Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Design and Implementation of a Deep Learning-Based Real-Time Intrusion Detection System Using Transformers and Explainable AI

Pasupula Srinivas
Assistant Professor, Department of CSE (AI & ML)
Vignana Bharathi Institute of Technology,
Hyderabad-501301, India

Kunduru Adithya Reddy
UG Student, Department of CSE (AI & ML)
Vignana Bharathi Institute of Technology,
Hyderabad-501301, India

Mamidi Abhinaya
UG Student,
Department of CSE (AI & ML)
Vignana Bharathi Institute of Technology,
Hyderabad-501301, India

Md. Mubeen
UG Student,
Department of CSE (AI & ML)
Vignana Bharathi Institute of Technology,
Hyderabad-501301, India

Abstract - Contemporary network infrastructures depend on automated, intelligent mechanisms to detect malicious activity and preserve operational security. This paper presents the design and implementation of a real-time Intrusion Detection System (IDS) that integrates Transformer-based deep learning models with Explainable Artificial Intelligence (XAI) techniques. The proposed system analyzes live network traffic, performs multi-class intrusion classification, and generates feature-level explanations to support security analyst decision-making. A supervised Transformer model constitutes the core detection engine, while SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) are employed to ensure transparency and foster operational trust. The system is evaluated on two widely accepted benchmark datasets, NSL-KDD and CICIDS2017, achieving accuracy values of 97.1% and 98.3%, respectively. Inference latency measurements confirm the system's suitability for near-real-time deployment. The findings demonstrate that integrating explainability directly into an end-to-end deep learning-based IDS pipeline is both feasible and practically beneficial.

Keywords - Anomaly Detection, Deep Learning, Explainable Artificial Intelligence, Intrusion Detection Systems, LIME, Network Security, SHAP, Transformer Models, Zero-Day Attacks.

1. INTRODUCTION

Intrusion Detection Systems (IDS) have become an indispensable component of modern network security architectures. Their primary role is to identify malicious activity that threatens the confidentiality, integrity, and availability of networked resources. As contemporary network environments grow in scale and complexity — driven by cloud computing, the proliferation of connected devices, and high-bandwidth applications — manual traffic inspection has become operationally impractical. Automated IDS solutions have therefore assumed a foundational role in enabling continuous, real-time monitoring and timely identification of threats.

Traditional signature-based IDS approaches depend on predefined rule sets to match known attack patterns. Although reliable for detecting previously catalogued threats, these systems are poorly equipped to handle novel, zero-day attacks and require frequent manual maintenance. Machine learning-based approaches partially address this limitation by learning generalizable patterns from labeled data; however, their effectiveness often hinges on handcrafted feature selection and may degrade in the face of high-dimensional, non-linear traffic characteristics.

Recent advances in deep learning have yielded intrusion detection models based on Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks. These architectures automate feature extraction and have achieved improved detection accuracy on standard benchmarks. Nevertheless, their black-box nature significantly impedes adoption in real-world security operations, where security analysts must understand and validate the reasoning behind automated alerts.

Transformer architectures, originally introduced for natural language processing tasks, have demonstrated strong performance across a range of sequential data modeling problems. Their self-attention mechanisms allow them to capture long-range dependencies within input sequences, a property that is particularly valuable when analyzing complex network traffic patterns with temporal structure. Despite this promise, most existing Transformer-based IDS studies focus narrowly on detection accuracy and offline evaluation, with limited attention to system-level integration and real-time feasibility.

Simultaneously, Explainable Artificial Intelligence (XAI) has emerged as a critical research direction aimed at improving

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

the transparency of machine learning systems. Post-hoc explanation methods such as SHAP and LIME enable feature-level attribution of model predictions, providing human-interpretable justifications for classification outcomes. Within the IDS domain, explainability facilitates analyst trust and supports incident response workflows. However, existing work predominantly treats explainability as a post-processing analysis tool rather than an integrated component of a deployable intrusion detection pipeline.

