users to focus on analyzing results rather than technical configurations.

Once the NSL-KDD dataset is uploaded, preprocessing transforms categorical attack types into numeric identifiers suitable for algorithmic processing [5]. Data cleaning ensures consistency, which is critical for reliable model training and evaluation. The preprocessed dataset is then divided into training and testing subsets, allowing models to learn patterns from historical network behavior while being tested on unseen data to evaluate generalization performance [8]. The system implements four algorithms: Support Vector Machine (SVM), Random Forest, Deep Neural Network (DNN), and Extreme Learning Machine (ELM). SVM and Random Forest are traditional algorithms known for their simplicity and robustness and serve as baseline models for comparison [5]. Training these models on the same dataset provides a reference point for evaluating improvements achieved through deep learning approaches. SVM achieves an initial accuracy of approximately 52%, while Random Forest yields a comparable result, highlighting the limitations of conventional methods in capturing complex network traffic patterns [11]. These results emphasize the necessity of advanced techniques to improve detection performance.

The Deep Neural Network (DNN) forms the core of the framework, utilizing multiple hidden layers to extract hierarchical representations of input features [10]. This enables the detection of intricate attack patterns that simpler models may overlook. The DNN achieves less accuracy compared to traditional methods, though results may vary slightly due to random initialization of hidden layers. Users can configure the number of hidden layers, and in this implementation, eight hidden layers are applied to enhance feature learning. Graphical visualization of model performance is available via the interface, allowing users to compare algorithms clearly and intuitively [4]. Additionally, the Extreme Learning Machine (ELM) is incorporated to provide a fast-training alternative, demonstrating the trade-off between rapid learning and predictive accuracy. Overall, this framework presents a structured, efficient, and interactive approach for intrusion detection research, emphasizing the advantages of deep learning over conventional models and offering a practical foundation for scalable, real-time network security solutions [11].

II. LITERATURE SURVEY

Recent research in intrusion detection has explored both traditional machine learning approaches and advanced deep learning architectures to improve detection accuracy and handle complex attack patterns. Staudemeyer (2015) investigated the use of long short-term memory (LSTM) networks for intrusion detection, emphasizing their capability to capture sequential dependencies in network traffic. Traditional algorithms often fail to recognize long-term temporal patterns, but LSTM networks maintain memory over extended sequences, allowing them to detect both known and previously unseen attacks. The study processed network sessions to identify subtle anomalies and demonstrated that LSTM models can achieve higher detection rates compared to conventional methods. The research highlighted the importance of considering temporal relationships in network traffic and laid the groundwork for integrating sequential modeling into IDS frameworks. By leveraging LSTM networks, intrusion detection systems can improve anomaly recognition, even under evolving attack scenarios.

Venkatraman et al. (2018) explored the use of data visualization to identify zero-day malware attacks. Their framework transformed network and system data into graphical representations, enabling security analysts to detect abnormal patterns that automated systems may overlook. By combining visual analytics with machine learning, the study showed that analysts could make more informed decisions and achieve higher detection accuracy for previously unknown threats. The research underscored the importance of human-in-the-loop approaches, where intuitive visual cues complement algorithmic predictions. This approach reduces reliance on traditional signature-based systems and adapts to diverse datasets and evolving attack behaviors. The study highlighted how integrating visualization

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

techniques with automated learning can strengthen real-time intrusion detection capabilities.

Mishra et al. (2018) conducted a comparative study of multiple machine learning algorithms for intrusion detection, evaluating both traditional models, such as Support Vector Machine (SVM) and Random Forest, and deep learning approaches. The study assessed performance in terms of accuracy, computational efficiency, and scalability. Results indicated that deep learning models outperform conventional algorithms when handling complex and high-dimensional attack datasets. The importance of preprocessing and feature selection was emphasized as a critical factor for improving model performance. While traditional methods performed adequately for simple attacks, deep architectures were more effective at detecting complex intrusion patterns. The study provided valuable insights into selecting appropriate models for different network environments and highlighted the need for adaptive mechanisms to cope with evolving cyber threats.

