SvedbergOpen
DISSEMINATIONOFKNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Protocol-Aware Anomaly Detection for Transport and Network Layer Attacks: A Comprehensive Review and Intelligent Framework Design

Chandan Wagh[1],Kamal Kant Hiran[2], Amit Kumar Goel[3]

[1]PhD Research Scholar, Department of Computing & Informatics, SPSU, Udaipur, India
chandanwagh@gmail.com
[2]Associate Professor, Department of Computing and Informatics SPSU, Udaipur, India
kamalhiran@gmail.com
[3]Professor, Department of Computing and Informatics SPSU, Udaipur, India
dean.fci@spsu.ac.in

Abstract
The threats are more complex in modern networks such as IoT systems, cloud infras-tructures, and distributed corporate networks. Among certain transport and network based protocols like TCP, UDP, and ICMP, have been used in flooding, amplification and spoof-ing attacks; in particular, it is the transport layer that is the primary target. Traditional intrusion detection systems generally depend on signature-based or flow-level analysis, which can leave out details ofprotocol-level anomalies. This paper introduces a protocol-aware anomaly detection framework based on packet-level analysis instead of aggregated traffic flows. Specifically, this system combines feature extraction from TCP, UDP, and ICMP headers with a lightweight hybrid deep learning model architecture using CNN and GRU architectures. This design allows spatial- and temporal-scale detection of network traffic patterns.

Keywords
Intrusion Detection System (IDS), Cyber Threat Intelligence, Protocol-Aware Anomaly Detection, TCP Flooding Attacks, UDP Amplification Attacks, ICMP Abuse Detection, Lightweight Deep Learning, CNN-GRU Hybrid Model, Real-Time Network Security, Au-tomated Mitigation

1 Introduction

Rapid development of digital technologies such as cloud computing, Internet of Things (IoT), intelligent transportation systems leads to increased complexity of contemporary communication networks, especially with the rapid growth of digital technology. These technological developments help in ensuring accurate data sharing or automation as well as have serious cybersecurity consequences. Especially transport and network layer pro-tocols such as TCP, UDP and ICMP are continuously attacked by Distributed Denial of Service (DDoS), flooding, spoofing, and reconnaissance attacks. Such emerging and previously unknown types of evolving and novel threats can in most cases not be well addressed by signature-based security mechanisms [1, 2]. Intrusion Detection Systems (IDS) are widely employed in monitoring network traffic and identifying malicious ac-tivities. These systems are typically divided into signature-based, anomaly-based, or hybrid approaches [3]. Signature-based techniques are good for identifying attack routes that are already known but work very poorly compared to new attacks and new threats. Conversely, anomaly-based systems try to discover a characteristic of normal network behavior and differentiate between abnormal behavior, making it more feasible for zero-day attack detection [4]. Superior performance of IDS models is being achieved due to recent developments in machine learning (ML) and deep learning (DL). CNN, LSTM, hybrid CNN–LSTM architectures, etc. have shown good performance in the detection of sophisticated attack structures [5, 6]. Moreover, attention-based and transformer-based models have given it additional potential for detection by catching temporal dependen-cies and context in network traffic [7]. Feature engineering and ensemble-based learning techniques have been successfully leveraged to improve detection robustness over widely used benchmark datasets including NSL-KDD and UNSW-NB15 [8,9]. Nonetheless, from previous experience, we realized that there still exist a number of limitations to current IDS research. Firstly, most of these models are primarily used to analyze the behavior at flow-level rather than probing the protocol-specific behavior. As such, TCP handshake manipulation, ICMP echo abuse, and UDP amplification attacks are not distinguished into three groups effectively [10]. Second, previous studies have mostly focused on detec-tion accuracy and have not accounted for automated approaches for prevention which were

considered to be key for real-time response [11]. Third, deep learning models are computationallyexpensive, restricting their implementation in resource-poor environments (e.g., IoT devices and edge networks) [12]. To improve transparency and trust in AI-based se-curity systems, explainable artificial intelligence (XAI) techniques have been introduced in recent studies [13]. Explainability is particularly important in critical systems where incorrect decisions may lead to serious consequences [14]. Additionally, synthetic data generation techniques have been used to address class imbalance and improve model ro-bustness [15]. Currently, however, most existing approaches do not consolidate protocol awareness, lightweight design, explainability, and automated mitigation into a single uni-fied framework. In this vein, this paper introduces a protocol-aware anomaly detection framework that leverages packet-level behavior instead of aggregated traffic characteris-tics as a sole focus. Adopting protocol-aware feature extraction with a lightweight and hybrid deep learning method, I demonstrate the ability to achieve effective multi-class real-time detection. The system also introduces automated mitigation mechanisms to rapidly respond to detected threats, making it suitable for the IoT (Internet of Things) and edge-enabled environment. The key contributions of this work can be summarized as listed below:

- Design protocol-aware anomaly detection model specially tailored to TCP, UDP, and ICMP attack patterns.
- Architecture for lightweight hybrid deep learning architecture for real-time deploy-ment.
- The integration of adaptive response with automation of mitigation mechanisms
- Anticipated validation based on common intrusion detection datasets.
- Comparison with existing ML/DL-based IDS methods

The rest of this paper is organized the way it does. Related work is described in Sec-tion II. Section III introduces the research gap and problem formulation. The proposed system architecture is presented in Section IV and discussed in subsequent section on methodology and analysis.

2 Related Work

Over the last decade, intrusion detection systems have become much more sophisticated with introduction of human-like technological approaches. Previous algorithms were bi-ased toward statistical and traditional models using existing algorithms. With network traffic becoming more complex, dynamic, and challenging, deep learning-based meth-ods are under increasing attention for being able to automatically extract insights from large data. Recently research has largely used the hybrid deep learning models, which uses CNN and RNN. Indeed, CNN-LSTM models have demonstrated great potential in catching both the spatial features and the temporal features of network traffic [16]. Likewise, bidirectional recurrent networks enhance contextual understanding by process-ing sequences in both forward and backward directions [17]. Typically, these hybrid approaches outperform traditional classifiers in multi-class intrusion detection scenar-ios. optimizations were performed in a way to improve the performance of the model aside from architectural enhancements. Swarm intelligence methods like Particle Swarm Optimization (PSO) commonly serve as an effective way to tune hyperparameters and enhance convergence [18]. These techniques, though can increase detection capability, they impose computational overhead and hence can hinder real-time deployment. IDS capabilities have been broadened by generative models as well. Specifically, Generative Adversarial Networks (GANs) are used to create synthetic attack data and handle prob-lems of class imbalance [19]. Although these approaches enhance robustness, they are limited by their training instability and make models more complex. Nowadays, fea-ture engineering has reached a critical stage of intrusion detection. Feature selection and decreasing the dimensionality reduce computational costs that affect detection per-formance [20]. Ensemble learning approaches, such as Random Forest and XGBoost, showed strong performing baseline model in IDS applications [21]. But, most such ap-proaches are dependent on the flow-level features and are not designed to be sensitive to implementation behaviour depending on the protocol. Given that the Software-Defined Networking (SDN) and IoT environments have emerged, these introduce new threats to intrusion detection. Machine Learning-Based IDS Model for SDN-enabled IoT Sys-tems has achieved promising results [22]. However, most of these systems depend on