Motivated by these observations, this paper presents the design and implementation of an end-to-end, real-time IDS that combines a Transformer-based classification engine with integrated XAI capabilities. The system emphasizes system-level integration, practical deployment feasibility, and operational transparency. The principal contributions of this work are as follows:

1. A real-time Transformer-based IDS pipeline capable of analyzing network traffic and producing timely, accurate intrusion classifications.
2. Direct integration of SHAP and LIME explainability mechanisms into the detection pipeline, enabling selective, on-demand generation of feature-level explanations.
3. Empirical validation on standard IDS benchmark datasets (NSL-KDD and CICIDS2017) to assess detection effectiveness and latency characteristics under simulated operational conditions.

The remainder of this paper is organized as follows. Section 2 reviews related literature on intrusion detection and XAI. Section 3 describes the proposed system architecture. Section 4 details the implementation methodology. Section 5 presents the experimental setup and evaluation metrics. Section 6 discusses the results. Section 7 addresses limitations and future research directions, and Section 8 concludes the paper.

2. RELATED WORK

Intrusion detection has been studied extensively over the past two decades, with approaches broadly categorized into signature-based, machine learning-based, and deep learning-based methods.

Signature-based IDS methods rely on predefined patterns or rules to detect known attack types. Tavallaee et al. [1] provided a detailed analysis of benchmark IDS datasets that helped expose the limitations of rule-based detection in generalizing to unseen attacks. Sharafaldin et al. [2] further contributed the CICIDS2017 dataset, providing a richer and more contemporary set of labeled network flows covering modern attack vectors. While these methods offer high precision for known threats, they are inherently reactive and unable to identify novel attack variants without manual rule updates.

Machine learning-based approaches improved detection generalizability by learning statistical patterns from labeled traffic data. Algorithms such as decision trees, support vector machines, and ensemble methods demonstrated the ability to identify attack categories without relying on explicit rule definitions. However, as noted by Sommer and Paxson [10], the direct application of standard machine learning techniques to network security settings introduces challenges related to high false positive rates and sensitivity to data distribution shifts.

Deep learning models have further advanced IDS capability by automating hierarchical feature extraction. Yin et al. [3] demonstrated the effectiveness of LSTM-based models for sequential network traffic analysis. Shone et al. [4] proposed a deep learning approach for network intrusion detection using non-symmetric deep autoencoders. Vinayakumar et al. [5] conducted a comprehensive evaluation of deep learning architectures for intrusion detection, reinforcing the advantage of automated representation learning over handcrafted feature engineering. Despite strong benchmark performance, these models operate as opaque black boxes, which limits analyst confidence in their outputs. Transformer architectures, introduced by Vaswani et al. [6] with the seminal self-attention mechanism, have been increasingly applied to sequential analysis tasks beyond NLP. Li et al. [7] explored attention-based deep learning models for network intrusion detection and reported promising results. Khan et al. [8] and Yang et al. [9] further examined scalable hybrid architectures, underscoring the capacity of attention mechanisms to capture complex inter-feature relationships in network traffic data. However, these works are predominantly focused on offline accuracy evaluation, and system-level integration aspects remain underexplored.

In the domain of explainable AI, Ribeiro et al. [11] introduced LIME, a technique that generates locally faithful explanations for individual model predictions by approximating decision boundaries with interpretable surrogate models. Lundberg and Lee [12] proposed SHAP, which provides theoretically grounded feature importance scores derived from cooperative game theory. Guidotti et al. [13] offered a comprehensive survey of black-box explanation methods, while Tjoa and Guan [14] examined the applicability of XAI to cybersecurity and healthcare contexts. Within the IDS domain, XAI has been explored to improve analyst trust and facilitate alert triage, though most studies treat explainability as an offline analysis step rather than a real-time pipeline component. The present work distinguishes itself from existing literature by concentrating on the end-to-end system design of a deployable IDS that integrates a Transformer classification engine with real-time explainability modules. Rather than proposing new learning algorithms, the emphasis is placed on modular architecture, operational feasibility, and the practical utility of integrated transparency mechanisms.

3. SYSTEM ARCHITECTURE AND DESIGN

The proposed IDS is conceived as a modular, end-to-end pipeline encompassing five functionally distinct components: data ingestion and preprocessing, intrusion detection, explainability generation, and visualization. Each module is designed for independent development and testing, ensuring system extensibility and facilitating future modifications.