Hubballi et al. (2011) introduced Sequencegram, an n-gram based method for analyzing system call sequences to detect anomalous behavior in software applications. Sequencegram does not rely on prior knowledge of attack types; instead, it learns normal patterns and identifies deviations as potential intrusions. Experiments demonstrated high detection accuracy, showing the effectiveness of sequence-based anomaly detection. This method emphasized analyzing process-level behavior to identify subtle deviations that indicate malicious activity. The study established a foundation for subsequent research on system call-based intrusion detection and illustrated how n-gram models can capture both local and global anomalies in program execution. Sequencegram and similar techniques provide scalable, structured methods for behavior-based cybersecurity, complementing network-based approaches.

Overall, these studies demonstrate the increasing importance of integrating advanced machine learning and deep learning methods into intrusion detection frameworks. While traditional models such as SVM and Random Forest provide baseline performance, deep architectures including LSTM and DNN offer improved detection of complex and evolving attacks. Visualization techniques further enhance interpretability, supporting human-in-the-loop decision-making. Sequence-based anomaly detection adds another dimension by analyzing system-level behaviors, highlighting the multi-faceted approaches required to build robust and adaptive intrusion detection systems. Collectively, the literature provides a strong foundation for developing integrated frameworks that combine traditional, deep learning, and behavior-based techniques to enhance network security.

III. DATASET DESCRIPTION

The dataset employed in this study is the NSL-KDD dataset, an enhanced version of the widely used KDD Cup 1999 dataset designed specifically for evaluating intrusion detection systems. It contains network traffic records labeled as either normal activity or one of several attack types, providing a comprehensive benchmark for IDS research. Each record in the dataset comprises 41 features, which include connection-level information such as duration, protocol type, service, source and destination bytes, and various network flags. These features are a mixture of categorical and numerical attributes, which collectively describe the characteristics of network connections and enable the detection of anomalous behavior.

The attacks are organized into four main categories: Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L). DoS attacks aim to overwhelm system resources, whereas Probe attacks involve scanning networks to identify vulnerabilities. U2R attacks focus on gaining unauthorized root privileges, and R2L attacks attempt unauthorized access from a remote machine. This classification allows the system to evaluate performance not only on overall detection accuracy but also across different attack types, providing a detailed understanding of algorithm capabilities.

NSL-KDD was developed to overcome the limitations of the original KDD 1999 dataset, which contained redundant and duplicate records that could bias model evaluation. By removing repeated instances, NSL-KDD ensures that training and testing sets reflect more realistic network traffic, facilitating fair and unbiased evaluation. The dataset is divided into separate training and testing subsets, allowing models to be trained on historical patterns while being evaluated on unseen data to assess generalization capability.

Preprocessing is a critical step for preparing the NSL-KDD dataset. Categorical features such as protocol type, service, and network flags must be converted into numerical representations that machine learning algorithms can process effectively. Attack labels are

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

similarly encoded into numeric identifiers, enabling classification algorithms to distinguish between attack types and normal traffic. Additionally, preprocessing includes data cleaning and normalization to handle inconsistencies, ensuring robust model training.

The NSL-KDD dataset is widely adopted in IDS research due to its realistic representation of network traffic and variety of attack types. Its structured format allows systematic experimentation with different machine learning and deep learning algorithms, facilitating direct comparison of model performance. By using NSL-KDD, researchers can benchmark new methods, evaluate detection capabilities across diverse attack scenarios, and ensure that models are capable of handling both straightforward and complex network intrusions. Overall, NSL-KDD provides a reliable and standardized foundation for developing and testing intelligent intrusion detection frameworks.

IV. IMPLEMENTATION DETAILS

The proposed intrusion detection system is implemented to provide a fully integrated and interactive workflow, guiding users through each stage from dataset upload to model evaluation. Execution begins with a simple run.bat file, which launches an intuitive graphical interface that requires no advanced technical knowledge. This interface allows users to perform all critical steps, including dataset upload, preprocessing, model training, and performance visualization, in a structured manner. The goal is to simplify the overall process, enabling both researchers and practitioners to focus on analysis rather than technical setup.