Table 1: Comparison of Commonly Used Intrusion Detection Datasets

Dataset	Year	Traffic Type	Attack Coverage	Protocol Detail	Limitations
KDD Cup 99	1999	Simulated	DoS, Probe, R2L, U2R	Low	Outdated and re-dundant data

NSL-KDD	2009	Simulated	DoS, Probe, R2L, U2R	Low	Limited representation of modern attacks
UNSW-NB15	2015	Realistic synthetic	Fuzzers, DoS, Exploits	Medium	Limited IoT realism
CICIDS2017	2017	Real traffic	DDoS, Botnet, Brute force	Medium	Primarily flow-based features
BoT-IoT	2018	IoT traffic	DDoS, Reconnaissance	Medium	Class imbalance issue
Recent IoT datasets	2022–2025	IoT / Edge	Mixed attacks	Medium	Lack of protocol-aware modeling

centralized detection and do not incorporate adaptive mitigation practices. Explain-able artificial intelligence (XAI) has recently emerged as an important design feature in IDS. In the sense of increasing transparency and trust in decision-making, models in-corporating systems with attention mechanisms and interpretability tools like SHAP are proposed [23]. While these methods contribute to interpretability, they do at the same time add computational complexity and are no longer suitable for lightweight implemen-tations. There are also lightweight IDS models researched for resource-limited settings. Some of these approaches convert protocol header information into image-like format for the CNN-based classification [24]. Although these techniques decrease the complexity of parameters, they are generally restricted to certain protocol structures and cannot be generalised to varied network conditions. New insights were added through wireless and vehicular networks research. IEEE 802.11 MAC layer state behavior analysis provides the capability of detecting anomalies at lower layer level [25]. Analogously time-based detecting techniques have been proposed using controller area networks (CAN) to detect masquerade attacks [26]. These methods are, however, domain specific and not applicable to general networks with an IP foundation. From the above the discussion, one can see that even though substantial progress has been achieved in ML and DL-based intrusion detection, there are still number of serious problems to solve. Most of the available solutions concentrate on providing excellent performance on benchmarks but do not model in the clear protocol-level for TCP, UDP, and ICMP traffic. Moreover, real-time mitigation isn't always built in and lightweight deployment is still difficult. The limitations of this approach indicate the demand for a common infrastructure combining protocol awareness with explainability, computational efficiency and adaptive mitigation.

3 Research Gap and Problem Statement

While machine learning and deep learning approaches have enhanced the performance of intrusion detection systems significantly, there are still some real-world issues to tackle. Due to these problems, existing IDS solutions are not practical for real-world operation

Table 2: Classification of IDS Approaches in Literature

Category	Learning Type	Protocol Awareness	Explainabi	itMyitigation	Deployment Suitability
Signature-Based IDS	Rule-based	Low	High	Manual	High
ML-Based IDS	Supervised	Limited	Low	Rare	Medium
DL-Based IDS	Deep Learning	Limited	Low	Rare	Low/Medium
GAN-Based IDS	Generative	Limited	Low	No	Low
XAI-Based IDS	Hybrid	Limited	High	Rare	Medium
Proposed Framework	Hybrid CNN-GRU	High	High	Automated	High

in diverse, heterogeneous networks [27]. To take it to an extra step, most existing IDS frameworks analyze behaviour at the flow level or aggregated traffic characteristics, with-out analysis for protocol-specific behavior. Benchmark datasets such as CICIDS2017 and UNSW-NB15 are being used mainly for

assessment [27] but there is no requirement to directly incorporate transport and network layer features. Thus, such attacks such as TCP SYN flooding, UDP amplification, and ICMP echo abuse are generally considered as general anomalies and not realized for their structural properties. Newer algorithms that leveraged hybrid CNN and LSTM architectures can still show remarkable accuracy, but they are not designed to directly model the protocol dynamics [28]. Second, per-formance of some IDS models is still highly dependent on evaluation metrics, including accuracy, precision, and F1-score [29]. Though useful metrics, they provide little indica-tion of real-world usability. The fact is that in practice, most detection systems send out alerts, where manual intervention is needed or a firewall configured externally [30]. This results in increased response time and less effectiveness of the system during large-scale attacks. Third, there is still massive limitation in computational complexity. Sophisti-cated architectures including swarm optimization, transformer-based learning, or adver-sarial training often consume heavy computational costs [31]. Although such models are capable of achieving excellent performance in controlled environments, they cannot be easily implemented in IoT devices, edge nodes, or distributed systems where resources are scarce. Fourthly, explainable artificial intelligence (XAI) methods have been introduced to enhance transparency [32].

Nevertheless, by and large, most XAI-based IDS approaches explain after detection and do not necessarily implement them into real-time decision-making or mitigation mechanisms. Also, such approaches usually increase the computational burden. Fifth, synthetic data-generation methods have been employed to compensate the dataset imbalance [33]. But above all, these methods mainly focus on increasing the diversity in datasets rather than enhancing the generalization for a particular protocol. A lot of existing research is limited to benchmark datasets and is not validated in diverse networks, including IoT, SDN, and enterprise systems. Another fundamental limitation is the absence of incorporation with cyber threat intelligence mechanisms. Most existing IDS systems do not provide for any adaptive feedback system for automatic adaptive detection thresholds/signatures and mitigation-policy updates. Thus, these systems have a difficult time responding to changing attacks. From these studies the research problem is characterized as:

How to design a lightweight, protocol-aware, and interpretable anomaly detection frame-work that could target TCP, UDP, and ICMP-based attacks and manage, in a highly-automated real-time manner, such real-data mitigation in a heterogeneous network en-vironment? This work presents a new unified framework that combines protocol-level feature extraction, hybrid deep learning (DL), adaptive thresholding, and automated response mechanisms to tackle this problem.

Figure 1 illustrates the overall workflow of the proposed detection and mitigation pro-cess. This framework adopts a methodical pipeline that guarantees a structured way of analyzing data processing, conscious detection, and adaptation.

3.1 Data Collection

It begins by extracting raw traffic from the network in the form of routers, gateways, switches, or interface monitors. Packet-level details (TCP flags, UDP datagram proper-ties, and ICMP message types) are stored to facilitate protocol-aware analysis.