3.1 Design Objectives

The architecture is guided by four primary objectives. First, real-time feasibility demands that the system process network traffic and generate detection decisions with latency suitable for operational deployment. Second, modular architecture ensures that individual components — preprocessing, model

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

inference, and explainability — can be independently modified or replaced without disrupting the broader pipeline. Third, detection effectiveness requires that the system achieve reliable classification performance across diverse attack categories and dataset characteristics. Fourth, explainability and trust necessitate that the system provide feature-level justifications for its decisions, enabling security analysts to interpret and validate classification outcomes.

3.2 Overall System Architecture

The system follows a layered design in which network traffic data flow sequentially through ingestion, preprocessing, classification, explanation generation, and visualization stages. Raw or pre-captured traffic data are first normalized and encoded in the preprocessing module before being passed to the Transformer-based detection engine. The detection engine produces classification labels indicating benign or malicious traffic. When required, the explainability module is activated to analyze the model's predictions and generate feature-level attribution reports. The resulting outputs, including detection labels and explanations, are displayed through a visualization interface designed to support analyst workflows.

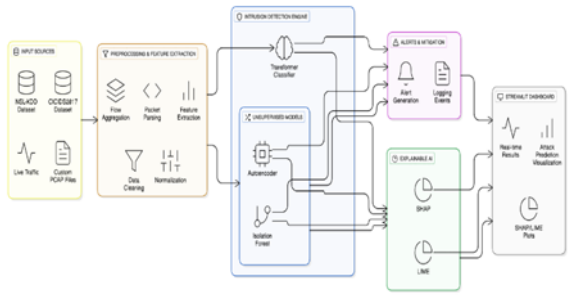

Fig. 1. Overall system architecture of the proposed IDS.

3.3 Data Ingestion and Preprocessing Module

The data ingestion module accepts network traffic records from benchmark datasets, including NSL-KDD and CICIDS2017-derived flow data. Input records undergo a structured preprocessing pipeline comprising attribute selection, missing value handling, numerical feature normalization, and categorical feature encoding. Feature sequencing operations are applied to preserve temporal and contextual relationships within traffic flows. The preprocessing module is optimized to operate with minimal computational overhead, ensuring that its latency contribution to the end-to-end pipeline remains negligible.

3.4 Intrusion Detection Engine

The detection engine constitutes the core classification component of the system. A supervised Transformer model is employed to analyze preprocessed traffic sequences and produce multi-class intrusion classification outputs. The model's self-attention mechanism enables it to capture complex intra-sequence dependencies across traffic features, which is particularly advantageous for detecting attack patterns that manifest across extended temporal windows. Model training is performed offline using labeled benchmark data, and the trained model parameters are persisted for runtime inference. The engine is designed to support extensibility through the optional integration of auxiliary detection modules, allowing hybrid detection strategies to be evaluated without architectural redesign.

3.5 Explainability Module

To address the transparency limitations inherent in deep learning-based classification, the system incorporates a dedicated explainability module. Post-hoc XAI techniques — specifically SHAP and LIME — are applied to quantify the contribution of individual input features to specific classification decisions. The module operates on demand and is selectively invoked based on detection confidence, thereby limiting its latency impact during routine operation. Both global explanations, which characterize overall model behavior across the dataset, and instance-level explanations, which justify individual predictions, are supported. This dual-level explainability enables analysts to both audit model behavior systematically and investigate specific alerts in detail.

3.6 Visualization and Monitoring Interface

The visualization and monitoring interface serves as the system's user-facing component. It presents real-time detection results, intrusion alert details, and feature-level explanation outputs in an accessible format. The interface is designed to support intuitive navigation of alert histories and explanation reports, facilitating informed decision-making during incident analysis.

4. SYSTEM IMPLEMENTATION AND METHODOLOGY

This section describes the implementation details of each system component and the operational methodology governing their integration and execution within the detection pipeline.