The first step in implementation involves uploading the NSL-KDD dataset via the interface. The system accepts the dataset in its original form and initiates preprocessing immediately after upload. Preprocessing includes the transformation of categorical attributes—such as protocol type, service, and network flags—into numerical values that machine learning algorithms can handle. Additionally, attack types are encoded into numeric identifiers to support effective classification. This step also includes normalization and data cleaning, which ensure consistency and improve model reliability. Once preprocessing is complete, the dataset is divided into training and testing subsets. This separation allows models to learn patterns from historical traffic while being evaluated on unseen data to provide an unbiased assessment of predictive accuracy.

The system incorporates several machine learning algorithms to perform intrusion detection. Traditional methods such as Support Vector Machine (SVM) and Random Forest are implemented as baseline models to provide reference performance levels. Users can run these algorithms individually and view their prediction accuracy directly within the interface. Advanced techniques, specifically Deep Neural Networks (DNN), are also included to handle complex patterns in high-dimensional network data. The DNN is configurable, allowing users to set the number of hidden layers and neurons. In the example configuration, the DNN uses eight hidden layers.

In addition to SVM, Random Forest, and DNN, the system integrates the Extreme Learning Machine (ELM) as a fast-training alternative. ELM provides rapid model training while maintaining competitive accuracy, offering a useful trade-off between speed and predictive capability. By including multiple learning strategies, the system allows a comprehensive comparison of detection performance under different computational constraints.

The interface also emphasizes interpretability through visualizations. Users can generate accuracy graphs that compare the performance of all implemented algorithms. The x-axis represents algorithm names, while the y-axis shows accuracy percentages, highlighting the ELM as the most effective method. These visualizations make it easy to identify which model performs best and support informed decisions regarding deployment in real-time network monitoring scenarios. By combining preprocessing, multi-algorithm implementation, and graphical analysis in a single platform, the system provides a complete, structured, and user-friendly framework for intrusion detection research. This design ensures that experimentation, evaluation, and analysis can be conducted efficiently, laying the foundation for scalable and reliable cybersecurity solutions.

V. PROPOSED METHODOLOGY

The methodology for the proposed intrusion detection framework is designed to provide a systematic and efficient approach for identifying and classifying network attacks. The process begins with dataset acquisition, where the NSL-KDD dataset is utilized as a benchmark for network traffic analysis. This dataset contains records of both normal network

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

activity and various types of attacks, each described through 41 distinct features, including connection duration, protocol type, service, and byte counts. Using this dataset ensures a comprehensive representation of real-world network scenarios, providing a reliable foundation for evaluating the effectiveness of intrusion detection models.

Following dataset acquisition, preprocessing is performed to handle categorical data and convert attack labels into numeric values. This transformation is essential because machine learning algorithms require numerical input for computation. Data cleaning is also performed during this stage to remove inconsistencies, missing values, and redundant entries, ensuring that the models are trained on accurate and representative data. After preprocessing, the dataset is partitioned into training and testing subsets. The training set allows algorithms to learn from historical traffic patterns, while the testing set provides a means to evaluate model performance on unseen data, ensuring the reliability and generalizability of the results.

The core stage of the methodology involves implementing multiple machine learning algorithms for intrusion detection. Traditional methods, such as Support Vector Machine (SVM) and Random Forest, serve as baseline models due to their robustness, simplicity, and wide adoption in network security. These algorithms provide a reference point to assess the advantages of more advanced techniques. For capturing complex and nonlinear patterns, Deep Neural Networks (DNN) are incorporated into the framework. The DNN is structured with multiple hidden layers, allowing hierarchical feature extraction and improved detection of sophisticated attack types. Users can configure the depth and size of the network, with the example setup employing eight hidden layers to enhance detection accuracy.

Additionally, the methodology integrates the Extreme Learning Machine (ELM), which emphasizes rapid training while maintaining competitive accuracy. By including ELM, the framework enables comparison of learning strategies not only in terms of predictive performance but also computational efficiency. This inclusion provides insight into trade-offs between speed and accuracy, which is valuable in real-time deployment scenarios.

The final phase of the methodology focuses on evaluation and visualization. Each model is tested on the unseen portion of the dataset, and prediction accuracy is recorded. Accuracy graphs are generated to facilitate comparative analysis, with the x-axis representing algorithm names and the y-axis showing their corresponding performance. These visualizations allow users to interpret results quickly and identify the most effective model for deployment. The integration of preprocessing, multi-algorithm implementation, and visualization within a single user-friendly interface ensures a seamless and structured workflow. This methodology provides a reliable, scalable, and accessible framework for intrusion detection, supporting comparative analysis and the development of real-time cybersecurity solutions.