3.2 Data preprocessing

The raw dataset is preprocessed for consistency. This involves removal of redundant or corrupted records, handling of missing values, numerical feature normalization as well as encoding of categorical fields. These steps prepare the data for effective model training.

3.3 Feature extraction

We analyze the packets to determine the protocol-aware features. Protocol behavior must be inferred instead of being a function of the known techniques, such as combi-nations of TCP flags, UDP traffic patterns, and ICMP message frequency. Next come some additional statistical features (packet rate, bytes). Dimensionality reduction and computational capabilities are handled by feature selection.

3.4 Design of the model

We are building a lightweight hybrid deep learning model by means of hybrid deep learn-ing model which combine convolutional and recurrent layers. Convolution layers are the ones that learn how space among features is mapped, and recurrent layers (e.g., GRU, LSTM) encode network traffic temporal dependencies. For final classification, fully con-nected layers are used. The model is trained with labeled benchmark datasets. Both binary and multi-class classification tasks are possible. Training is carried out under backpropagation with a tuned adaptive learning rate to minimize classification loss.

3.5 Anomaly Detection

We analyse the incoming traffic streams after training and assign an anomaly score to each segment. The traffic is categorized into normal vs. TCP/UDP/ICMP attack. Confidence values for predicting reliability are also calculated.

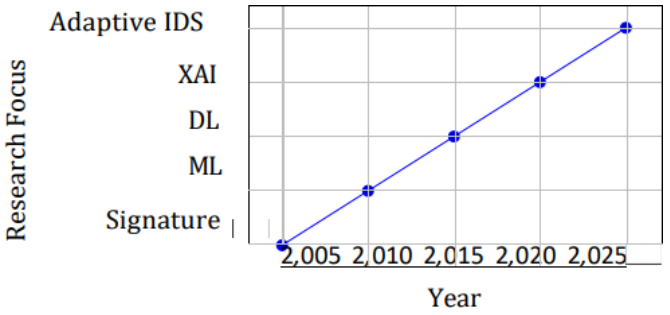

Figure 1: Evolution of Intrusion Detection Research Paradigms

3.6 Threshold Establishment

A dynamic threshold is employed to differentiate regular traffic variance and malicious traffic changes. The threshold is adjusted based on historical traffic patterns as well as the confidence level of detection. These include standard metrics (accuracy, precision, recall, F1-score, and false positive rate) to evaluate the performance of the system. Cross-validation is used to guarantee a robust design against different traffic conditions.

3.7 Real-Time Detection

The model then is tested in a real-time mode to make sure it works. Sliding window analysis allows dynamic monitoring of live traffic streams, enabling early detection of protocol-specific attacks.

3.8 Alert and Automated Prevention

When attacks are found, the system automatically performs mitigation actions like block-ing suspicious IP addresses, applying rate limiting, and updating threat intelligence logs. This minimizes the latency response time and improves network resilience.

Table 3: Summary of Representative AI-Based Intrusion Detection Studies

Ref.	Model Type	Dataset	Protocol Focus	Explainability	Mitigation	Key Limitation
[28]	CNN–LSTM Hybrid	NSL-KDD	Flow-based	No	No	Limited protocol awareness
[30]	Deep Neural Network	CICIDS2017	Flow-based	No	No	High computational cost
[31]	Autoencoder-based DL	UNSW-NB15	Flow-based	No	No	No adaptive thresholding
[38]	GAN-based IDS	CICIDS2017	Flow-based	No	No	Training instability
[18]	XAI-based DL Model	IoT Dataset	Partial	Yes	No	Lack of real-time mitigation
[36]	Distributed DL IDS	IoT Network	Flow-based	No	No	Deployment complexity
[44]	Random Forest	IoT Dataset	Flow-based	Limited	No	Weak zero-day detection
Proposed Work	CNN–GRU Hybrid	Multi-dataset (Planned)	Protocol-aware	Yes	Yes	Pending validation

4 Proposed System Architecture

The architecture of our AI-powered cyber threat intelligence platform for real-time anomaly detection and automated mitigation of TCP, UDP and ICMP-based attacks is described in this section. It is a modular pipeline in which different stages contribute to the effi-cient processing of the data, accurate detection and fast response. Aspects of the overall

flow of the system are demonstrated in Fig. 2. From raw data acquisition to automated mitigation of detected threats: the framework operates in an organized series of steps.

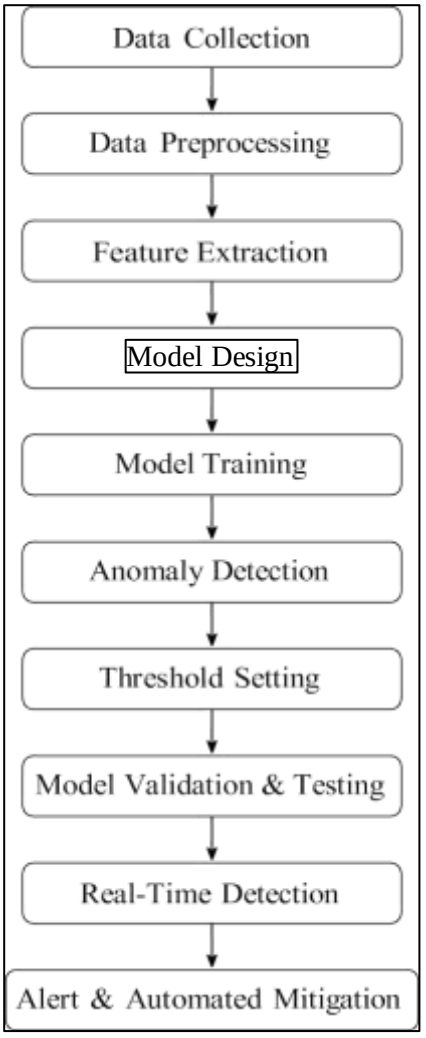

Figure 2: Proposed Real-Time Detection and Automated Mitigation Architecture

The system operates as a sequence of interconnected stages. Although the workflow is sequential, each component is designed to be modular and adaptable based on deployment requirements. The main stages of the system are summarized as follows:

1. Data Collection
2. Data Preprocessing
3. Feature Extraction
4. Model Design
5. Model Training
6. Anomaly Detection
7. Threshold Setting
8. Model Validation and Testing
9. Real-Time Detection
10. Alert and Automated Mitigation

4.1 Data Collection

Raw network traffic from routers, gateways, switches, or monitoring interface sources is collected, the first step. Unlike the classical data processing methods, the system can still store packet-level information in form of TCP flags, UDP packet characteristics, and ICMP message types.