4.1 Data Sources and Preparation

The system is evaluated using two benchmark IDS datasets: NSL-KDD and a processed flow-based subset of CICIDS2017. NSL-KDD, an improved version of the KDD Cup 1999 dataset, provides 41 features spanning four attack categories (DoS, Probe, U2R, and R2L) and has been widely adopted as a standard evaluation benchmark. The CICIDS2017-derived dataset contains 78 flow-level features and covers a broader range of modern attack types including DDoS, brute force, botnets, web attacks, and infiltration.

Prior to inference, all records undergo preprocessing to ensure consistency across datasets. This includes removal of redundant or irrelevant attributes, imputation of missing values, min-max normalization of continuous features, and one-hot encoding of categorical attributes. The preprocessing pipeline is implemented as a reusable module within the system to ensure reproducibility across experimental configurations.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

4.2 Feature Handling and Sequencing

Processed traffic records are organized into fixed-length input sequences appropriate for the Transformer's positional encoding mechanism. Feature ordering is preserved to maintain temporal and contextual relationships among traffic instances. Dimensionality reduction is applied as a configurable preprocessing step to mitigate computational overhead while retaining the most discriminative attributes for classification.

4.3 Intrusion Detection Engine Configuration

The Transformer model is configured with a multi-head self-attention layer followed by position-wise feed-forward sub-layers and layer normalization. Model training is conducted offline using a labeled training split of each benchmark dataset, optimized using cross-entropy loss and the Adam optimizer with a scheduled learning rate. Trained weights are serialized and loaded at inference time, enabling the detection engine to operate without retraining during deployment. Hyperparameters are selected through grid search over a validation partition.

4.4 Explainability Integration

SHAP is applied using a gradient-based explainer adapted for the Transformer architecture, computing Shapley values that quantify each feature's marginal contribution to the model's output. LIME constructs locally linear surrogate models around individual prediction instances to generate interpretable approximations of the Transformer's local decision boundary. Both methods are invoked through the explainability module's API and their outputs are associated with the corresponding detection records for downstream visualization. Selective invocation — triggered only for high-confidence or anomalous classifications — limits the module's impact on system throughput.

4.5 Execution Workflow

The system's execution follows a linear, modular workflow. Network traffic data are ingested and passed through the preprocessing module, after which the Transformer engine performs classification. Detection outputs are evaluated against a confidence threshold; instances below the threshold or flagged as malicious trigger explainability computation. Final results — encompassing classification labels, confidence scores, and, where applicable, feature attributions — are forwarded to the visualization interface.

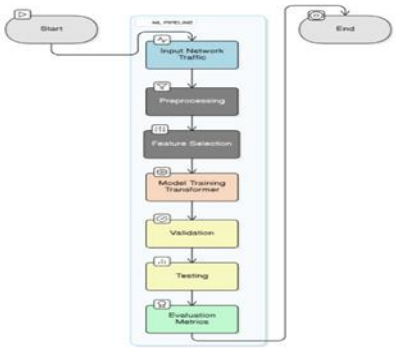

Fig. 2 Workflow of the Intrusion Detection System.

5. EXPERIMENTAL SETUP AND EVALUATION METRICS

5.1 Experimental Configuration

All experiments are conducted in an offline environment that simulates operational conditions. The detection model is trained on a labeled training partition of each dataset and evaluated on a held-out test partition to assess generalization performance. Preprocessed data are fed into the pipeline in batch sequences consistent with the system's real-time execution mode, ensuring that evaluation results reflect realistic deployment behavior. The system's modular design permits the independent evaluation of individual components, supporting reproducibility and facilitating future experimental extensions.

Table 1 provides a summary of the datasets used in this evaluation.

Dataset	No. of Features	Attack Categories
NSL-KDD	41	DoS, Probe, U2R, R2L
CICIDS2017 (Processed)	78	DDoS, Brute Force, Botnet, Web Attacks, Infiltration

Table 1. Summary of Datasets Used for Evaluation.