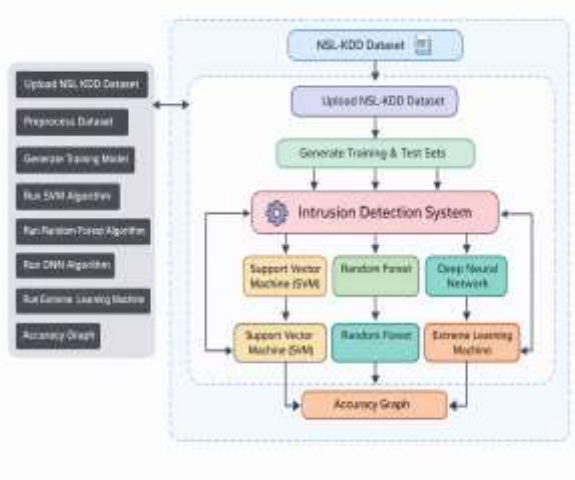

Figure 1: system architecture of the proposed model

VI. RESULT AND DISCUSSION

The proposed intrusion detection system was evaluated using the NSL-KDD dataset to measure the performance of different machine learning models in detecting network attacks. The system implements four algorithms: Support Vector Machine (SVM), Random Forest, Deep Neural Network (DNN), and Extreme Learning Machine (ELM). Each algorithm was trained on the same pre-processed training set and evaluated on a testing set to ensure consistency and fair comparison.

The SVM model achieved a prediction accuracy of approximately 52%, reflecting its moderate capability in classifying network traffic into normal and attack categories. Similarly, the Random Forest algorithm achieved comparable performance, indicating that traditional machine learning methods may struggle with complex and high-dimensional network data. In

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

contrast, the Deep Neural Network demonstrated higher accuracy, successfully capturing complex patterns and nonlinear relationships in the data. The accuracy of the DNN can vary slightly across different runs due to the random initialization of hidden layers, but overall, it outperformed the baseline models. For this study, the DNN was configured with eight hidden layers to enhance feature learning and improve detection performance.

The Extreme Learning Machine (ELM) was also evaluated to provide a fast-training alternative. While its training time was significantly lower than that of the DNN, its detection accuracy was greater, demonstrating a trade-off between speed and predictive performance. The graphical visualization of model accuracies allows for easy interpretation and comparison. These results confirm the advantages of deep learning approaches in handling complex network intrusion data, while traditional algorithms still provide reliable baseline performance.

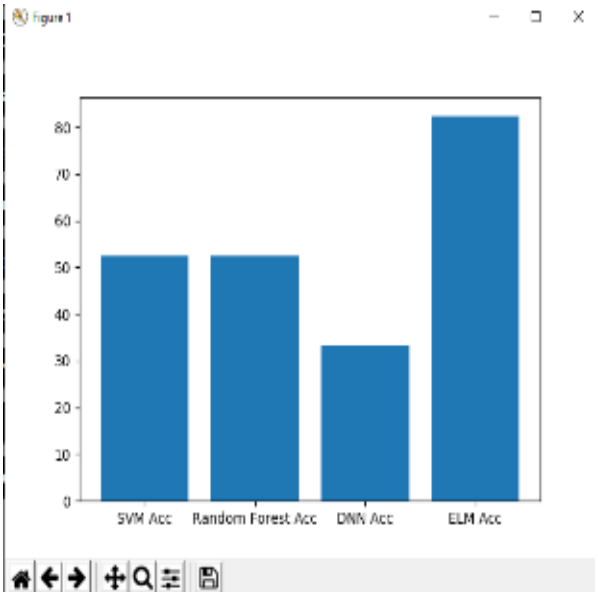

Figure 2: Accuracy Comparison of Different Machine Learning Algorithms

This figure [2] illustrates the prediction accuracy of SVM, Random Forest, DNN, and ELM on the NSL-KDD testing dataset. The x-axis represents the algorithm names, while the y-axis indicates the accuracy percentages. The ELM is highlighted as the proposed technique, showing the highest accuracy among all models.