4.2 Data Preprocessing
The collected traffic is processed to make it suitable for machine learning models. This involves removing duplicate or corrupted records, dealing with missing values, normalizing numerical features, and encoding categorical variables. These steps are used for enhancing the consistent data and stabilizing the training process.

4.3 Feature Extraction
Feature extraction focuses on deriving meaningful attributes from packet-level data. This methodology focuses on protocol-aware features, not only based on flow-level statistics. E.g., TCP flag patterns, UDP traffic rate variations, ICMP message frequency. Comple-mentary statistical features such as packet count and byte rate are introduced as well. Feature selection is used for dimension reduction and increase in efficiency.

4.4 Model Design
To support real-time operation, a lightweight hybrid deep learning model is developed. It is composed of convolutional layers to extract spatial relationships and recurrent layers (such as GRU or LSTM) for capturing temporal dependencies in traffic sequences. Fully connected layers are used for final classification.

4.5 Model Training
The model is trained on labeled datasets. It supports binary (normal vs attack) and multi-class (attack types in different classes) classification. Training uses backpropagation and adaptive optimization techniques as well as regularization to prevent overfitting.

4.6 Anomaly Detection
After training, incoming traffic is processed by this model, assigned a prediction score and an initial classification of traffic (normal, TCP, UDP, or ICMP). The prediction reliability is also evaluated using values of probability.

4.7 Threshold Setting
Distinguishing among normal traffic variances and malicious activity is enabled via its high level dynamic threshold. The system does not set a specific threshold, but adapts it according to past behaviour and familiarity with detection.

4.8 Model Validation & Testing
The system is also open for evaluation in terms of performance metrics, such as accuracy, precision, recall, F1-score, and false positive rate. Since the model works over many network states cross-validation helps keep the model generalizing well.

4.9 Real-time detection
Once validated, the model is then set up for real time monitoring. Incoming traffic is continuously checked using sliding window to enable early detection of attacks such as flooding, spoofing, and abuse of the protocols.

4.10 Alert and automatic mitigations
The automated system activates protective controls when the automatic system detects the attack. Examples of such actions are blocking suspicious IP addresses, implementing rate limiting and updating threat intelligence logs. This automated reaction reduces man-ual intervention and increases system responsiveness. The architecture of this proposal is closed-loop systems. Detection and mitigation can work together as a combination to defend the system against future cyber attacks.

5 Methodology
In this section, the method for protocol-aware anomaly detection and automation of pro-tection against TCP, UDP, and ICMP-based attacks is described. The pipeline structure of the project is comprised of: traffic representation, feature extraction, hybrid model construction, anomaly scoring, and adaptive response.

5.1 Traffic Modelingand Data Representation
sNetwork traffic is represented as a stream of packets followed over time. Let us now describe incoming traffic stream as:

$T = \{p1, p2, p3, ..., pn\}$ (1)

with p_i for each packet containing protocol specific header information. We transform the packet information into a feature vector for making the data amenable to learning:

$\mathbf{x}_i = [f1, f2, ..., fm]$ (2)

In this case, m is the value of the extracted features at the protocol level including TCP flags, behavior of ICMP messages and statistically computed traffic features and the characteristics of all the attributes.

5.2 Temporal model of sliding windows

Given that time series attacks are often short, packets are grouped via sliding window. For some window size w, we define the sequence of input as:

$\mathbf{X}_t = [\mathbf{x}_{t-w+1}, ..., \mathbf{x}_t]$ (3)

The representation allows for time dependencies to be accounted for, making this highly efficient in detecting burst-based attacks, for example, flooding and amplification.

5.3 Protocol-Conscious Feature Engineering

In contrast to a traditional flow-level statistics approach that is often oriented more towards feature-specific extraction based on multiple protocols, this work is focused on protocol-based feature extraction. Main categories of features are:

- TCP attributes like SYN/ACK ratio, flag combinations, and retransmission pat-terns
- UDP characteristics like packet rate variation and port entropy
- ICMP parameters like echo request frequency and the type distribution
- Statistical parameters such as packet count, byte rate and variance (on each time window)

Feature selection and normalization strategies are employed to minimize redundancy and scale the data to achieve efficiency.

5.4 Hybrid Learning Methodology

. We propose detection model as a lightweight hybrid model of convolutional and recur-rent layers. The latter, it is the learning space among attributes that the model learns, along with the temporal interdependencies across packet sequences. The prediction func-tion is expressed by:

$\hat{y} = F(\mathbf{X}_t; \theta)$ (4)

where $\hat{y}$ is the predicted class label, θ is the model parameters, and $F(\cdot)$ is the CNN–GRU mapping function. Local features pattern are extracted by convolutional layers:

$\mathbf{h}_c = \sigma(\mathbf{W}_c * \mathbf{X}_t + b_c)$ (5)

The recurrent (GRU) captures temporal dependencies:

$\mathbf{h}_t = GRU(\mathbf{h}_c)$ (6)

Finally, we rely on a softmax function to compute class probabilities:

$$P(y = k|\mathbf{X}_t) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \quad (7)$$

where K represents the number of output classes (Normal, TCP attack, UDP attack, ICMP attack).

5.5 Anomaly scoring and thresholding

. To quantify prediction confidence, we calculate an anomaly score S as maximum pre-dicted probability:

$$S = \max_k P(y = k|\mathbf{X}_t) \quad (8)$$

A dynamic threshold τ is used at τ to distinguish normal from anomalous traffic:

$$\text{If } S < \tau \rightarrow \text{Anomaly} \quad (9)$$

Rather than applying a specific value, the threshold is adjusted according to historical confidence:

$$\tau = \mu_S - \alpha\sigma_S \quad (10)$$

where μ_S and σ_S represent the mean and standard deviation of confidence scores, and α controls the sensitivity. The approach is developed for interpretability and for analyzing feature contributions for each prediction. This will provide for the examination of which protocol level attributes drove the decision, and make the system accessible and verifiable.

5.6 An automatic mitigation strategy

On detection of anomalous behavior, the system reacts by taking mitigation measures specific to the attack type:

$$M = \begin{cases} \text{Block IP} & \text{for TCP flooding} \\ \text{Rate limit} & \text{for UDP amplification} \\ \text{ICMP throttle} & \text{for ICMP abuse} \end{cases} \quad (11)$$

Automated response minimizes reaction time and enhances network security.

5.7 Computational Considerations

. Let d represent number of features, and w the size of the window. Computational complexity of each window can approximated in the form:

$$O(w \cdot d) \quad (12)$$

Due to the lightweight nature, the model can be deployed in resource-constrained envi-ronments like IoT gateways and edge devices.