5.2 Evaluation Metrics

System performance is assessed using four standard IDS classification metrics. Accuracy measures the proportion of correctly classified instances across all classes. Precision indicates the fraction of flagged instances that are genuinely malicious. Recall quantifies the system's sensitivity in identifying all malicious instances. The F1-Score provides a harmonic mean of precision and recall, offering a balanced measure robust to class imbalance. In addition to classification metrics, end-to-end inference latency is measured in milliseconds to evaluate the system's suitability for real-time operation.

6. RESULTS AND DISCUSSION

6.1 Classification Performance

The proposed system achieves strong classification performance on both evaluation datasets. On NSL-KDD, the system attains an accuracy of 97.1%, precision of 96.8%, recall of 97.4%, and an F1-Score of 97.1%. On the CICIDS2017-derived dataset, performance is further improved, with accuracy of 98.3%, precision of 98.0%, recall of 98.5%, and an F1-Score of 98.2%. These results, summarized in Table 2, confirm the system's capability to reliably distinguish between benign and malicious traffic across datasets with substantially different feature spaces and attack distributions.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
NSL-KDD	97.1	96.8	97.4	97.1
CICIDS2017	98.3	98.0	98.5	98.2

Table 2. Intrusion Detection Classification Performance.

6.2 Comparison with Prior Work

Table 3 presents a comparative overview of the proposed system against representative prior IDS methods on the NSL-KDD and CICIDS2017 datasets. The results indicate that the proposed Transformer-based approach achieves competitive or superior accuracy relative to CNN- and LSTM-based methods, while additionally offering integrated real-time explainability — a capability absent in prior works listed below.

Method	Dataset	Accuracy (%)	F1-Score (%)	Explainability
LSTM-IDS [4]	NSL-KDD	95.6	95.2	No
Deep Autoencoder [5]	NSL-KDD	96.9	96.5	No
Attention-Based DL [8]	CICIDS2017	97.8	97.5	Partial
Hybrid Deep IDS [9]	CICIDS2017	97.4	97.1	No
Proposed System	NSL-KDD	97.1	97.1	Yes (SHAP+LIME)
Proposed System	CICIDS2017	98.3	98.2	Yes (SHAP+LIME)

Table 3. Comparison with Prior IDS Methods.

Note: Accuracy values for prior methods are sourced from the respective original publications. Direct reproduction of all experimental conditions is not guaranteed; the comparison is intended to provide indicative context rather than definitive benchmark ranking.

6.3 Inference Latency

Table 4 summarizes the measured latency of individual system components. Data preprocessing contributes an average of 3.8 ms per instance, while Transformer inference requires 11.6 ms. When the explainability module is active, SHAP and LIME computation adds 18.4 ms, resulting in an end-to-end latency of 33.8 ms — within practical bounds for near-real-time intrusion detection.

System Component	Average Latency (ms)
Data Preprocessing	3.8
Transformer-Based Detection	11.6
Explainability Module (SHAP/LIME)	18.4
End-to-End System Latency	33.8

Table 4. Inference Latency of System Components.

6.4 Analysis of System Behavior

Several key observations emerge from the experimental results. The modular architecture enables stable performance across datasets with heterogeneous feature spaces and attack distributions, requiring no dataset-specific architectural modifications. The Transformer's self-attention mechanism proves effective as a core detection component, capturing contextual dependencies within sequential traffic representations that are difficult to model with traditional convolutional or recurrent architectures. The integration of SHAP and LIME explanations provides meaningful feature-level attributions without inducing prohibitive latency, as the selective invocation strategy balances transparency with throughput.

7. LIMITATIONS AND FUTURE WORK

While the proposed system demonstrates strong performance and practical feasibility, several limitations merit acknowledgment. First, evaluation is conducted in an offline experimental setting using benchmark datasets. Although NSL-KDD and CICIDS2017 are widely adopted and representative, operational network environments exhibit a degree of traffic diversity, concept drift, and adversarial variation that controlled benchmarks may not fully capture. Validation on live network traffic remains an important direction for future work.

Second, the current system architecture relies on supervised learning with labeled training data. While this approach is well-suited for known attack categories, it provides limited support for detecting genuinely novel or zero-day threats without retraining. Incorporating semi-supervised or online learning mechanisms would enhance the system's adaptability to evolving attack landscapes.