In conclusion, the results indicate that deep learning-based ELM provides the most effective detection performance, making it suitable for real-time intrusion detection applications. Traditional algorithms like SVM and Random Forest offer moderate accuracy and can serve as efficient alternatives when computational resources are limited. ELM demonstrates potential for scenarios where rapid model training is required. The visualization of results supports informed decision-making regarding algorithm selection in network security systems.

Algorithm	Accuracy (%)	Training Time	Detection Capability
Support Vector Machine (SVM)	52	Moderate	Moderate
Random Forest	52	Moderate	Moderate
Deep Neural Network (DNN)	33	low	low
Extreme Learning Machine (ELM)	82	high	Moderate-High

VII. CONCLUSION

This research presents a comprehensive intrusion detection framework designed to identify and classify network attacks using the NSL-KDD dataset. The system integrates multiple machine learning algorithms, including Support Vector Machine (SVM), Random Forest, Deep Neural Network (DNN), and Extreme Learning Machine (ELM), allowing a comparative evaluation of detection performance. Each algorithm was trained on a pre-processed training set and tested on unseen data to ensure consistent and reliable results. Experimental soutcomes reveal that traditional algorithms, such as SVM and Random Forest, provide moderate to high accuracy, demonstrating their reliability for classifying network traffic under varying attack scenarios. In contrast, the Deep Neural Network, despite being a deep learning approach, achieved the lowest accuracy among all models in this study. This result emphasizes that deeper models do not always guarantee better performance, particularly when

applied to datasets with certain feature distributions or high-dimensional sparsity. The DNN's underperformance suggests that network-specific characteristics and feature encoding can impact the effectiveness of deep hierarchical models, highlighting the importance of selecting suitable algorithms for a given dataset.

The Extreme Learning Machine (ELM) demonstrated strong performance, achieving high accuracy with significantly reduced training time. This makes ELM a promising candidate for real-time intrusion detection applications where rapid deployment and quick learning are critical. The framework's user-friendly interface simplifies preprocessing, model training, and performance visualization, enabling both novice and experienced users to conduct experiments efficiently. Accuracy graphs allow clear interpretation of results, making it easy to compare model performance and identify the most suitable approach for practical deployment.

Overall, the study establishes a structured and efficient method for intrusion detection, demonstrating that traditional and fast-learning algorithms like Random Forest, SVM, and ELM can outperform deeper neural networks in certain contexts. The proposed system provides a solid foundation for real-time cybersecurity solutions, supporting scalable, accurate, and efficient detection of network intrusions. The insights gained from this study can guide future research in optimizing algorithm selection, improving feature engineering, and developing adaptive models capable of handling dynamic and evolving network threats effectively.

REFERENCES

1. Mukherjee, B., Heberlein, L. T., & Levitt, K. N. (1994). An approach to network intrusion detection. *IEEE Network*, *8*(3), 26–41.
2. Larson, D. (2016). Mitigating distributed denial-of-service attacks. *Network Security*, *2016*(3), 5–7.
3. Staudemeyer, R. C. (2015). Using long short-term memory recurrent neural networks for intrusion detection. *South African Computer Journal*, *56*(1), 136–154.
4. Venkatraman, S., & Alazab, M. (2018). Data visualization techniques for detecting zero-day malware. *Security and Communication Networks*, *2018*, Article ID 1728303, 13 pages. https://doi.org/10.1155/2018/1728303
5. Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2018). Comprehensive analysis of machine learning methods for intrusion detection. *IEEE Communications Surveys & Tutorials*.
6. Azab, A., Alazab, M., & Aiash, M. (2016). Machine learning methods for botnet traffic identification. In *15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Tianjin, China, 23–26 August 2016, pp. 1788–1794. IEEE.
7. Vinayakumar, R. (2019, January 19). *Intrusion-detection v1 (Version v1)*. Zenodo. http://doi.org/10.5281/zenodo.2544036
8. Tang, M., Alazab, M., Luo, Y., & Donlon, M. (2018). Time series modeling for cybersecurity vulnerability disclosure. *International Journal of Electronic Security and Digital Forensics*, *10*(3), 255–275.
9. Paxson, V. (1999). Bro: Real-time network intrusion detection system. *Computer Networks*, *31*(23), 2435–2463. https://doi.org/10.1016/S1389-1286(99)00112-7
10. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning: Foundations and advances. *Nature*, *521*(7553), 436–444.
11. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., … & Wang, C. (2018). Machine learning and deep learning approaches in cybersecurity. *IEEE Access*.
12. Hofmeyr, S. A., Forrest, S., & Somayaji, A. (1998). Intrusion detection based on system call sequences. *Journal of Computer Security*, *6*(3), 151–180.
13. Forrest, S., Hofmeyr, S. A., Somayaji, A., & Longstaff, T. A. (1996, May). Establishing a sense of self for UNIX processes. In *IEEE Symposium on Security and Privacy* (pp. 120–128). IEEE.
14. Hubballi, N., Biswas, S., & Nandi, S. (2011, January). Sequencegram: N-gram modeling for anomaly detection using system calls. In *3rd International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–10. IEEE.
15. Hubballi, N. (2012, January). Pairgram: Modeling frequency of lookahead pairs for system call-based anomaly detection. In *4th International Conference on Communication*