5.8 Implementation Roadmap (Planned)

The framework proposed is intended for use across common tools in practice. Packet capture can be done with libraries like Scapy, and model development is done over Ten-sorFlow or PyTorch. Firewall or network control APIs allow for real-time mitigation. Experimental validation using benchmark datasets and evaluation under real-world traf-fic conditions will be the focus of future work.

6 Threat Model

Adversarial assumptions, attack capabilities, and system boundaries are described in this section, which are included in the proposed framework. Identifying the proper model allows realistic assessment and structured defense planning.

6.1 Adversarial Capabilities

It is assumed that the attacker possesses the following characteristics:

- The capability to generate high-volume TCP, UDP, or ICMP traffic.
- Able to spoof source IP addresses.
- Ability to exploit protocol-specific vulnerabilities such as SYN flooding, UDP am-plification, and ICMP echo abuse.
- Access to distributed botnet infrastructure for volumetric attacks.

But the attacker is not assumed to have:

- Physical access to network hardware.
- Compromise of the internal IDS model parameters.
- Ability to tamper with the threat intelligence repository.

6.2 Attack Surface

The common attack surface is:

- Network gateways and routers.
- Transport layer communication channels.
- IoT devices and edge-enabled nodes.
- Public-facing servers and APIs.

The framework addresses attacks on transport and network layers before they reach application-layer services.

6.3 Considered Attack Categories

The proposed system is directed at protocol-based anomalies, i.e.,

6.3.1 1) TCP-Based Attacks

- SYN flooding.
- ACK/RST flooding.
- Session hijacking attempts.
- Connection exhaustion attacks.

These attacks take advantage of loopholes in TCP handshake and connection manage-ment.

6.3.2 2) UDP-Based Attacks

- UDP flooding.
- DNS amplification.
- Reflection-based attacks.
- Port scanning anomalies.

High UDP packet rate irregularities and amplification behavior characterize UDP attacks.

6.3.3 3) ICMP-Based Attacks

- ICMP echo request flooding (Ping flood).
- Smurf attacks.
- ICMP tunneling anomalies.

These attacks take advantage of control-message protocols to lower the network perfor-mance.

6.4 Defense Assumptions

The framework proposed assumes the following:

- The network monitoring system has packet header information access.
- Sufficient computational resources are available for lightweight model inference.
- Rules for mitigation can be enforced even at the gateway or firewall level.

No encrypted payload inspection is needed as the model is essentially protocol-level meta-data dependent.

6.5 Security Objectives

The main security objectives of this framework are:

- Early detection of specific protocol anomalies.
- False positive rate minimization.
- Rapid automated mitigation.
- Retention of normal traffic flows.
- Adaptation to evolving attack patterns.

Defining attacker capabilities and system assumptions helps the proposed architecture systematically defend against transport-layer and network-layer threats. In this section, we provide a conceptual comparison of the conceptual framework of the present study and existing classes ofintrusion detection systems (IDS). Instead of evaluating accuracy alone, the comparison highlights architectural prowess, deployability, and operations. We con-clude that as such, a holistic and unified system solution addressing protocol awareness, lightweight design, explainability, and automated mitigation to provide modern network integration needs to be considered.

6.6 Comparative Capability Matrix

Table 4 summarizes the major differences between the conventional IDS, flow-based ma-chine learning IDS, deep learning-based IDS, generative/augmentation-based approaches, explainable IDS systems, and the proposed framework. The comparison is driven by met-rics typically addressed in recent literature [34]– [40].

Table 4: Comparative analysis of IDS categories and the proposed framework (feature-based comparison).

Capability	Tradition IDS	lFlow-based ML IDS	DL/ Hybrid IDS	GAN-based IDS	XAI-based IDS	Proposed Frame-work

Protocol-aware modelling (TCP/UDP/ICMP)	Low	Limited	Limited	Limited	Limited	High
Detection of un-known attacks	Low	Medium	Medium	Medium wLow	Medium Medium/Lo	High
Suitability foredge deployment	High	High	Medium/Lo	Low Low	High Low/Mediu	wHigh
Explainability	Low	Low	Low	Medium/Hi Medium	gMedium Medium	High
Automated miti-gation support	Low	Low	Low			mHigh
Adaptability to evolving threats	Low	Medium	Medium			High
Deployment in heterogeneous networks	Medium	Medium	Medium			High

6.7 Discussion

1) Constraints of traditional and flow-based IDS: For traditional IDS and vari-ous machine learning solutions, the basis of the IDS is mainly predefined signatures or aggregate flow statistics. As promising a technology as these may be to identify known attack vectors, none of them model protocol behavior overtly. Consequently, attacks such as tampering with the TCP handshake, UDP amplification, or abuse based on ICMP are typically considered as arbitrary outliers rather than as actual protocol-level events. Many of the IDS solutions are dataset-driven and do not include protocol-specific modeling [34, 35].

2) Deep learning and hybrid models: A number of IDS model applications, including CNN-based IDS and LSTM hybrid models, have been described, which enhance feature representation and classification speed and accuracy. However, these models are typically computationally expensive and non-interpretable. Moreover, many of these systems do not consider all other scenarios and can be validated just on some benchmark datasets [36, 37]. This establishes a misalignment between theoretical performance and practical usability.

3) Generative and augmentation-related methods: To address class imbalance and diversify training resources, GAN-based IDS models in conjunction with synthetic data generation are used. These approaches introduce larger training complexity and some instability as the model converges. However, these methods are not effective for real-time deployment and can also complicate their applications within edge environments or resource-constrained systems [38, 39].

4) Explainable IDS systems: Explainable AI methods enhance transparency through explanations of model decisions. These techniques provide an explanation for why an alert was generated, giving insight into a process leading to trust in the system. Nonetheless, the majority of XAI-based IDS paradigms are mainly post-analysis, failing to incorporate interpretability along with real-time mitigation [40].

5) Pros of the proposed framework: Finally, the proposed framework overcomes the shortcomings noted in existing approaches by combining multiple capabilities into a single architecture. It relies on protocol-aware feature extraction for TCP, UDP, and ICMP traffic in order to detect anomalies at the protocol level with greater specificity. A lightweight hybrid approach is applied to ensure real-time implementation, especially for IoT as well as edge architectures. Moreover, the framework presents interpretable outputs to inform management decisions, as well as automatic controls (e.g., blocking, throttling, rate constraints). Deep learning models have recently been shown to attain good classification outcomes [12, 15, 18, 21]. GAN-based approaches are also strong in robustness but increase complexity [24, 27, 29], while explainable IDS systems are helpful in interpretability [31, 33]. These features are, however, generally implemented indepen-dently. By synthesizing the above factors, the proposed framework presents a unified approach for practical application in modern heterogeneous networks.