Third, the post-hoc explainability methods employed, while effective, introduce computational overhead proportional to the number of features and the complexity of explanation requests. For large-scale, high-throughput deployment scenarios, this overhead may become a bottleneck. Future work will investigate lightweight, model-intrinsic interpretability approaches — such as attention visualization

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

and integrated gradients — that may offer comparable insight with reduced latency.

Additional future directions include extending the system to support real-time packet-level analysis in live network environments, adapting the architecture for resource-constrained settings such as IoT networks and software-defined networking infrastructures, and exploring federated learning paradigms to enable privacy-preserving, distributed intrusion detection.

8. CONCLUSION

This paper presented the design and implementation of a real-time, end-to-end Intrusion Detection System integrating a Transformer-based deep learning classification engine with explainable AI mechanisms. The system was developed as a modular pipeline capable of ingesting network traffic, performing multi-class intrusion classification, and generating feature-level explanations to support security analyst decision-making.

Experimental evaluation on the NSL-KDD and CICIDS2017 benchmark datasets yielded classification accuracy values of 97.1% and 98.3%, respectively, alongside F1-Scores above 97% on both benchmarks. End-to-end inference latency of 33.8 ms confirms the system's suitability for near-real-time operation. The integration of SHAP and LIME explainability within the detection pipeline enhances operational transparency without imposing prohibitive computational costs.

The results demonstrate that combining modern Transformer architectures with integrated interpretability mechanisms in a cohesive system yields a practical and trustworthy intrusion detection solution. This work contributes a deployment-oriented perspective that complements existing research focused on algorithmic detection performance, and establishes a foundation for further investigation into adaptive, explainable, and large-scale network security systems.

9. ACKNOWLEDGMENT

The authors gratefully acknowledge the Department of Computer Science and Engineering at Vignana Bharathi Institute of Technology for providing the necessary computing facilities and technical infrastructure to conduct this research. The authors also thank the faculty members and academic mentors for their sustained guidance and encouragement throughout the development of this project.

10. REFERENCES

[1] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009.

[2] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. ICISSP, 2018.

[3] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.

[4] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50, 2018.

[5] R. Vinayakumar et al., "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, 2019.

[6] A. Vaswani et al., "Attention is all you need," in Proc. NeurIPS, 2017.

[7] Y. Li, R. Wang, J. Hu, and H. Liu, "An attention-based deep learning model for network intrusion detection," IEEE Syst. J., vol. 15, no. 1, pp. 1–10, 2021.

[8] M. A. Khan, M. Karim, and Y. Kim, "A scalable and hybrid intrusion detection system based on deep learning," IEEE Access, vol. 7, pp. 65564–65575, 2019.

[9] Y. Yang et al., "Building an effective intrusion detection system using deep learning," IEEE Access, vol. 9, pp. 113720–113733, 2021.

[10] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in Proc. IEEE S&P, 2010.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in Proc. ACM SIGKDD, 2016.

[12] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Proc. NeurIPS, 2017.

[13] R. Guidotti et al., "A survey of methods for explaining black box models," ACM Comput. Surv., vol. 51, no. 5, pp. 1–42, 2018.

[14] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (XAI): Toward medical and cybersecurity applications," IEEE Access, vol. 8, pp. 220–235, 2020.

11. AUTHOR BIOGRAPHIES

P. Srinivas is an Assistant Professor in the Department of Computer Science and Engineering at Vignana Bharathi Institute of Technology, Hyderabad, India. His research interests include machine learning, deep learning, network security, and data analytics.

Mamidi Abhinaya is an undergraduate student in the Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning) at Vignana Bharathi Institute of Technology, Hyderabad, India. She is the team leader of this project. Her research interests include deep learning, intrusion detection systems, explainable artificial intelligence, and cybersecurity.

Kunduru Adithya is an undergraduate student in the Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning) at Vignana Bharathi Institute of Technology, Hyderabad, India. His research interests include machine learning, network security, and data-driven system development.

Md. Mubeen is an undergraduate student in the Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning) at Vignana Bharathi Institute of Technology, Hyderabad, India. His research interests include deep learning, cybersecurity, and intelligent system design.