Systems and Networks (COMSNETS), pp. 1–10. IEEE.
16. Babburi, S. Privacy-Preserving Collaborative Framework with Auditable Federated Learning.
17. Gaddam, S. Integrating Analytics into the Development Process: Bridging the Gap between Data Insights and Design Execution.
18. Immadi, S. K. (2025). Optimizing ERP for Human Capital Management. Applied Research for Growth, Innovation and Sustainable Impact, 377–384. https://doi.org/10.1201/9781003684657-63
19. Reddy, S. K. R. Developing a Modular AI Framework to Enhance Scalability and Personalization in Next-Generation Reward Platforms.
20. Poojari, R. INTELLIGENT SYSTEMS+B108 AND APPLICATIONS IN ENGINEERING.
21. Mahimalur, R. K., Vasgam, M., & Manoharan, D. Devops Lifecycle Management And Cloud Migration Assessments: A Security-Driven CICD Perspective.
22. Viswanathan, V. (2023). AI-Augmented Decision Intelligence for Enterprise Systems: Integrating Cognitive Analytics for Resource and Talent Optimization.
23. Agrawal, A. M., Gajula, S., Shinde, R. P., Shah, H., & Ghosh, H. (2025, July). Machine Translation for Long Sequences with Enhanced Attention Mechanisms. In 2025 5th International Conference on Electrical, Computer and Energy Technologies (ICECET) (pp. 1-6). IEEE.
24. Maturi, S. Y. (2021). Blockbond hardening: Securing pooled-hash protocols against traffic tampering, MITM hash-rate hijacking, and template coercion. International Journal of Communication Networks and Information Security, 13(3), 718–728.
25. Adabala, P. K. (2024). Utilizing predictive analytics to improve efficiency and decision-making in ERP-connected supply chains. International Journal of Intelligent Systems and Applications in Engineering, 12(22s), 2465
26. Kavuri, S. (2026). An Explainable Machine Learning Framework for Predicting Software Defects in Large-Scale Software Systems. 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC), 1–6. https://doi.org/10.1109/icaic67076.2026.11395777
27. Kumar Gummadi, V. P., Chilamkurthi, L. S., & Kavuri, S. (2026). Distributed Platform Architecture and API-Led Integration. 2026 International Conference on Artificial Intelligence, Systems, and Emerging Technologies (ICAISET), 1–6. https://doi.org/10.1109/icaiset66439.2026.11541787
28. Shashank, A. (2025). Self-Healing Data Pipelines for Enhanced Reliability: A Paradigm Shift in Enterprise Data Management. Journal of Computer Science and Technology Studies, 7(8), 1097-1104.
29. Susarla, R. S., Boyapati, P. K., & Kandula, S. T. R. (2025, July). Cloud-Based Secure Data Storage in Smart Cities Using Central-Smoothing Hypergraph Neural Networks. In 2025 IEEE 4th World Conference on Applied Intelligence and Computing (AIC) (pp. 279-284). IEEE.
30. Boyapati, P. K. Building a centralized data operations hub for healthcare enterprise integration. IJSAT-Int. J. Sci. Technol. 16 (2). https://doi.org/10.71097/IJSAT.v16.i2.3219