7 Expected Performance Discussion

While experimental validation is not included in this conceptual study, the design of the proposed framework provides an opportunity to conceptualize and analyze its expected performance. The trade-off between protocol-driven feature extraction, lightweight hy- brid learning, and automated

mitigation is thought to enhance detection quality and ease deployment in systems as a real-time tool.

7.1 Enhanced Detection Accuracy

Current intrusion detection apps have reduced the features of flow to their aggregate features, and such can be blind to more refined network features. As a result, non-standard threats, such as exploitation of TCP handshake, UDP amplification, and ICMP abuse are not always obvious compared to natural traffic movement. We expect a new framework to incorporate protocol-level features explicitly, resulting in some specificities:

- A more precise separation between lawful high-volume traffic from malicious flood-ing.
- Improve the visibility of the transport-layer attacks that take advantage of protocol behavior.
- Prevent normal traffic bursts from being misclassified as anomalies.

The CNN/GRU combination captures both feature-level interaction and temporal pat-terns. This has been useful especially for short burst attacks and a series of anomaly occurrences. Reducing false positive rate is a requisite to effective IDS operation as too many alarms will make most of the security staff start to doubt and distrust the system. In the proposed framework based on statistical distribution of prediction confidence, a dynamic thresholding system is developed. Rather than stationary thresholds, changes in traffic patterns of it will probably result in:

- Allow more adaptive decision borders.
- Even across different sorts of network problems you would do better to improve robustness.
- Decrease needlessly mitigation actions.

7.2 Real-Time Detection Capability

Deep learning models frequently confront inference delay and computational expenses challenges. To solve this, a lightweight CNN–GRU architecture for efficient execution is used in the proposed system. Expected benefits include:

- Reduced inference time for each traffic window.
- Compatibility with low-resource environments like IoT gateways.
- Scalability to distributed network nodes.

With sliding window processing, traffic streams can be continuously observed without requiring batch processing; nearly real-time detection can be achieved.

7.3 Mitigation Latency Reduction

Unlike traditional IDS approaches which only produce alerts, we propose a mechanism for adopting automated mitigation. The system should, by connecting detection output with response actions such as blocking, rate limiting, and traffic control, be able to:

- Minimize delay caused by manual intervention.
- Reduce the duration and impact of attacks.
- Improve overall system responsiveness.

Because of this closed-loop design, the detection and response activities work together and provide enhanced defensive efficiency.

7.4 Scalability and Adaptability

This modular structure of the framework allows the framework to be deployed across heterogeneous environments, including enterprise networks, IoT systems, and edge infras-tructures. As the model is mainly reliant on protocol-level metadata, and not payload inspection, it is still efficient in encrypted traffic scenarios. Taken together, the pro-posed framework is expected to provide a balanced trade-off among detection accuracy, computational efficiency, interpretability, and automated response. These characteristics render it feasible for modern network environments where performance in combination with scalability is quite critical.

8 Research Gap Summary

Several limitations were observed repeatedly based on the previous existing methods of intrusion detection. The gaps are summarized in Table 5, and the proposed framework addresses each.

Table 5: Summary of Identified Research Gaps and Proposed Improvements

Identified Limitation	Existing IDS Ap-proaches	Proposed Framework
Flow-centric detection	Predominantly used	Replaced with protocol-aware analysis
Lack of protocol aware-ness	Frequently overlooked	Explicit protocol-level model-ing
Absence of automated mitigation	Common in most systems	Integrated real-time response mechanism
High computational com-plexity	Observed in deep learn-ing models	Lightweight hybrid architec-ture
Limited explainability	Partially addressed	Integrated feature-level inter-pretability
Static thresholding	Fixed decision rules	Dynamic adaptive threshold-ing

9 Protocol-Aware Attack–Feature Mapping

To better understand how protocol-level characteristics contribute to attack detection, Table 6 shows the mapping between common attack types and corresponding observable features. These capabilities are directly derived from TCP, UDP, and ICMP packet behavior and can therefore be used as the major indicators of an anomaly detection network.

Table 6: Mapping of attack types to protocol-aware features and detection indicators

Attack Type	Key Protocol Feature	Detection Indicator
TCP SYN Flood	Elevated SYN to ACK ratio	Irregular TCP handshake be-havior
TCP RST Flood	High frequency of RST flags	Abnormal connection termina-tion patterns
UDP Amplification	Large variation in packet size and payload	Sudden traffic bursts with am-plification characteristics
UDP Flood	Rapid increase in packet trans-mission rate	Port-level traffic spike detec-tion
ICMP Echo Flood	Increased echorequest fre-quency	Time-window based anomaly in ICMP traffic
ICMP Smurf Attack	Broadcast-based ICMP re-sponses	Reflection and amplification pattern detection

10 Detection Algorithm

Algorithm 1 Protocol-Aware Anomaly Detection and Mitigation
Capture incoming network packet stream in real time
Extract protocol-aware features from TCP, UDP, and ICMP headers
Organize packets into a sliding time window
Compute anomaly score using the hybrid CNN–GRU model
if a **then**
nomaly score < τ (dynamic threshold) Classify traffic as malicious
Identify attack type (TCP/UDP/ICMP)
Apply corresponding mitigation rule (block, rate limit, or throttle)
Update threat intelligence repository with new event
else

Classify traffic as normal and allow transmission
end if

11 Threat–Defense Mapping

Table 7 explains the correlation of prevalent protocol-level threats and their automated defenses. The mapping indicates how the proposed framework transforms detection outputs to practical defensive actions.

Table 7: Mapping of protocol-level threats to automated mitigation strategies

Protocol Layer	Threat Type	Mitigation Strategy
Transport (TCP)	SYN Flood	Blocking of suspicious source IP addresses
Transport (TCP)	RST Flood	Enforcement of connec-ion/session validation rules
Transport (UDP)	Amplification Attack	Application of rate-limiting policies
Network (ICMP)	Echo Flood	Controlled ICMP traffic throt-tling
Network (ICMP)	Smurf Attack	Filtering of broadcast-based ICMP traffic

TCP flooding and UDP amplification attacks continue to be widely observed in modern network environments, particularly in large-scale distributed attacks [5, 9].

12 Evolution of Intrusion Detection Systems

Intrusion Detection Methodologies have experienced significant evolution, evolving from static rule-based methods that are predicated on predefined rules to dynamic models that are dynamic and adaptive. Threats to networks and the network-side are also getting more complex, and network-side threats are being seen to evolve over time, thus making detection of threat more proactive and sophisticated. Most of the early IDS solutions used a signature-based approach, relying on predefined rules to detect known attack patterns. While these systems worked for known threats, they failed to identify new or unknown attacks. Machine learning techniques were implemented to enable IDS models to identify patterns and detect anomalies in their data. This is a shift in the direction of data-driven detection. Then deep learning methods also improved the detection by learning the complex correlation and temporal dependencies among features in traffic on the network. More recently, explainable artificial intelligence (XAI) has been integrated in the context of improving transparency and interpretability of IDS decisions. However many of these systems still do not integrate with protocol level analysis and automatic response mechanisms. As a result, the proposed framework advances this process and integrates protocol-aware feature extraction, automated adaptive learning, explainability, and automated mitigation into one system.

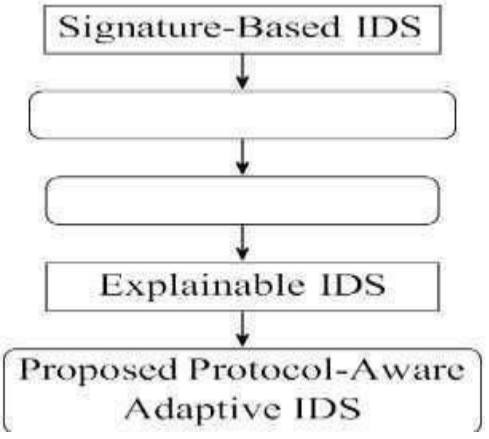

Figure 3: Evolution of intrusion detection systems from signature-based methods to protocol-aware adaptive frameworks

13 Limitations and Future Work

Although the framework presented proposes a methodical, protocol-based approach for anomaly detection and automated mitigation, some limitations do need to be acknowl-edged.

13.1 Noexperimental validation

The current research remains limited to concept and architecture formulation without experimental benchmarking. Although the proposed system technique and analysis can be said to improve detection efficiency, they need empirical support. The proposed work will have to be implemented and evaluated on benchmark datasets (e.g. CICIDS2017, UNSW-NB15) and real-world traffic traces to quantify detection capacity, false positives rate and computational efficiency.

13.2 Challenges Associated with Dataset Generalization

All network traffic characteristics vary between ERPs, IoT networks, cloud infrastructure, etc. Our proposed framework is based on the assumption that a representative set of training data is available. In future work, cross domain evaluation will also be reviewed to see the robustness and generalization of performance of this model across different environments.

13.3 Model Tuning and Edge Integration

The framework, while light, is a lightweight architecture and does not scale in resource limited environment. Traditional approaches on IoT gateways and edge IoT devices will require the optimization of memory usage and inference time. We will investigate methods such as model pruning, quantization and knowledge distillation to maximize deployment potential.

13.4 Adversarial robustness

In adversarial machine learning, recent work has shown that IDS models are susceptible to evasion attacks. The current framework does not explicitly cover an adversarial ro-bustness. Adversarial learning will be further integrated in future studies in addition to resisting such attacks and robustness-aware learning concepts, which will play a pivotal role in minimizing risk and resilience of defense.

13.5 Integration With Cyber Threat Intelligence Systems

The framework has threat intelligence but does not elaborate on how it can be inte-grated with external threat feeds and distributed intelligence platforms. In terms of fu-ture extensions we will look into real-time threat intelligence sharing, federated learning methodologies and adaptive synchronisation of policies across distributed environments.

13.6 Further extend Application-Layer Attacks

Current design focuses largely on transport-layer (TCP/UDP) and network-layer (ICMP) attacks. Application layer attacks such as HTTP-based attacks, SQL injections, and en-crypted traffic manipulation are ignored in this work. Subsequent studies could expand the framework towards multi-layer security monitoring. Nevertheless, despite the afore-mentioned limitations, the proposed protocol-aware and AI-driven framework has set a reasonable base for dynamic, explainable and efficient intrusion detection systems to cope with emergent cyber threats.

14 Conclusion

This paper introduced a protocol-aware, AI-powered cyber threat intelligence framework to identify, prevent, and mitigate TCP, UDP, and ICMP-based attacks. Unlike the ear-lier traditional flow-centric intrusion detection systems, the proposed paradigm relies on protocol-level behavior modeling to better capture structural anomalies in network traffic. The framework integrates protocol-aware feature extraction, a lightweight hybrid deep learning model, dynamic thresholding, explainability mechanisms, and automated mitiga-tion within one unified architecture. We are currently able to handle many problems found in existing IDS solutions, including lack of protocol awareness, limited interpretability, absence of real-time response capabilities, and so forth. The goal of the proposed de-sign is to obtain a compromise between detection accuracy, computational efficiency, and real-world deployment prospects. Notably, the lightweight hybrid model enables infer-ence in real time, whereas the closed-loop threat intelligence element supports automatic adjustment to different attack patterns. Although the study does not use experimental validation, architectural design and theoretical analysis show tremendous promise toward early detection for flooding, amplification, and protocol abuse attacks. We intend to im-plement the proposed system and evaluate its performance with benchmark datasets and real-world traffic. Further work will be applied to fine-tuning the model for edge deploy-ment, improving its robustness against adversarial attacks, and extending the

framework for multi-layer threat detection. This framework is a pragmatic approach to developing intelligent, scalable, and adaptive intrusion detection systems that can be used in modern network environments like those associated with IoT, enterprise, and distributed infrastructures.

References

[1] H. Alipour, Y. B. Al-Nashif, P. Satam, and S. Hariri, "Wireless Anomaly Detection Based on IEEE 802.11 Behavior Analysis," IEEE Trans. Inf. Forensics Security, vol. 10, no. 10, pp. 2158–2173, Oct. 2015.

[2] S. Lee, H. J. Jo, A. Cho, D. H. Lee, and W. Choi, "TTIDS: Transmission-Resuming Time-Based Intrusion Detection System for CAN," IEEE Access, vol. 10, pp. 52139–52152, 2022.

[3] P. Sharma, J. Anandan, H. Liu, and J. Grover, "Synthetic Attack Dataset Generation With ID2T for AI-Based Intrusion Detection in Industrial V2I Network," IEEE Open
J. Veh. Technol., vol. 6, pp. 2509–2518, 2025.

[4] A. J. Bhuvaneshwari, P. Kaythry, N. Elango, R. Ramu, and S. S. Eshwar, "Improving Security in 5G Vehicular Networks Using ASCON and ML-Based NIDS," Discover Internet Things, vol. 5, 2025.

[5] A. Hozouri, A. Mirzaei, and M. Effatparvar, "A Comprehensive Survey on Intrusion Detection Systems With ML and DL Advances," Discover Artificial Intelligence, vol. 5, 2025.

[6] A. I. Getman, D. A. Rybolovlev, and A. G. Nikolskaya, "Deep Learning Applications for Intrusion Detection in Network Traffic," Programming Comput. Softw., vol. 50, no. 7, pp. 493–510, 2024.

[7] Sarika and R. Dass, "Design of a Novel IDS for SDN-Based IoT Network Using ML," Optoelectron., Instrum., Data Process., vol. 61, no. 3, pp. 396–407, 2025.

[8] M. El-Sherif, A. Khattab, and M. El-Soudani, "TCP/IP Single Packet Header Binary Image IDS for IoT Networks," Cybersecurity, vol. 8, 2025.

[9] M. A. Alsharaiah et al., "Explainable Transformer-Based IDS for IoMT," Discover Applied Sciences, 2025.

[10] M. S. Ataa, E. E. Sanad, and R. A. El-Khoribi, "Deep Learning-Based Intrusion Detection in SDN," Scientific Reports, 2024.

[11] A. Sharma, S. Rani, and M. Shabaz, "AI-Augmented Smart Grid Intrusion Detec-tion," Scientific Reports, 2025.

[12] F. S. Alsubaei, "Smart Deep Learning Model for IoT Intrusion Detection," Scientific Reports, 2025.

[13] A. M. Alashjaee, "Attention-CNN-LSTM Model for Network Intrusion Detection," Scientific Reports, 2025.

[14] M. Farhan et al., "Network-Based Intrusion Detection Using Deep Learning," Scien-tific Reports, 2025.

[15] N. Dash et al., "Optimized LSTM-Based Intrusion Detection Using Swarm Intelli-gence," Cluster Computing, 2025.

[16] F. Mujawib Alashjaee, "Hybrid CNN-LSTM for Real-Time Intrusion Detection," Electronics, 2025.

[17] A. Mirzaei et al., "Machine and Deep Learning Solutions for Intrusion Detection in IoTs," IEEE Access, 2024.

[18] S. Alzakari et al., "Explainable AI-Based Cyber Resilience for IoT Networks," Sci-entific Reports, 2025.

[19] K. Karam et al., "ISAAF: IoT Security and Attack Prevention Framework," IEEE Access, 2025.

[20] A. Getman et al., "CNN-BiLSTM Model for Network-Based IDS," Programming Comput. Softw., 2024.

[21] S. Sarika and R. Dass, "Hierarchical ML/DL IDS Framework for SDN-Based IoT," Optoelectron., Instrum., Data Process., 2025.

[22] A. Hozouri et al., "AI-Based IDS Architecture and Emerging Cybersecurity Chal-lenges," Discover Artificial Intelligence, 2025.

[23] Y. Liu and B. Lang, "Machine Learning and Deep Learning Methods for Intrusion Detection," IEEE Commun. Surveys Tuts., 2019.

[24] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and IDS Evaluation Framework," ICISSP, 2018.

[25] M. Tavallaee et al., "A Detailed Analysis of the KDD Cup 99 Dataset," IEEE CISDA, 2009.

[26] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection," MILCOM, 2015.

[27] M. Ring et al., "A Survey of Network-Based Intrusion Detection Data Sets," Com-puters &

Security, 2019.

[28] W. Wang et al., "HAST-IDS: Learning Hierarchical Spatial-Temporal Features for IDS," IEEE Access, 2017.

[29] Y. Kim et al., "Long Short-Term Memory Recurrent Neural Network Classifier for Intrusion Detection," IEEE ICC, 2016.

[30] R. Vinayakumar et al., "Deep Learning Approach for Intelligent Intrusion Detection," IEEE Access, 2019.

[31] M. Shone et al., "A Deep Learning Approach to Network Intrusion Detection," IEEE Trans. Emerg. Topics Comput. Intell., 2018.

[32] H. Hindy et al., "A Taxonomy of Network Threats and the Effect of ML/DL-Based IDS," IEEE Access, 2020.

[33] A. Javaid et al., "A Deep Learning Approach for Network Intrusion Detection," EAI SecureComm, 2016.

[34] A. Ferrag et al., "Deep Learning for Cyber Security Intrusion Detection: Approaches and Trends," Future Generation Computer Systems, 2020.

[35] M. Conti et al., "Survey on Security in Internet of Things," IEEE Commun. Surveys Tuts., 2018.

[36] S. Diro and N. Chilamkurti, "Distributed Attack Detection Using Deep Learning in IoT," Future Generation Computer Systems, 2018.

[37] J. Kim et al., "Real-Time Intrusion Detection Using Deep Neural Networks," IEEE ICC, 2018.

[38] Z. Lin et al., "IDSGAN: Generative Adversarial Network for IDS Adversarial Attack," IEEE Access, 2019.

[39] H. Lashkari et al., "CICIDS2017 Dataset and Intrusion Detection Evaluation," IEEE ICISSP, 2017.

[40] M. Ahmed et al., "A Survey of Network Anomaly Detection Techniques," Journal of Network and Computer Applications, 2016.

[41] M. Ring, D. Schlör, D. Wunderlich, and A. Hotho, "Flow-Based Network Traffic Generation Using Generative Adversarial Networks," Computers & Security, vol. 82,
pp. 156–172, 2019.

[42] A. Ferrag, L. Maglaras, H. Janicke, and J. Jiang, "Deep Learning Techniques for Cyber Security Intrusion Detection: A Detailed Analysis," IEEE Access, vol. 7, pp. 101234–101254, 2019.

[43] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier," Computer Networks, vol. 174, 2020.

[44] K. S. Sahoo, B. Sahoo, A. K. Turuk, B. K. Sahoo, and M. S. Obaidat, "Random Forest Based Intrusion Detection System for IoT Networks," Procedia Computer Science, vol. 167, pp. 196–205, 2020.

[45] T. Kim, B. Kim, and H. Kim, "LSTM-Based System-Call Language Modeling for Anomaly-Based Host Intrusion Detection," IEEE Access, vol. 7, pp. 141536–141547, 2019.

[46] J. Ashraf, S. Latif, and S. Rho, "Hybrid Deep Learning Model for Network Intrusion Detection," Electronics, vol. 10, no. 9, 2021.

[47] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 303–336, 2014.

[48] S. M. Kasongo and Y. Sun, "A Deep Learning Method With Filter-Based Fea-ture Engineering for Wireless Intrusion Detection System," IEEE Access, vol. 7, pp. 38597–38607, 2019.

[49] A. A. Diro and N. Chilamkurti, "Distributed Attack Detection Scheme Using Deep Learning Approach for Internet of Things," Future Generation Computer Systems, vol. 82, pp. 761–768, 2018.

[50] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A Taxonomy of Network Threats and the Effect of Machine Learn-ing for Cybersecurity Intrusion Detection Systems," IEEE Access, vol. 8, pp. 104650–104675, 2020.